

Índice general

1. Planificación y configuración	5
1.1. Descripción de planificación de tiempos de desarrollo	5
1.1.1. Recursos humanos y roles	5
1.1.2. Fechas para los entregables	6
1.1.3. Plazos e hitos	6
1.1.4. Reuniones de Proyecto	6
1.1.5. Elección de tecnologías de desarrollo y frameworks	7
1.2. Demostración configuración entorno de desarrollo y herramientas de desarrollo, Configuración del sitio web de alojamiento del proyecto	7
1.2.1. Tickets (Asignación tareas Fase 0)	7
1.2.2. Wiki (Podemos ver una primera página editada con la primera reunión del equipo)	9
1.2.3. SVN (Con un fichero subido de prueba)	10
1.2.4. Entorno de Desarrollo NetBeans 6.0	11
1.3. Posibles herramientas a aplicar	12
1.3.1. Ant	12
1.3.2. JUnit	12
1.3.3. Log4j	12
1.4. Instalación y configuración de la herramienta TortoiseSVN	12
1.5. Instalación y configuración de MySQL	19
2. Requisitos y Arquitectura	29
2.1. Especificación de Requisitos y Casos de Uso del Sistema	29
2.1.1. ¿Qué es WSDM?	29
2.1.2. Descripción de los casos de uso	36
2.2. Arquitectura del sistema	44
2.2.1. Modelo	46
2.2.2. Vista	47
2.2.3. Controlador	47
3. Documento de Análisis	49
3.1. Diagrama de Despliegue	49
3.2. Diagrama de Clases	50
3.3. Diagrama de navegabilidad	51
3.4. Diagrama Entidad/Relación	52
4. Documentos de diseño	53
4.1. Diagramas Robustness	53
4.2. Diagramas de actividad	61
4.3. Capturas de pantalla	84
5. Casos de prueba	93

6. Persistencia de datos	95
6.0.1. Esquema Entidad-Relación de la Base de Datos	95
6.1. Hibernate	97
6.1.1. Descripción	97
6.1.2. Características	97
6.1.3. Consideraciones generales del funcionamiento	98
6.1.4. Ventajas e inconvenientes	99
6.1.5. Requisitos e incompatibilidades	100
6.1.6. Configuración de Hibernate	100
6.1.7. Creando una clase Ayuda con Hibernate	106
6.1.8. Generando ficheros de mapeo Hibernate y Clases Java	107
6.1.9. Creando la clase de ayuda UsuarioAyuda.java	109
7. Implementación	113
8. Wiki de desarrollo	213
9. Manual de instalación	215
9.1. Diagrama de despliegue	215
9.2. Instalación y configuración de MySQL	215
9.3. Creación de la Base de Datos	225
9.4. Apache Tomcat 6.0.18	226

Índice de figuras

1.1. Miembros del equipo	5
1.2. Tickets - Asignación tareas Fase 0	8
1.3. Wiki - Primera reunión del equipo	9
1.4. svn - Fichero de prueba	10
1.5. NetBeans 6.0	11
1.6. Entorno de Desarrollo NetBeans 6.0	11
1.7. Primer paso, página http://tortoisesvn.tigris.org/	13
1.8. Página http://tortoisesvn.tigris.org/ "Descargas"	14
1.9. Primer paso en la instalación - Bienvenida	15
1.10. Segundo paso en la instalación - Control de versiones	16
1.11. Tercer paso en la instalación - Opción "SVN Checkout"	17
1.12. Cuarto paso en la instalación - Espacio URL of repository	18
1.13. Primer paso en la instalación MySQL	19
1.14. Segundo paso en la instalación - Tipo de instalación	20
1.15. Tercer paso en la instalación - ¿Asistente?	21
1.16. Primer paso en la configuración MySQL	22
1.17. Segundo paso en la configuración MySQL	23
1.18. Tercer paso en la configuración MySQL - Servidor	24
1.19. Cuarto paso en la configuración MySQL - Tipo de BD	25
1.20. Quinto paso en la configuración MySQL - Opciones de red	26
1.21. Sexto paso en la configuración MySQL	27
2.1. Clasificación de usuarios	30
2.2. Diagrama general de casos de uso	32
2.3. Diagrama de casos de uso de la administración de usuarios	33
2.4. Diagrama de casos de uso de la administración de páginas	34
2.5. Diagrama de casos de uso de la administración de etiquetas	35
2.6. Modelo Vista Controlador en JSF	45
2.7. MVC - Modelo	46
2.8. MVC - Controlador	47
3.1. Diagrama de despliegue	49
3.2. Diagrama de clases	50
3.3. Diagrama de navegabilidad	51
3.4. Diagrama Entidad/Relación	52
4.1. Diagrama Robustness Alta usuario	53
4.2. Diagrama Robustness Alta usuario por parte del administrador	54
4.3. Diagrama Robustness Listado de usuarios	54
4.4. Diagrama Robustness Listado de usuarios	55
4.5. Diagrama Robustness Listado de usuarios	55
4.6. Diagrama Robustness Crear página	56

4.7. Diagrama Robustness Modificar página	56
4.8. Diagrama Robustness Publicar/Despublicar página	57
4.9. Diagrama Robustness Visitar página	57
4.10. Diagrama Robustness Listado de páginas	58
4.11. Diagrama Robustness Etiquetar página	58
4.12. Diagrama Robustness Des-etiquetar página	59
4.13. Diagrama Robustness Listar etiquetas de una página	59
4.14. Diagrama Robustness Modificar etiqueta de una página	60
4.15. Diagrama de actividad login usuario	61
4.16. Diagrama de actividad listado usuarios	62
4.17. Diagrama de actividad alta usuario	63
4.18. Diagrama de actividad alta usuario por administrador	64
4.19. Diagrama de actividad modificar usuario	65
4.20. Diagrama de actividad modificar usuario por administrador	66
4.21. Diagrama de actividad crear página	67
4.22. Diagrama de actividad listar páginas	68
4.23. Diagrama de actividad visitar página	69
4.24. Diagrama de actividad editar página	70
4.25. Diagrama de actividad publicar página	71
4.26. Diagrama de actividad despublicar página	72
4.27. Diagrama de actividad etiquetar página	73
4.28. Diagrama de actividad des-etiquetar página	74
4.29. Diagrama de actividad listar etiquetas de una página	75
4.30. Diagrama de actividad publicar página administrador	76
4.31. Diagrama de actividad despublicar página administrador	77
4.32. Diagrama de actividad listar páginas administrador	78
4.33. Diagrama de actividad etiquetar página administrador	79
4.34. Diagrama de actividad des-etiquetar página administrador	80
4.35. Diagrama de actividad listar etiquetas de una página administrador	81
4.36. Diagrama de actividad modificar etiqueta de una página	82
4.37. Diagrama de actividad modificar etiqueta de una página	83
4.38. Pantalla de bienvenida a la Wiki	84
4.39. Registro en la wiki por parte de un usuario	84
4.40. Registro fallido por parte de un usuario	85
4.41. Registro correcto por parte de un usuario	85
4.42. Login por parte de un usuario	86
4.43. Login fallido por parte de un usuario	86
4.44. Página de inicio de un usuario registrado, cabecera	87
4.45. Página de inicio de un usuario registrado	87
4.46. Página de inicio del administrador	88
4.47. Modificación por parte del usuario fallida	88
4.48. Modificación por parte del usuario satisfactoria	89
4.49. Listado de los usuarios registrados	89
4.50. Listado de los usuarios registrados para modificar	90
4.51. Listado de los usuarios registrados, modificando datos	90
4.52. Listado de los usuarios registrados, eliminar usuario	91
6.1. Diagrama E/R de nuestra Base de Datos	95
6.2. Mapeo Objeto-Relacional	96
6.3. Configuración de Hibernate paso 1	101
6.4. Configuración de Hibernate paso 2	102
6.5. Configuración de Hibernate paso 3	103
6.6. Configuración de Hibernate paso 4	104
6.7. Configuración de Hibernate paso 5	105

6.8. Creando clase ayuda con Hibernate	106
6.9. Ficheros de mapeo Hibernate y Clases Java, paso 1	107
6.10. Ficheros de mapeo Hibernate y Clases Java, paso 2	108
6.11. Ficheros de mapeo Hibernate y Clases Java, paso 3	108
6.12. Creando la clase UsuarioAyuda.java, paso 1	110
6.13. Creando la clase UsuarioAyuda.java, paso 2	111
7.1. nicEditorIcons.gif	212
9.1. nicEditorIcons.gif	215
9.2. Primer paso en la instalación MySQL	216
9.3. Segundo paso en la instalación - Tipo de instalación	217
9.4. Tercer paso en la instalación - ¿Asistente?	218
9.5. Primer paso en la configuración MySQL	219
9.6. Segundo paso en la configuración MySQL	220
9.7. Tercer paso en la configuración MySQL - Servidor	221
9.8. Cuarto paso en la configuración MySQL - Tipo de BD	222
9.9. Quinto paso en la configuración MySQL - Opciones de red	223
9.10. Sexto paso en la configuración MySQL	224
9.11. Instalación Tomcat paso 1	227
9.12. Instalación Tomcat paso 2	228
9.13. Instalación Tomcat paso 3	229
9.14. Instalación Tomcat paso 4	230
9.15. Instalación Tomcat paso 5	231
9.16. Instalación Tomcat paso 6	232
9.17. Instalación Tomcat paso 7	233
9.18. Instalación Tomcat paso 8	234
9.19. Instalación Tomcat paso 9	235

1

Planificación y configuración

Capítulo

1.1. Descripción de planificación de tiempos de desarrollo

1.1.1. Recursos humanos y roles

Nuestro equipo está formado por cuatro personas que desempeñan los siguientes roles:

- Sergio Puerta Aragón: Analista, Administrador de la Base de Datos, Administrador de Sistemas
- Lorena Gutiérrez Madroñal: Diseñador de Interfaz de Usuario, Administrador de Sistemas, Analista de Sistemas
- Jose Alberto Cordero Ríos: Programador, Codificador de Pruebas, Arquitecto de Software
- Félix Reyes Fernández: Jefe de Proyecto, Programador, Arquitecto de Software

Estos roles podrían ir rotando, según las necesidades del Proyecto, y circunstancias personales de cada integrante del equipo.

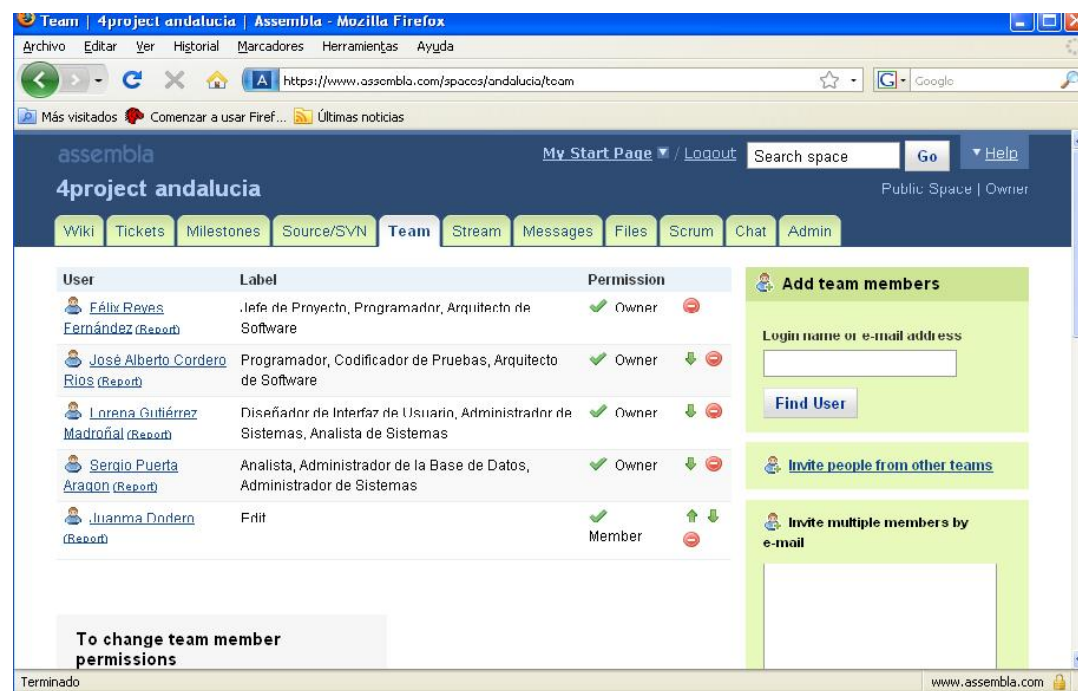


Figura 1.1: Miembros del equipo

1.1.2. Fechas para los entregables

Intentaremos cumplir con las fechas de cada una de las fases para no retrasar el proyecto final. Para ello, cada uno de los integrantes del equipo se comprometerá a realizar sus tareas dentro del plazo asignado.

- Fase 0 - Planificación y configuración - 20 Abril
- Fase 1 - Especificación de requisitos y análisis - 27 Abril
- Fase 2 - Administración de usuarios - 6 Mayo
- Fase 3 - Administración de páginas - 13 Mayo
- Fase 4 - Seguridad - 13 Mayo
- Fase 5 - Servicio web de publicación - 20 Mayo
- Fase 6 - Administración de etiquetas - 27 Mayo
- Fase 7 - Servicio web de etiquetado - 3 Junio

1.1.3. Plazos e hitos

Para cada fase, dividiremos el trabajo a entregar, en distintos apartados, y cada uno de los integrantes se dedicará a desarrollar su parte. Para ello, usaremos los tickets de nuestro espacio en asamblea, donde controlaremos las tareas asignadas a cada miembro del equipo.

En la wiki de nuestro espacio, habilitaremos una página por fase, y en cada página iremos editando la descripción de las tareas que ha ido realizando cada miembro del equipo.

Para esta primera fase (Fase 0), cuya entrega finaliza el Lunes, ya estamos trabajando en las distintas tareas que han sido asignadas a cada miembro del equipo.

Estas son las tareas actuales de esta Fase 0:

1. Preparar configuración: Sergio Puerta Aragon
2. Búsqueda de información de herramientas: José Alberto Cordero Ríos
3. Realizar el entregable E1: Lorena Gutiérrez Madroñal
4. Configuración herramientas, redacción documento: Félix Reyes Fernández

1.1.4. Reuniones de Proyecto

Las reuniones se harán de manera semanal en las clases correspondientes a la asignatura, es decir, Miércoles de 18:00 a 21:00.

Si fuesen necesarias más reuniones, concensuaremos lugar y hora entre todos, y también organizaremos reuniones de tipo virtual, a través de chats. Para organizar estas reuniones, se utilizará la Wiki del espacio web.

1.1.5. Elección de tecnologías de desarrollo y frameworks

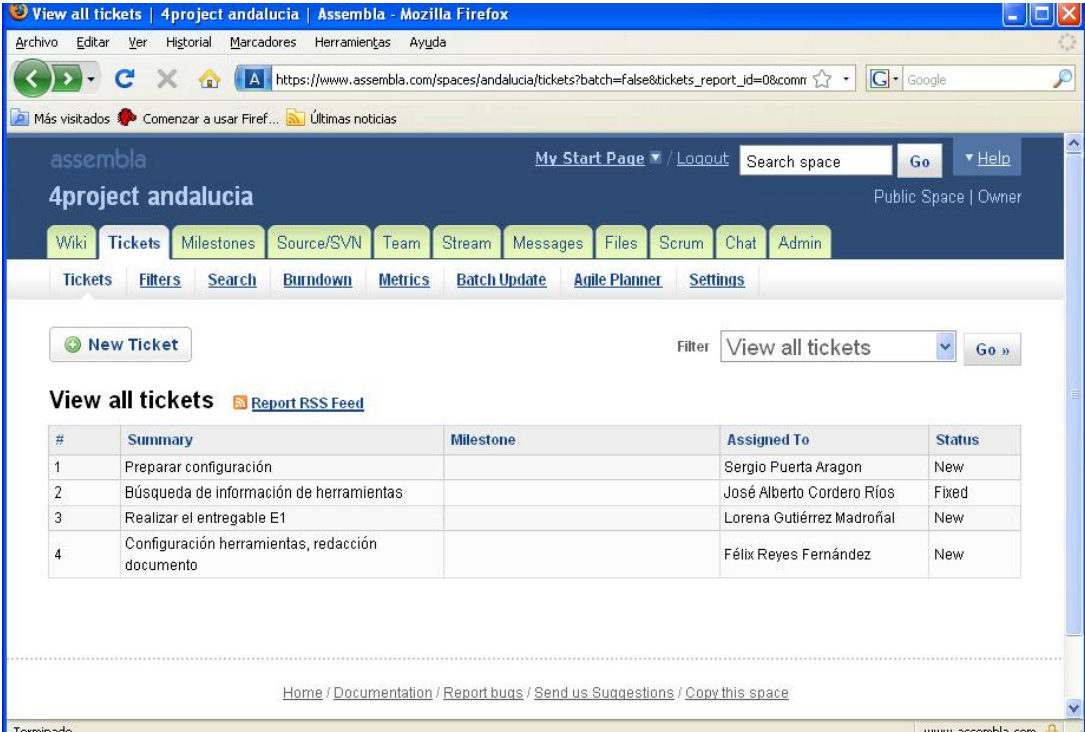
Vamos a trabajar con el framework JSF, y nuestro entorno de desarrollo será NetBeans. Hay otras herramientas que aún no sabemos si vamos a utilizarlas porque nos tenemos que documentar. Conforme vayamos desarrollando el Proyecto, iremos completando este apartado.

¿Por qué elegimos JSF? JSF nos proporciona las siguientes ventajas:

- Ofrece una clara separación entre el comportamiento y la presentación.
- JSF se hace fácil de usar al aislar al desarrollador del API de Servlet.
- La separación de la lógica de la presentación también le permite a cada miembro del equipo de desarrollo de una aplicación Web enfocarse en su parte del proceso de desarrollo.
- JSF ofrece una gran cantidad de componentes opensource para las funcionalidades que se necesiten.
- Además, también existe una gran cantidad de herramientas para el desarrollo IDE en JSF al ser el estándar de JAVA.
- La tecnología JavaServer Faces proporciona una rica arquitectura para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, y manejar eventos.
- Además, ofrece una rápida adaptación para nuevos desarrolladores.

1.2. Demostración configuración entorno de desarrollo y herramientas de desarrollo, Configuración del sitio web de alojamiento del proyecto

1.2.1. Tickets (Asignación tareas Fase 0)



The screenshot shows the Assembla web interface in a Mozilla Firefox browser. The page title is "View all tickets | 4project andalucia | Assembla - Mozilla Firefox". The browser's address bar shows the URL: https://www.assembla.com/spaces/andalucia/tickets?batch=false&tickets_report_id=0&comin. The Assembla header includes the logo, "My Start Page", "Logout", a search bar, and a "Go" button. Below the header, there are tabs for "Wiki", "Tickets", "Milestones", "Source/SVN", "Team", "Stream", "Messages", "Files", "Scrum", "Chat", and "Admin". The "Tickets" tab is selected. Below the tabs, there are links for "Tickets", "Filters", "Search", "Burndown", "Metrics", "Batch Update", "Agile Planner", and "Settings". A "New Ticket" button is visible. A filter dropdown is set to "View all tickets" with a "Go" button. The main content area shows a table of tickets with the following data:

#	Summary	Milestone	Assigned To	Status
1	Preparar configuración		Sergio Puerta Aragon	New
2	Búsqueda de información de herramientas		José Alberto Cordero Ríos	Fixed
3	Realizar el entregable E1		Lorena Gutiérrez Madroñal	New
4	Configuración herramientas, redacción documento		Félix Reyes Fernández	New

At the bottom of the page, there are links for "Home", "Documentation", "Report bugs", "Send us Suggestions", and "Copy this space".

Figura 1.2: Tickets - Asignación tareas Fase 0

1.2.2. Wiki (Podemos ver una primera página editada con la primera reunión del equipo)



Figura 1.3: Wiki - Primera reunión del equipo

1.2.3. SVN (Con un fichero subido de prueba)

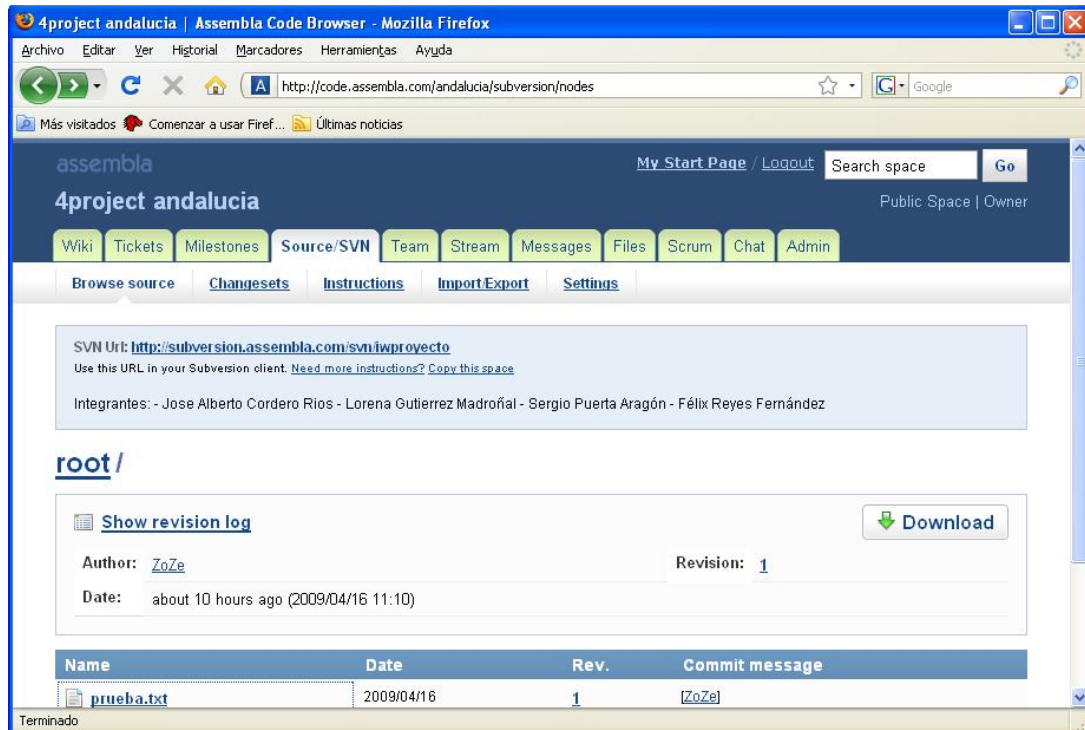


Figura 1.4: svn - Fichero de prueba

1.2.4. Entorno de Desarrollo NetBeans 6.0



Figura 1.5: NetBeans 6.0

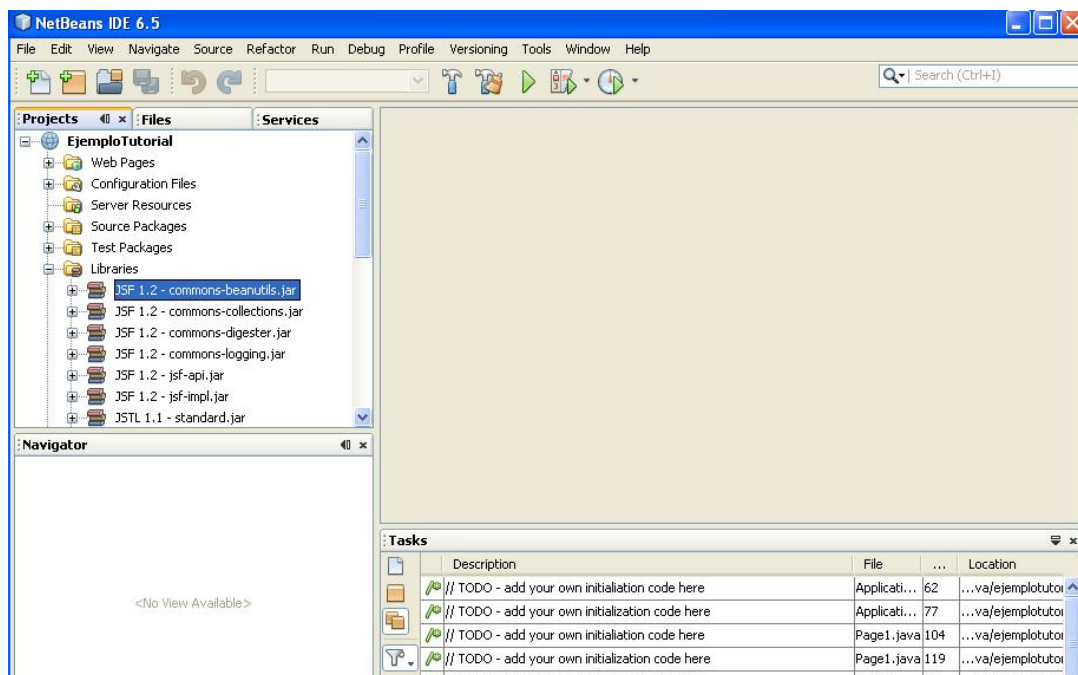


Figura 1.6: Entorno de Desarrollo NetBeans 6.0

1.3. Posibles herramientas a aplicar

1.3.1. Ant

La herramienta Ant, sirve para conseguir de una forma sencilla hacer una serie de tareas tediosas pero necesarias de una forma sencilla. Principalmente estamos hablando de tareas de construcción de proyectos, borrado de archivos .class, generación de documentación con Javadoc...

Esta herramienta utiliza tecnología XML y trabaja a modo de make pero en un aspecto bastante más amplio. Sin embargo creo que al utilizar NetBeans, tenemos una forma sencilla de construir y limpiar nuestro proyecto, así que no lo considero 100 % necesario.

1.3.2. JUnit

La herramienta JUnit sirve para hacer pruebas automáticas de nuestro programa, contiene una serie de librerías que permiten ejecutar pruebas sobre ciertas clases de manera controlada. Contiene muchas características interesantes y puede que debiéramos intentar echarle un vistazo más en profundidad, no parece muy difícil de configurar, pero requiere un esfuerzo adicional de programación la preparación de todas estas clases, también es verdad que nuestro proyecto va a ser bastante pequeño así que no se si compensará, aprender algo desde cero para aplicarlo aquí.

1.3.3. Log4j

Por último, pero no menos importante, tenemos la herramienta Log4j que sirve para realizar un log de nuestro servidor. Lo más interesante es que son ficheros externos y podemos modificarlos según dice por aquí en tiempo de ejecución al ser dinámico y en formato XML. Se pueden enviar los datos de los log a varios archivos establecidos o dirigirlo donde más nos convenga.

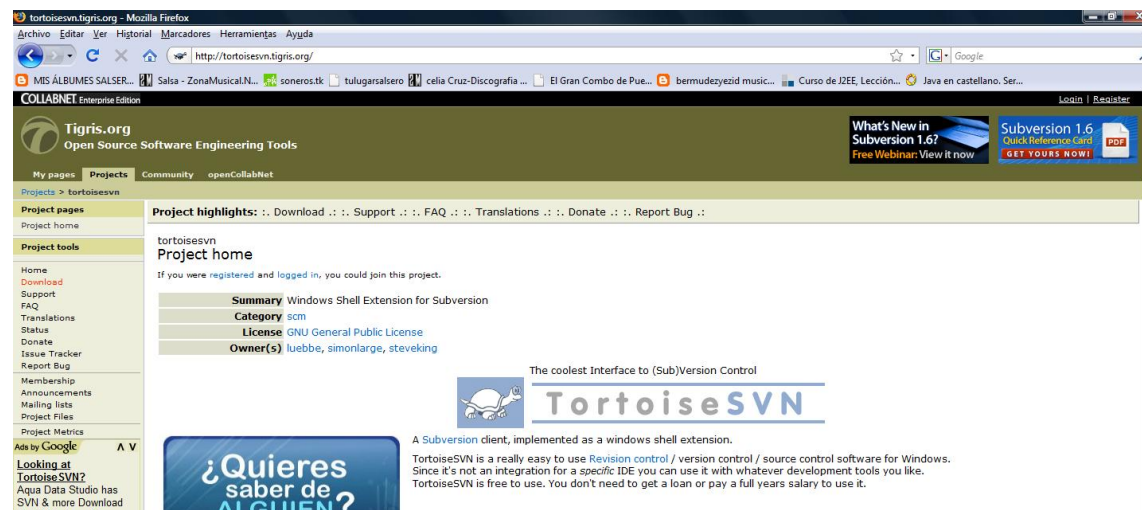
Tiene bastante variedad a la hora de dar los logs y parece bastante completo se podría intentar echarle un vistazo a ver qué tal, de momento es el único que he visto que merezca la pena.

1.4. Instalación y configuración de la herramienta TortoiseSVN

Como sistema de control de versiones, vamos a utilizar Subversion. Pero en vez de usar Subversion desde la línea de comandos, vamos a usar un cliente de Subversion, implementado como un shell de Windows. En concreto, vamos a utilizar TortoiseSVN.

Para ello, tenemos que seguir los siguientes pasos:

1. En primer lugar, descargarnos la instalación del programa. Para ello, vamos a la siguiente página: <http://tortoisesvn.tigris.org/>

Figura 1.7: Primer paso, página <http://tortoisesvn.tigris.org/>

2. En la parte izquierda, tenemos la sección de descargas (Downloads). Al pulsar llegaremos a la siguiente página:

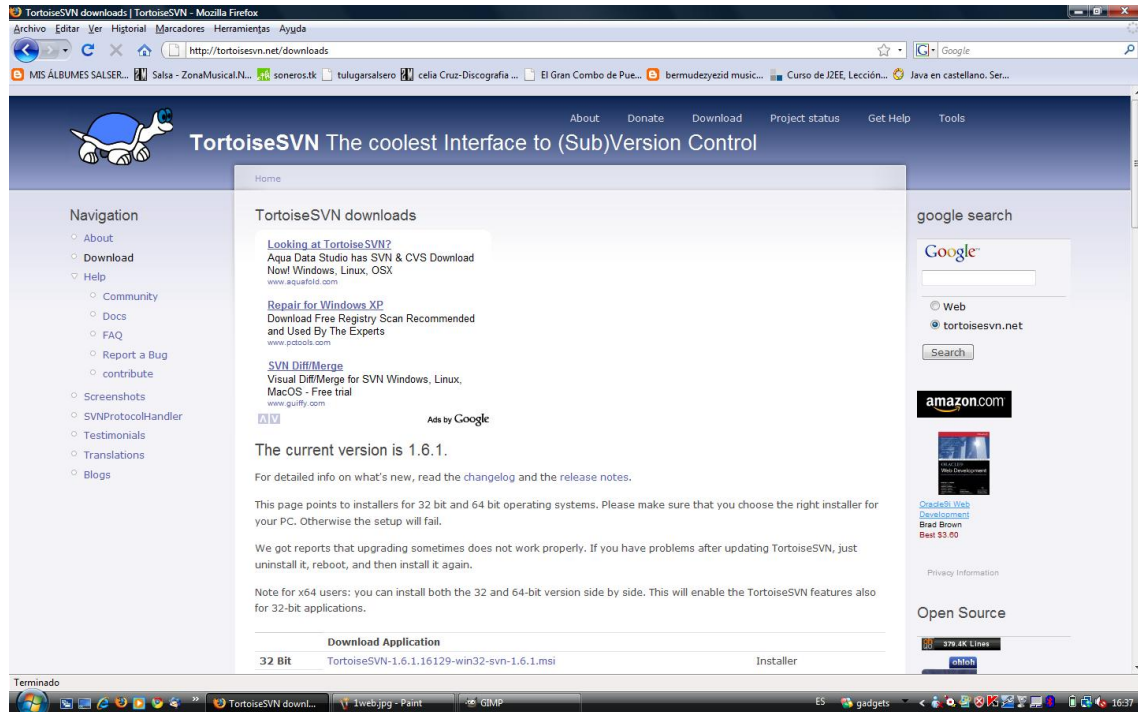


Figura 1.8: Página <http://tortoisesvn.tigris.org/> “Descargas”

Dependiendo de nuestro sistema operativo, deberemos descargar una u otra versión. En nuestro caso hemos descargado el siguiente paquete: TortoiseSVN-1.6.1.16129-win32-svn-1.6.1.msi

3. Una vez descargado el programa, procederemos a su instalación. Para ello, ejecutamos el paquete que previamente nos hemos descargado, apareciéndonos la siguiente pantalla de bienvenida:



Figura 1.9: Primer paso en la instalación - Bienvenida

A partir de este punto, debemos instalar el programa con las opciones por defecto.

4. Cuando hayamos terminado de instalar el programa en nuestra máquina, tenemos que configurarlo para que use el espacio de control de versiones que nos ofrece assembla.

Como resultado de la instalación, vemos que el programa ha creado un grupo de menús en el menú contextual de Windows. La forma más rápida y efectiva de configurar el programa para usar el espacio de control de versiones web es el siguiente. Una de las opciones del menú contextual es "SVN Checkout". Seleccionamos la carpeta en la que queremos que se controle las versiones de documentos con el botón derecho:

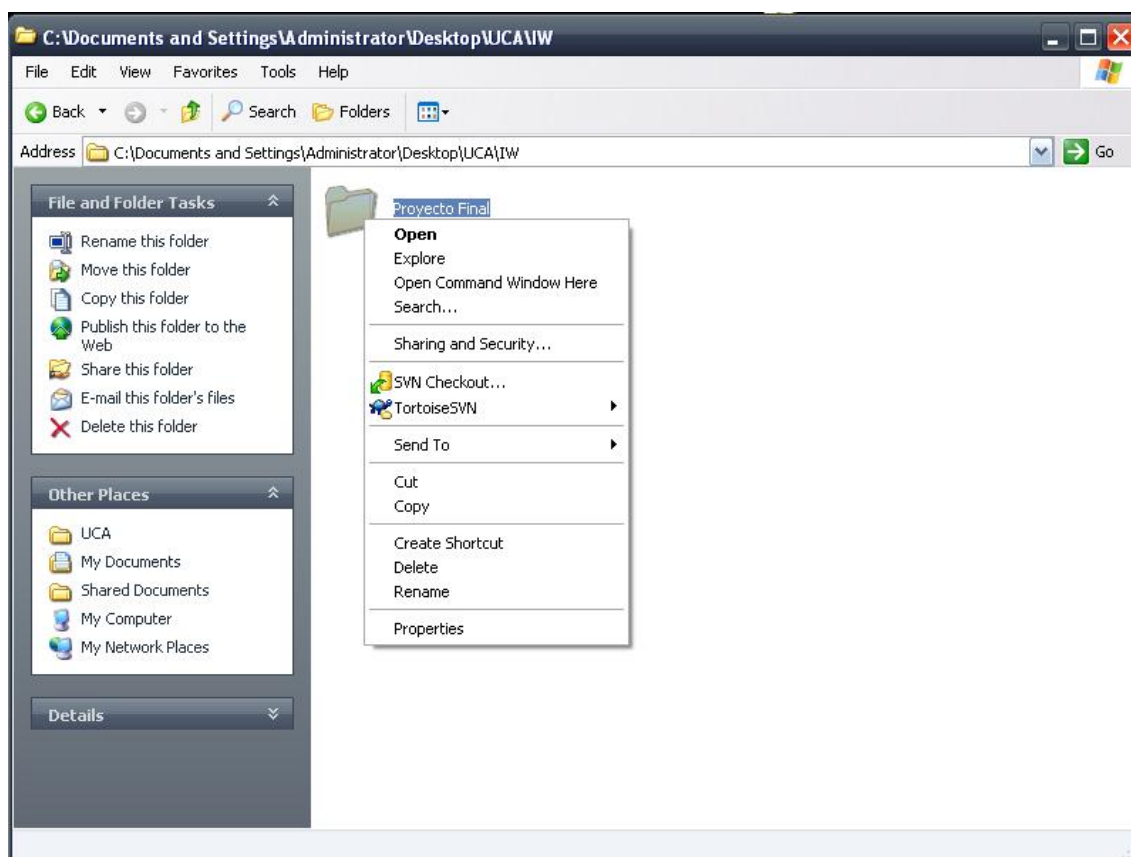


Figura 1.10: Segundo paso en la instalación - Control de versiones

Seleccionamos la opción "SVN Checkout"

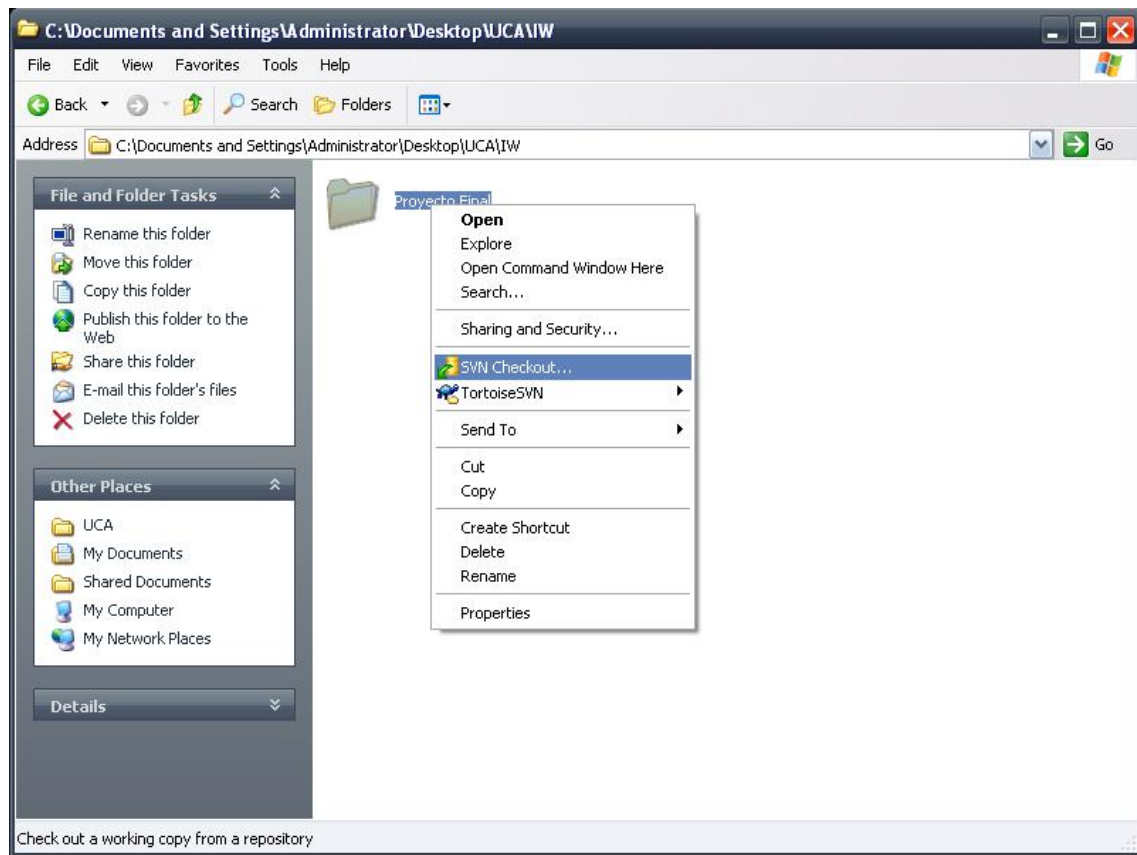


Figura 1.11: Tercer paso en la instalación - Opción "SVN Checkout"

Nos aparecerá el siguiente menú, en el que debemos escribir en el espacio “URL of repository” la URL del espacio de control de versiones de asamblea.

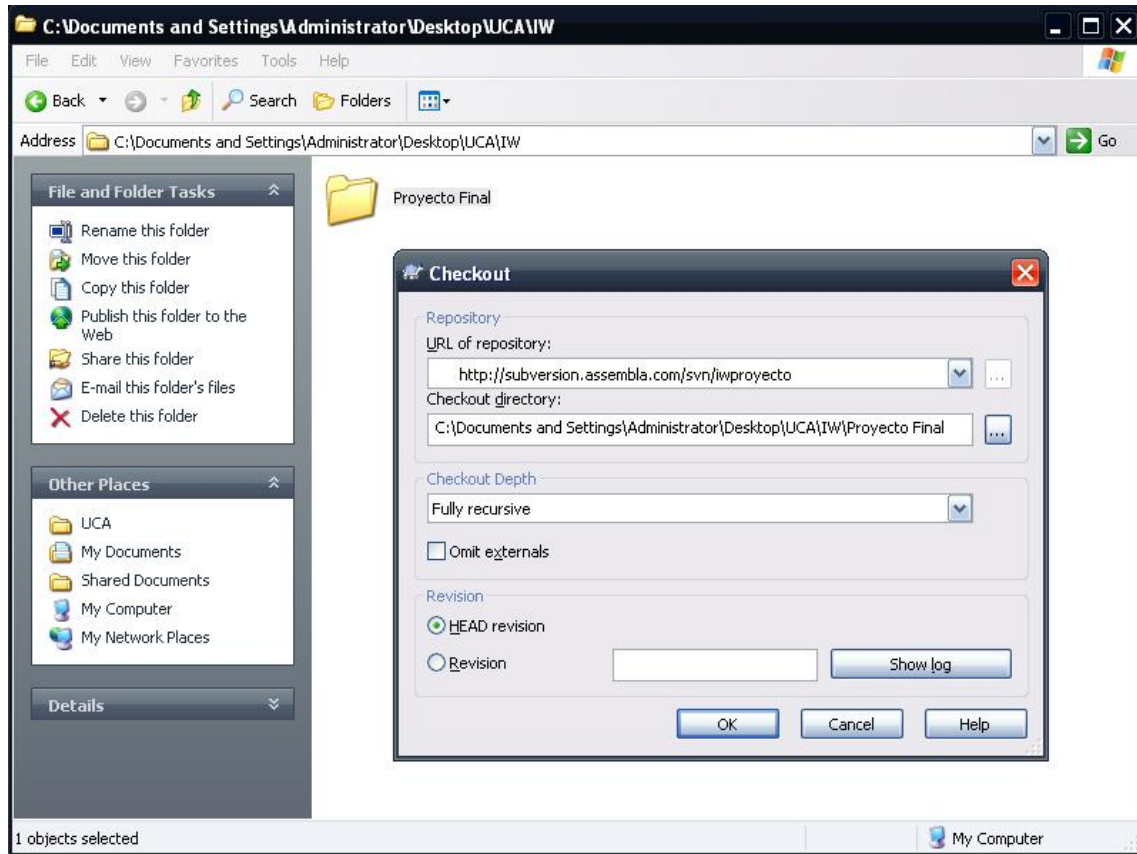


Figura 1.12: Cuarto paso en la instalación - Espacio URL of repository

1.5. Instalación y configuración de MySQL

Como sistema de gestión de bases de datos, vamos a usar MySQL, ya que es gratuito y potente. Para instalar y configurar nuestro SGBD, vamos a seguir los siguientes pasos:

1. En primer lugar, debemos descargarnos el programa de instalación. Para ello, vamos a la página <http://www.mysql.com/> y entramos en la sección de descargas (Downloads). La versión que vamos a usar es plenamente funcional y completa, además de gratuita. Seleccionamos la versión MySQL Community Server.
2. Una vez descargado el archivo, comenzamos con su instalación. Al ejecutarlo, nos aparecerá la siguiente pantalla:



Figura 1.13: Primer paso en la instalación MySQL

Seleccionamos el tipo de instalación que queremos hacer, en nuestro caso Custom para configurar las opciones que queremos.

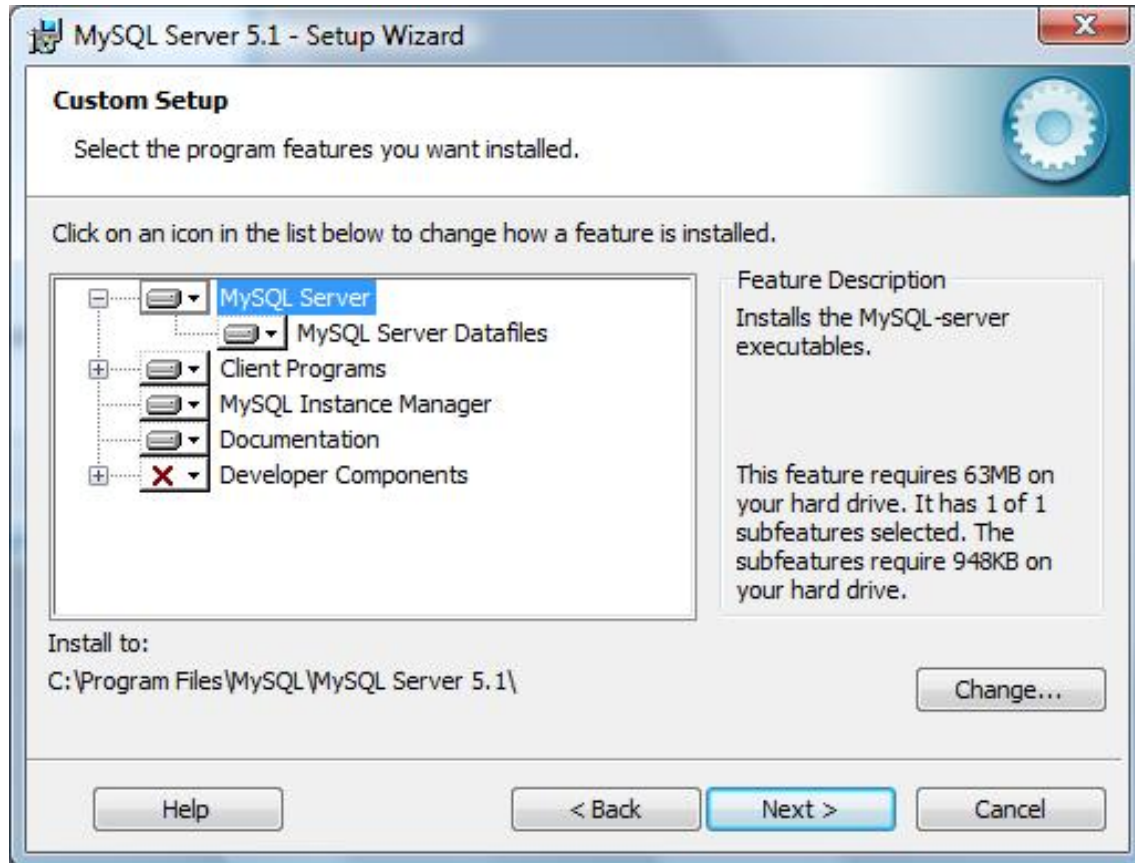


Figura 1.14: Segundo paso en la instalación - Tipo de instalación

Después de configurarlo, instalamos el programa, pero en la última ventana antes de salir del asistente, nos dan la opción de que sea el propio asistente quien configure la instancia de MySQL con las opciones por defecto. En nuestro caso, desactivamos esta casilla y lo instalamos nosotros mismos.

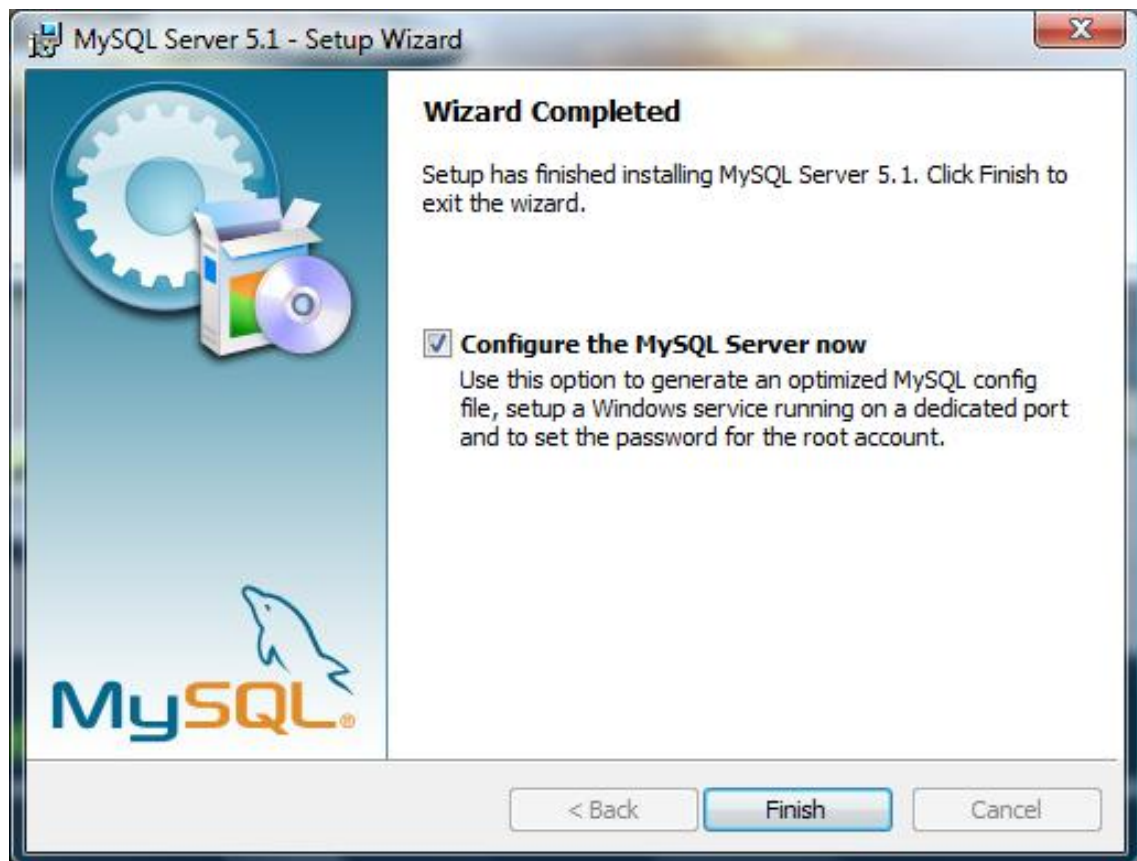


Figura 1.15: Tercer paso en la instalación - ¿Asistente?

3. A continuación vamos a configurar la instancia de MySQL. Nos vamos al grupo de programa creado, e iniciamos "MySQL Server Instance Config Wizard".



Figura 1.16: Primer paso en la configuración MySQL

4. El siguiente paso es escoger entre la configuración Standard o Detailed. La primera es una configuración general, mientras que la segunda nos permite configurar el SGBD a nuestro gusto. Optamos por esta segunda opción.

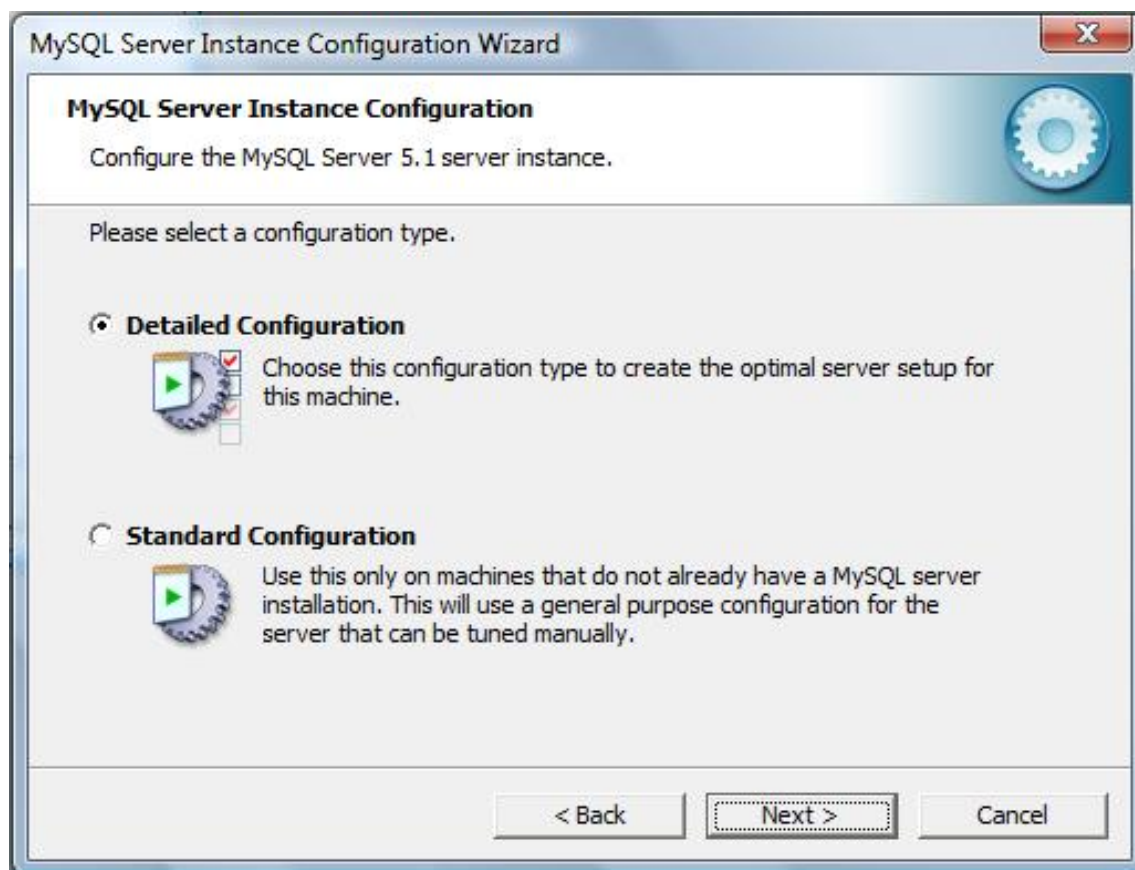


Figura 1.17: Segundo paso en la configuración MySQL

5. A continuación se nos pregunta qué tipo de servidor vamos a tener. Entre las opciones que nos ofrece, optamos por "Developer Machine", suficiente para nuestro caso.



Figura 1.18: Tercer paso en la configuración MySQL - Servidor

6. El siguiente punto es escoger el tipo de base de datos a usar. En este caso, vamos a optar por una base de datos Transaccional únicamente (Transactional Database only), ya que nos proporcionará la funcionalidad suficiente para nuestro proyecto.

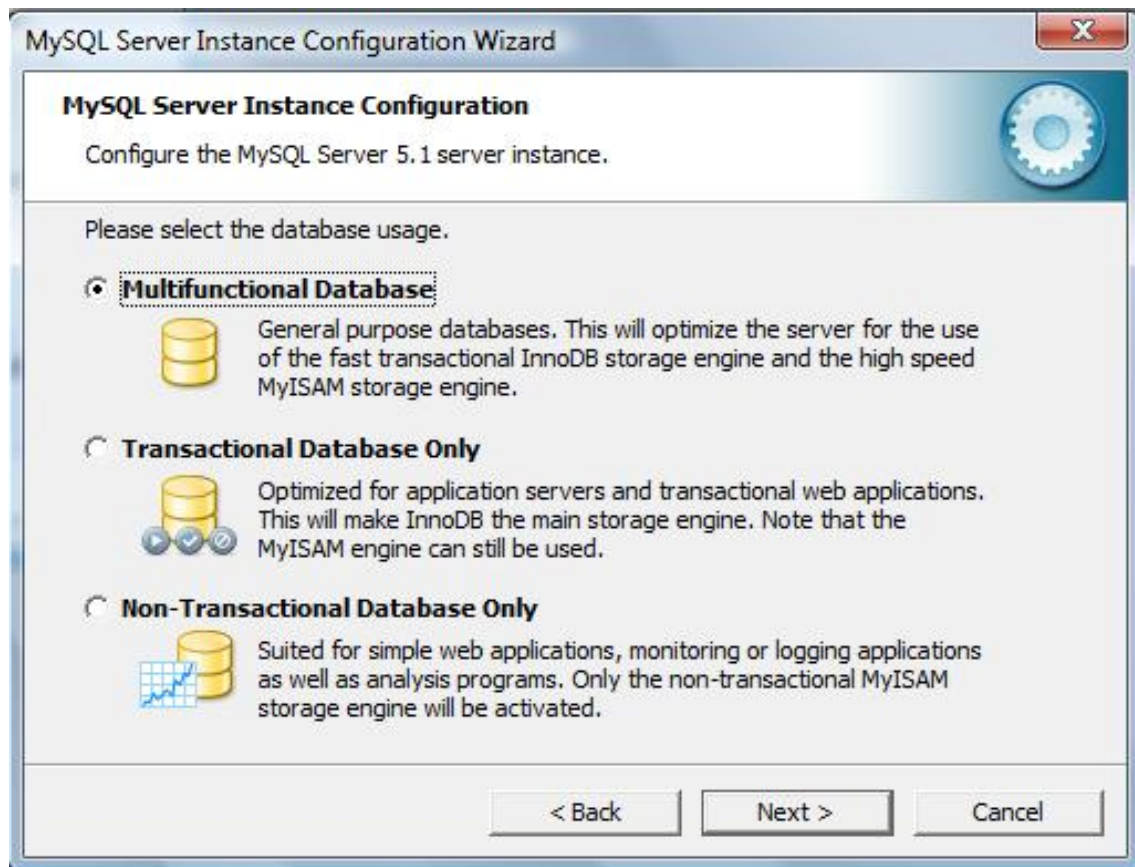


Figura 1.19: Cuarto paso en la configuración MySQL - Tipo de BD

7. La siguiente pantalla nos pregunta por las opciones de la red: si vamos a usar conexiones externas o únicamente conexiones internas desde nuestra máquina. En nuestro caso, optamos por conexiones externas. Seleccionamos el puerto por el que nos vamos a conectar y le indicamos que cree una excepción para el firewall. También dejamos marcado la opción "Enable Strict Mode", que ofrece más funcionalidad que un servidor de base de datos solamente.

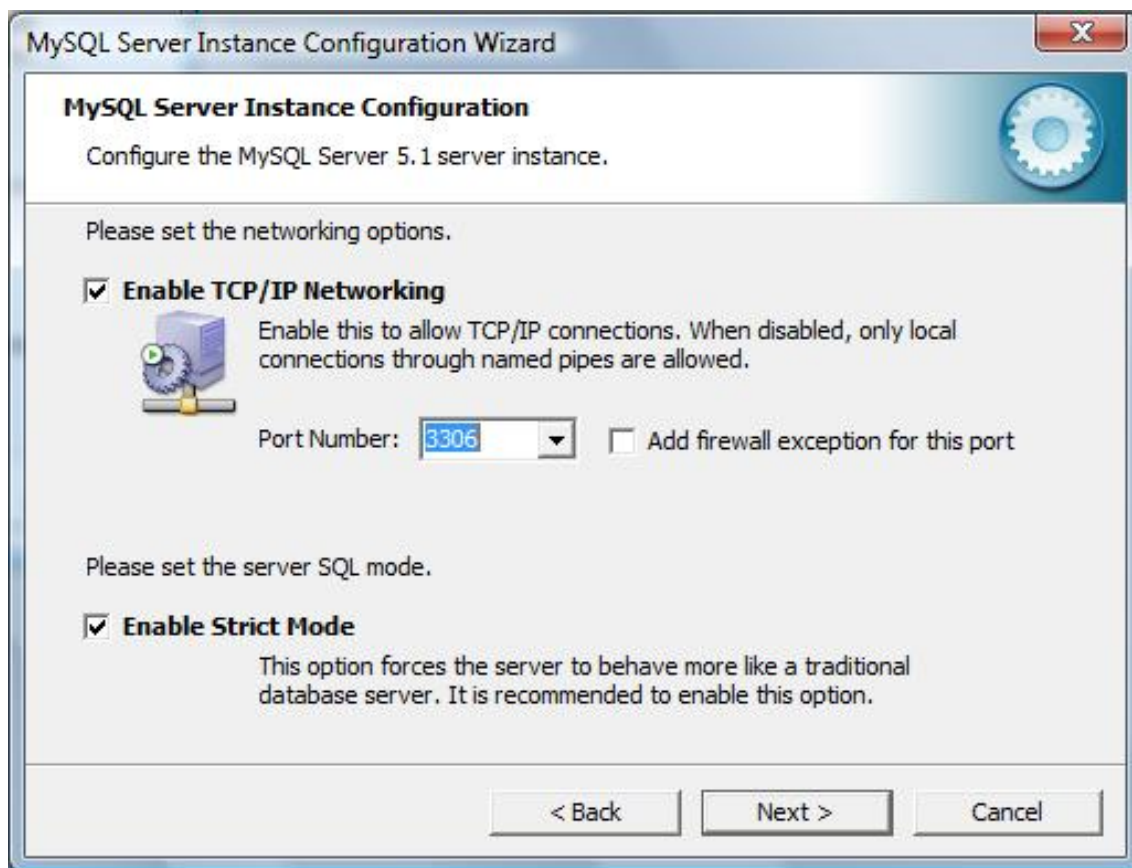


Figura 1.20: Quinto paso en la configuración MySQL - Opciones de red

8. A continuación le indicamos que instale el servicio del sistema MySQL, que lo inicie automáticamente cuando encendamos la máquina, y que cree incluya en la variable PATH el directorio de ejecución del servicio para poder llamarlo desde la consola del sistema.

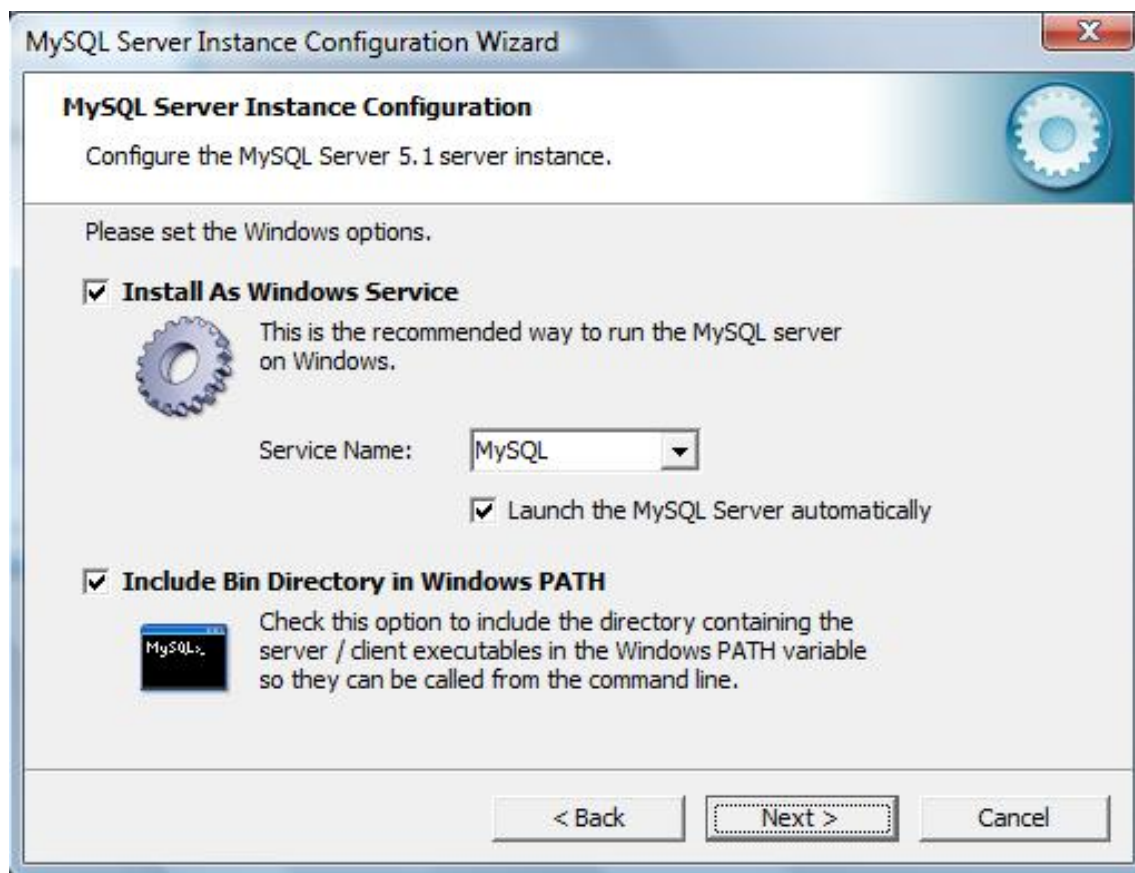


Figura 1.21: Sexto paso en la configuración MySQL

9. A continuación le indicamos que instale el servicio del sistema MySQL, que lo inicie automáticamente cuando encendamos la máquina, y que cree incluya en la variable PATH el directorio de ejecución del servicio para poder llamarlo desde la consola del sistema.
10. A continuación tendremos que introducir una contraseña para el usuario root. Y finalmente, si todo ha ido bien, en la siguiente pantalla ejecutará todas las opciones que hemos indicado y configurará nuestro sistema.

2

Capítulo

Requisitos y Arquitectura

2.1. Especificación de Requisitos y Casos de Uso del Sistema

Como fase previa a la especificación de requisitos y casos de uso del sistema, desarrollamos un Modelado de Usuarios utilizando el método WSDM, cuya documentación hemos visto en el Campus Virtual de la asignatura.

2.1.1. ¿Qué es WSDM?

Es un Método de Diseño para Sitios Web (Web Site Design Method), donde hay un acercamiento al usuario que define los objetos de información basado en sus requisitos de información para el uso de la Web. En este método se definen una aplicación Web a partir de los diferentes grupos de usuarios que vaya a reconocer el sistema.

Propone cuatro etapas: modelo de usuario, diseño conceptual, diseño de la implementación e implementación. El tratamiento de requisitos se lleva a cabo en la etapa inicial, donde, en primer lugar, se identifican y clasifican los usuarios que van a hacer uso de la aplicación Web. A continuación, se describen los requisitos de cada grupo de usuarios y sus fases.

Nosotros vamos a utilizar este método únicamente para la Fase de Modelo de Usuario.

Fase de Modelo de Usuario, se intenta detectar los perfiles de usuarios para los cuales se construye la aplicación.

- Clasificación de usuarios: Se deben identificar y clasificar a los usuarios que van a hacer uso del sistema. Para ello, WSDM propone el estudio del entorno de la organización donde se vaya a implantar el sistema y los procesos que se vayan a generar, describiendo las relaciones entre usuarios y actividades que realizan estos usuarios.

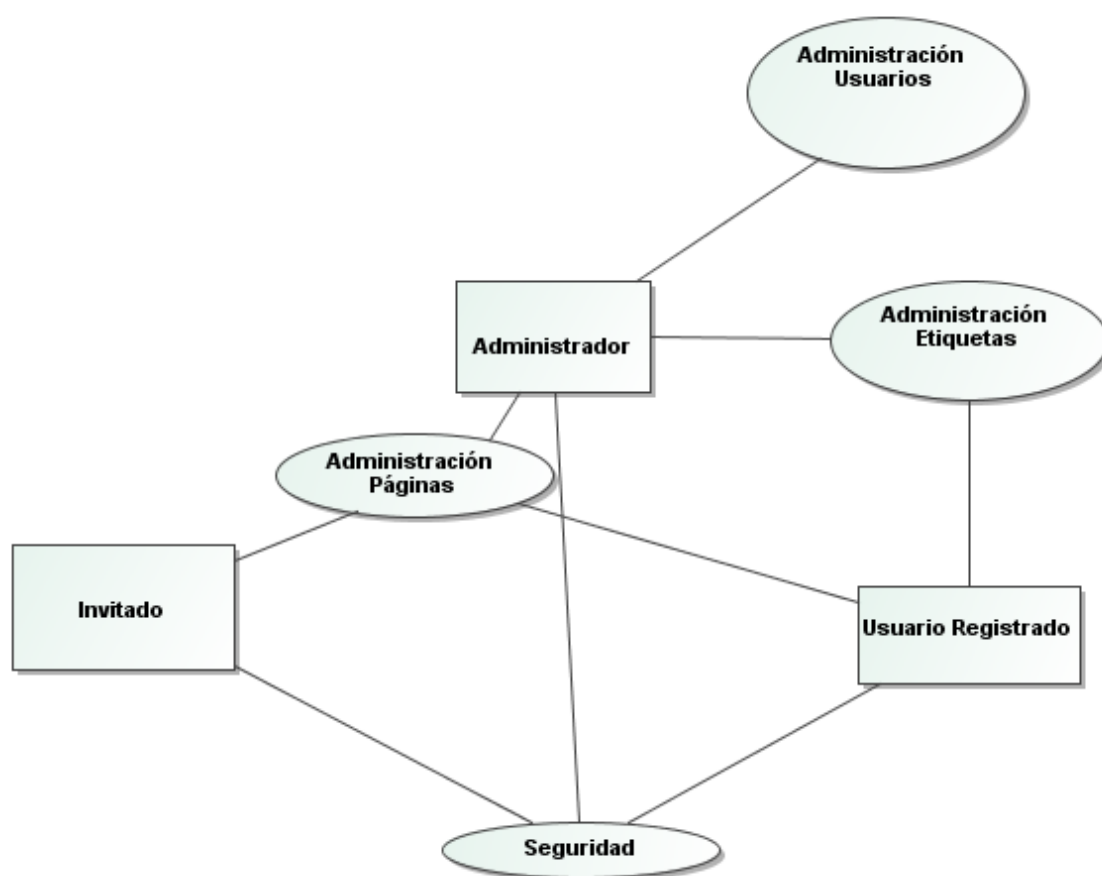


Figura 2.1: Clasificación de usuarios

- Descripción de los grupos de usuarios: Se describen con más detalles los grupos de usuarios detectados en la etapa anterior. Para ello, se debe elaborar un diccionario de datos, en principio con formato libre, en el que indican los requisitos de almacenamiento de información, requisitos funcionales y de seguridad para cada grupo de usuarios.

Administrador:

1. Administración de usuarios: Altas, bajas, modificaciones y listados de usuarios registrados en el sistema.
2. Administración de etiquetas: Crear, borrar y listar etiquetas, etiquetar y desetiquetar páginas.
3. Seguridad: Inicio de sesión en la aplicación.
4. Administración de páginas: Ver página, listar páginas, crear/editar páginas, publicar página, despublicar página.

Usuario Registrado:

1. Administración de usuarios:
2. Administración de etiquetas: Crear, borrar y listar etiquetas, etiquetar y desetiquetar páginas.
3. Seguridad: Inicio de sesión en la aplicación.
4. Administración de páginas: Ver página, listar páginas, crear/editar páginas, publicar página, despublicar página.

Invitado:

1. Administración de etiquetas:
2. Seguridad: Registrarse para el acceso a las páginas.
3. Administración de páginas: Consultar página.
4. Administración de usuarios:

Ahora desarrollamos el diagrama de Casos de Uso y la descripción de cada Caso de Uso.

Tenemos 2 niveles de abstracción dentro de nuestros diagramas de Casos de Uso, el primero nivel, más general, contempla a grandes rasgos las funciones generales que pueden realizar los distintos actores, en nuestro caso:

- Administración de Páginas
- Administración de Usuarios
- Administración de Etiquetas
- Registro de Usuarios
- Inicio de sesión

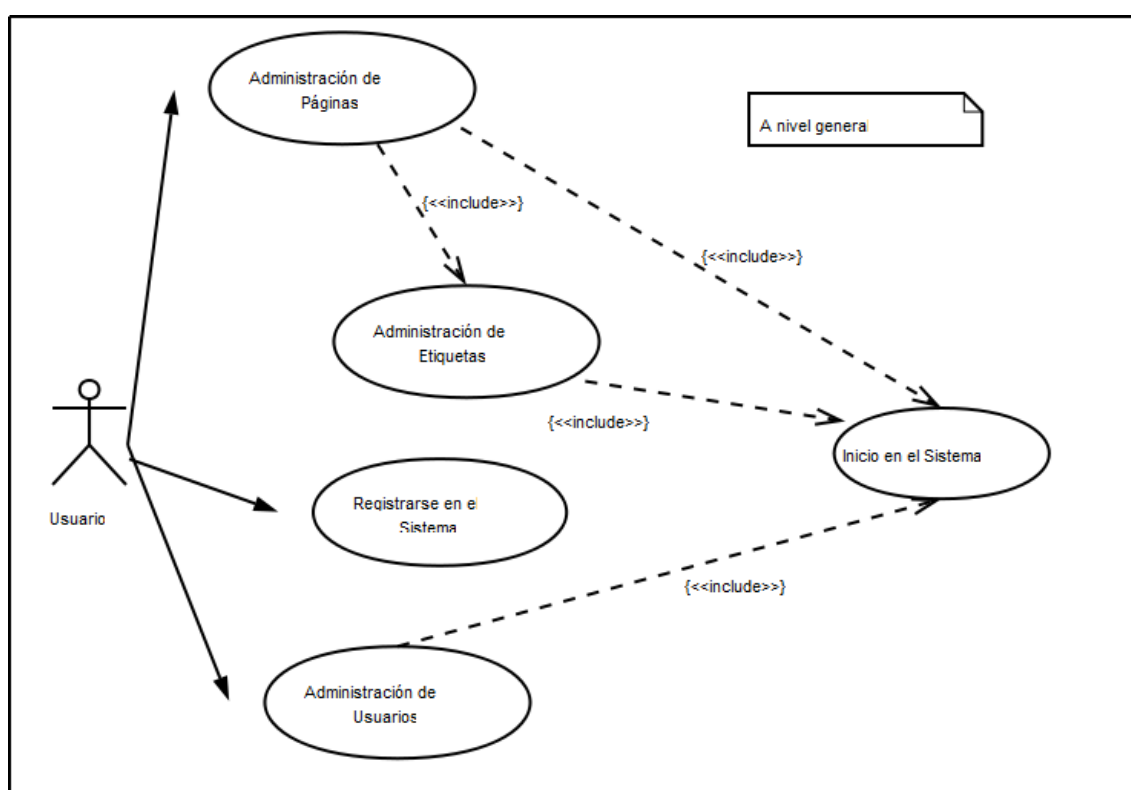


Figura 2.2: Diagrama general de casos de uso

El siguiente diagrama de Casos de Uso corresponde a Administración de Usuarios. A este nivel nos interesa especializar los usuarios en los tipos de usuarios que existirán en nuestro sistema para poder ver con mayor claridad las funciones que van a desempeñar cada uno de los roles. Estos roles son:

- Administrador del Sistema
- Usuario Registrado
- Visitante

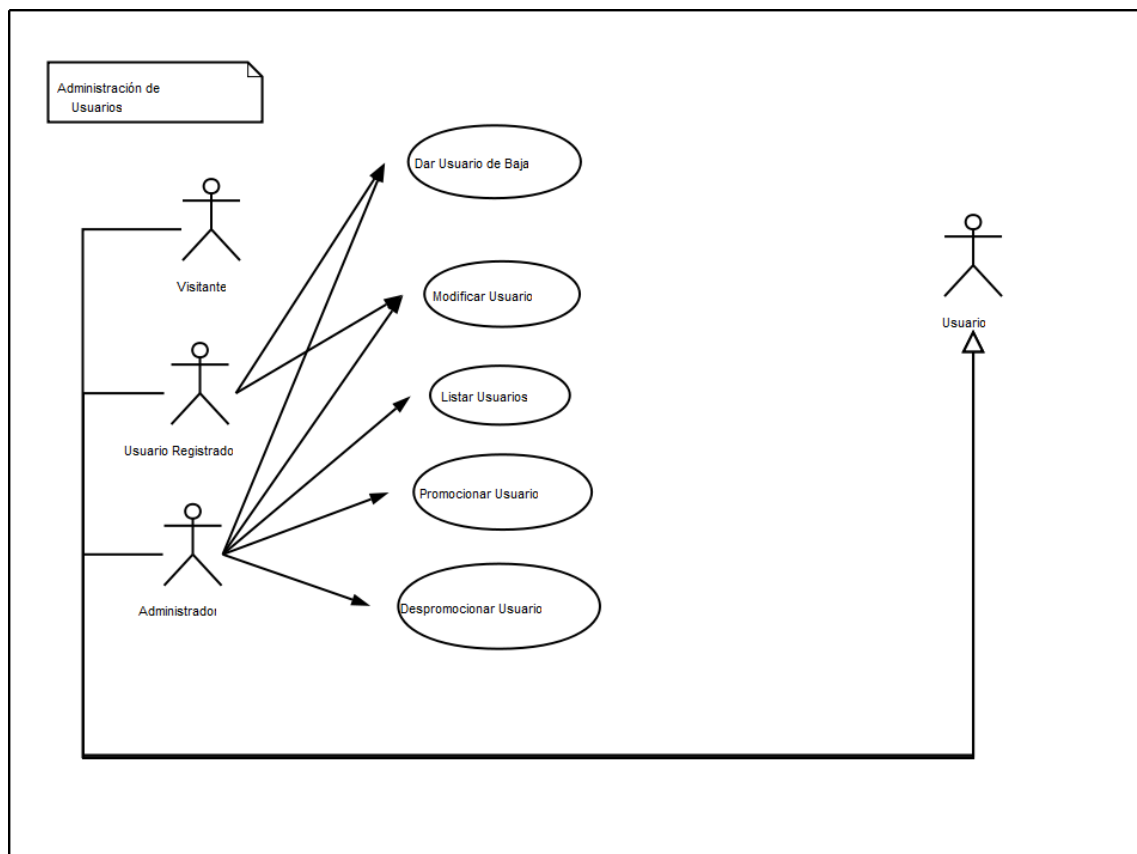


Figura 2.3: Diagrama de casos de uso de la administración de usuarios

Los siguientes diagramas de uso siguen la misma filosofía a la hora de especializar los usuarios, pero la diferencia principal es la función que desempeñan:

- Administración de Páginas
- Administración de Etiquetas

Presentamos los diagramas respectivamente del listado anterior.

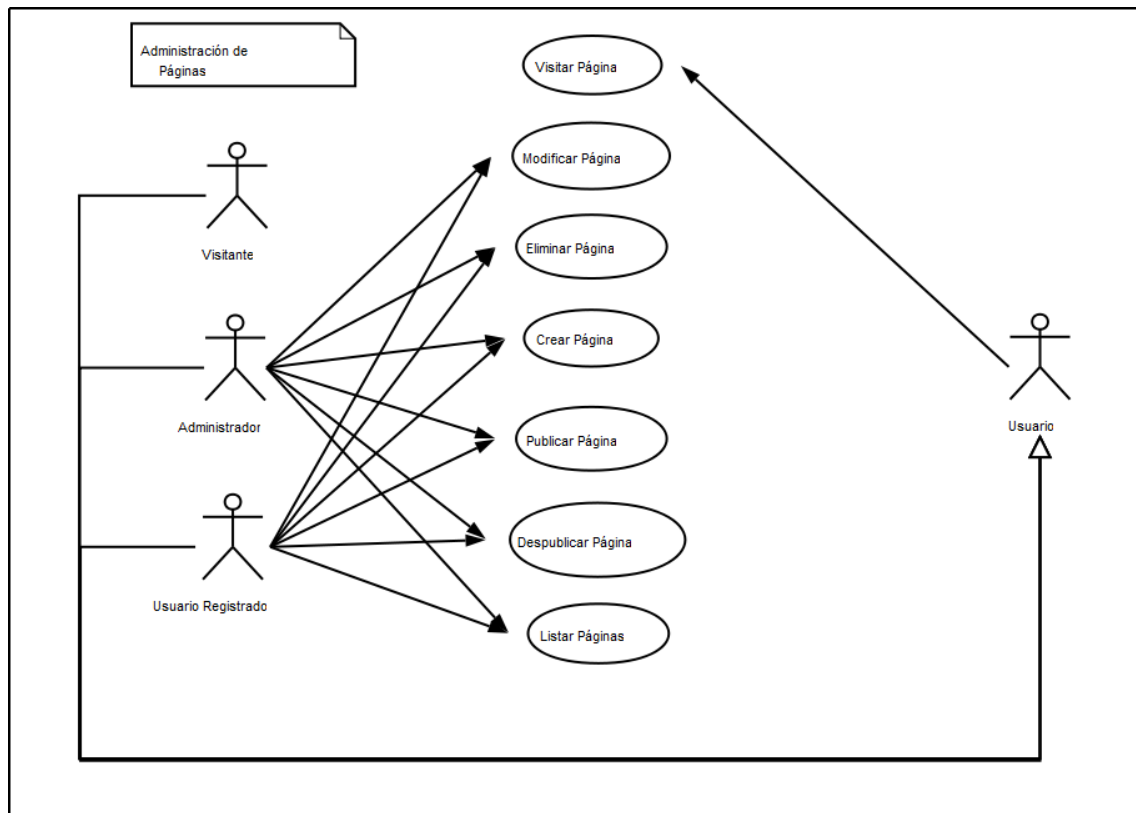


Figura 2.4: Diagrama de casos de uso de la administración de páginas

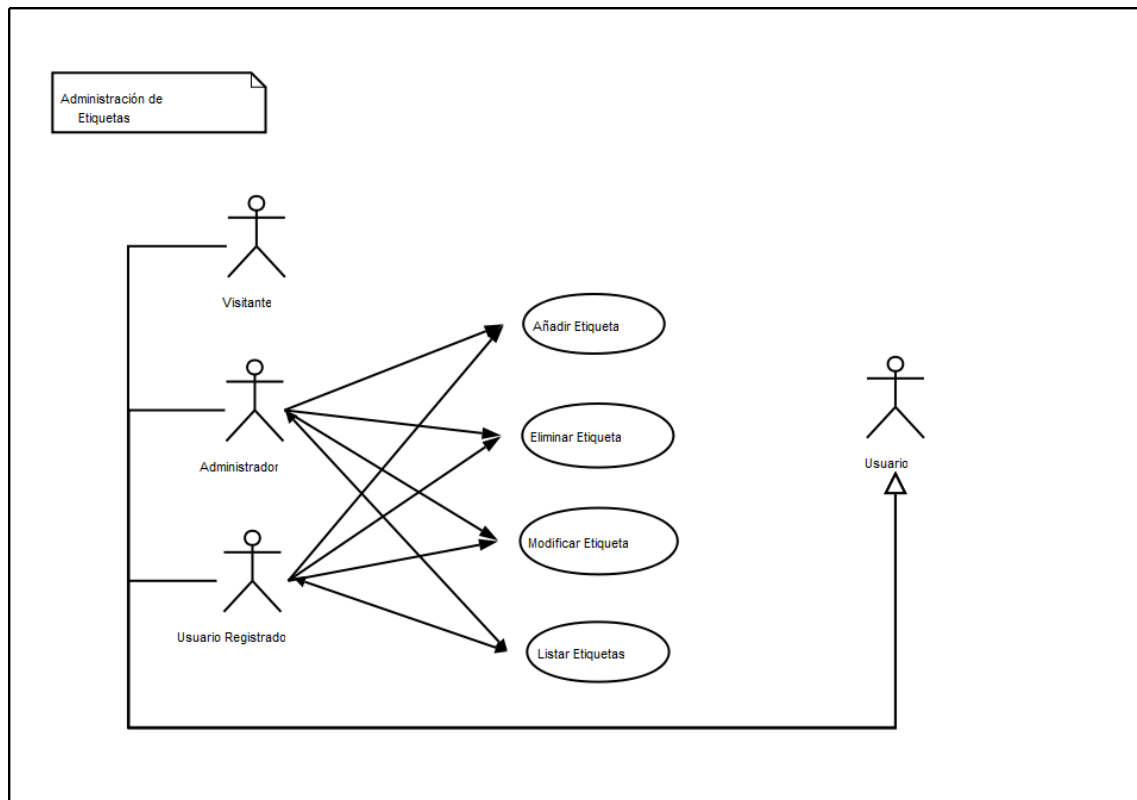


Figura 2.5: Diagrama de casos de uso de la administración de etiquetas

2.1.2. Descripción de los casos de uso

Caso de Uso: Modificación de usuario

Descripción: El administrador procede a modificar los datos de un usuario que pertenece a la wiki.

Precondición: El administrador debe estar logeando en el sistema.

Secuencia normal:

1. El administrador solicita al sistema modificar los datos de un usuario.
2. El administrador suministra el identificador del usuario registrado que desea modificar.
3. El sistema comprueba que el usuario está registrado.
4. El sistema recupera los datos del usuario.
5. El administrador modifica los datos que necesite dle usuario registrado.
6. El administrador confirma los cambios en los datos del usuario registrado.
7. El sistema actualiza los datos del usuario seleccionado y devuelve una confirmación.

Excepciones:

3.a El identificador del usuario no es correcto, el sistema cancela la operación y el caso de uso queda sin efecto.

Postcondición: Los datos del usuario quedan modificados.

Caso de Uso: Listado de usuario

Descripción: El administrador procede a los usuarios que pertenecen a la wiki.

Precondición: El administrador debe estar logeado en el sistema. Secuencia normal:

1. El administrador solicita al sistema listar los usuarios de la wiki.
2. El sistema recupera los datos del usuario.

Postcondición: El sistema muestra los usuarios registrados.

Caso de Uso: Modificación de datos por parte de usuario registrado

Descripción: El administrador procede a los usuarios que pertenecen a la wiki.

Precondición: El usuario debe estar registrado en la wiki y logeado en el sistema.

Secuencia normal:

1. El usuario registrado solicita modificar sus datos al sistema.
2. El sistema recupera los datos del usuario.
3. El usuario registrado modifica los datos necesarios y los manda al sistema.
4. El sistema modifica los datos del usuario registrado y muestra una confirmación.

Excepciones:

3.a El usuario no confirma la la modificación de los datos por tanto el sistema cancela la petición y el caso de uso queda sin efecto.

Postcondición: Los datos del usuario quedan modificados.

Caso de Uso: Crear Página

Descripción: Un usuario registrado desea crear una nueva página de contenido.

Precondición: El usuario debe estar registrado en la wiki y logeado en el sistema.

Secuencia normal:

1. El usuario registrado solicita crear una nueva página al sistema e introduce el nombre de la página.
2. El sistema comprueba si ya existe un nombre con esa página y manda una petición de confirmación al usuario.
3. El usuario confirma la creación de la página.
4. El sistema almacena en la base de datos el nuevo registro de página.
5. El sistema muestra la confirmación del alta de la página al usuario.

Excepciones:

- 2.a El nombre de la página ya existe y el caso de uso vuelve al paso 1.

Postcondición: La página queda dada de alta en el sistema.

Caso de Uso: Borrar Página

Descripción: Un usuario registrado desea borrar una nueva página de contenido.

Precondición: El usuario debe estar registrado en la wiki y logeado en el sistema y la página debe existir en el sistema.

Secuencia normal:

1. El usuario registrado solicita borrar una página al sistema.
2. El sistema comprueba y muestra todas las páginas que pertenecen al usuario.
3. El usuario registrado elige la página a eliminar.
4. El sistema solicita la confirmación de la eliminación al usuario.
5. El usuario confirma la petición.
6. El sistema elimina los datos de la página en la base de datos.
7. El sistema muestra un mensaje de confirmación de eliminación de la página solicitada.

Excepciones:

5.a El usuario no confirma la petición dando lugar a que el sistema cancele la operación quedando el caso de uso sin efecto.

Postcondición: La página queda eliminada del sistema.

Caso de Uso: Listar Páginas

Descripción: Un usuario registrado desea listar sus páginas de contenido.

Precondición: El usuario debe estar registrado en la wiki y logeado en el sistema y la página existe en el sistema.

Secuencia normal:

1. El usuario registrado solicita ver todas sus páginas al sistema.
2. El sistema muestra todas las páginas que pertenecen al usuario.

Postcondición: Las páginas quedan listadas en la pantalla del usuario.

Caso de Uso: Visitar Página

Descripción: Un usuario desea visualizar una página.

Precondición: El usuario debe estar logeado en el sistema y la página debe existir en el sistema.

Secuencia normal:

1. El usuario solicita ver una página al sistema.
2. El sistema pide al usuario el identificador de la página.
3. El usuario introduce el identificador de la página.
4. El sistema comprueba que la página exista y que esta sea pública.
5. El sistema muestra los datos de la página.

Excepciones:

5.a La página introducida no existe y el caso de uso vuelve al paso 2.

5.b La página introducida no es pública y el sistema cancela la operación quedando el caso de uso sin efecto.

Postcondición: La página queda mostrada en la pantalla del cliente.

Caso de Uso: Editar Página

Descripción: Un usuario registrado desea editar una página.

Precondición: El usuario debe estar registrado y logeado en el sistema y la página debe existir en el sistema.

Secuencia normal:

1. El usuario solicita modificar una página del sistema.
2. El sistema muestra todas las páginas pertenecientes al usuario.
3. El usuario elige la página a editar.
4. El sistema muestra el contenido de la página.
5. El usuario modifica los datos necesarios de la página y confirma la modificación de los datos.
6. El sistema modifica los datos de la página y muestra la confirmación de los cambios.

Excepciones:

*.a El usuario cancela la operación y el caso de uso queda sin efecto.

Postcondición: La página queda modificada.

Caso de Uso: Publicar Página

Descripción: Un usuario registrado desea publicar una página.

Precondición: El usuario debe estar registrado y logeado en el sistema y la página debe existir en el sistema.

Secuencia normal:

1. El usuario solicita publicar una página del sistema.
2. El sistema muestra todas las páginas pertenecientes al usuario.
3. El usuario elige la página a publicar.
4. El sistema comprueba que la página no esté publicada y la publica.

Excepciones:

4.a La página ya está publicada cancelando el sistema la operación y quedando el caso de uso sin efecto.

Postcondición: La página queda publicada.

Caso de Uso: Despublicar Página

Descripción: Un usuario registrado desea despublicar una página.

Precondición: El usuario debe estar registrado y logeado en el sistema y la página debe existir en el sistema.

Secuencia normal:

1. El usuario solicita despublicar una página del sistema.
2. El sistema muestra todas las páginas pertenecientes al usuario.
3. El usuario elige la página a despublicar.
4. El sistema comprueba que la página está publica y la despublica.

Excepciones:

4.a La página ya está despublicada y el sistema cancela la operación quedando el caso de uso sin efecto.

Postcondición: La página queda despublicada.

Caso de Uso: Inicio Sesión

Descripción: Un usuario quiere iniciar sesión en la página.

Precondición:

Secuencia normal:

1. El usuario solicita inciar sesión en el sistema.
2. El usuario introduce su login y su password.
3. El sistema comprueba que el login y el usuario son correctos y el usuario accede al sistema.

Excepciones:

3.a El login o el password son incorrectos y el sistema cancela la operación quedando el caso de uso sin efecto.

Postcondición: El usuario accede al sistema

Caso de Uso: Registro de usuario

Descripción: Un usuario quiere iniciar sesión en la página.

Precondición:

Secuencia normal:

1. El usuario solicita registrarse en el sistema.
2. El usuario introduce la información necesaria para el registro.
3. El sistema comprueba que el login y el usuario son correctos.
4. El sistema guarda la información referente al nuevo usuario.

Excepciones:

3.a El login ya existe y el sistema cancela la operación quedando el caso de uso sin efecto.

Postcondición: El usuario queda registrado en el sistema.

Caso de Uso: Etiquetar Página

Descripción: Un usuario desea etiquetar una página.

Precondición: La página debe existir y las páginas ya se encuentran listadas.

Secuencia normal:

1. El usuario solicita etiquetar una página.
2. El sistema muestra la página a etiquetar.
3. El usuario etiqueta la página solicitada.
4. El sistema comprueba que la etiqueta no exista en la página y guarda la etiqueta mostrando un mensaje de confirmación.

Excepciones:

4.a La página ya tiene una etiqueta con el mismo nombre, el sistema cancela la operación quedando el caso de uso sin efecto.

Postcondición: La página tiene una nueva etiqueta asociada.

Caso de Uso: Des-etiquetar Página

Descripción: Un usuario desea des-etiquetar una página.

Precondición: La página debe existir y la página ya se encuentra listada.

Secuencia normal:

1. El usuario solicita desetiquetar una página.
2. El sistema muestra la página a des-etiquetar.
3. El usuario elimina una etiqueta de la página.
4. El sistema comprueba que la etiqueta exista.
5. El sistema elimina la etiqueta y guarda los cambios.

Excepciones:

4.a La etiqueta no existe, el sistema cancela la operación quedando el caso de uso sin efecto.

Postcondición: La página queda con una etiqueta menos.

Caso de Uso: Listar etiquetas de una Página

Descripción: Un usuario registrado listar las etiquetas de una página.

Precondición: La página debe existir y la página ya se encuentra listada.

Secuencia normal:

1. El usuario solicita listar las etiquetas de una página.
2. El sistema muestra la página con todas sus etiquetas.

Postcondición: Las etiquetas quedan listadas.

Caso de Uso: Publicar página administrador

Descripción: El administrador desea publicar una página.

Precondición: El administrador debe estar logueado en el sistema.

Secuencia normal:

1. El administrador solicita publicar una página.
2. El sistema solicita el identificador de la página.
3. El administrador introduce el identificador de la página.
4. El sistema comprueba que el identificador es correcto y publica la página.

Excepciones:

4.a El identificador de la página no es correcto, el sistema cancela la acción quedando el caso de uso sin efecto.

Postcondición: La página queda publicada.

Caso de Uso: Despublicar página administrador

Descripción: El administrador desea despublicar una página.

Precondición: El administrador debe estar logueado en el sistema.

Secuencia normal:

1. El administrador solicita despublicar una página.
2. El sistema solicita el identificador de la página.
3. El administrador introduce el identificador de la página.
4. El sistema comprueba que el identificador es correcto y despublica la página.

Excepciones:

4.a El identificador de la página no es correcto, el sistema cancela la acción quedando el caso de uso sin efecto.

Postcondición: La página queda despublicada.

Caso de Uso:Listado de páginas administrador

Descripción: El administrador desea despublicar una página.

Precondición:El administrador debe estar logueado en el sistema.

Secuencia normal:

1. El administrador solicita listar las páginas de la wiki.
2. El sistema recupera el listado de páginas y lo muestra por pantalla.

Postcondición: Las páginas de la wiki quedan listadas.

Caso de Uso:Borrar página administrador

Descripción: El administrador desea borrar una página.

Precondición:El administrador debe estar logueado en el sistema y la página debe existir.

Secuencia normal:

1. El administrador solicita borrar una página de la wiki.
2. El sistema solicita el identificador de la página.
3. El administrador introduce el identificador de la página.
4. El sistema comprueba que el identificador es correcto y muestra un mensaje de confirmación.
5. El administrador confirma el borrado de la página.
6. La página se borra del sistema devolviendo un mensaje de confirmación de la operación.

Excepciones:

4.a El identificador introducido por el usuario es incorrecto, se vuelve al paso 2.

*.a El administrador cancela la operación, quedando el caso de uso sin efecto.

Postcondición: La página queda borrada del sistema.

Caso de Uso:Etiquetar Página Administrador

Descripción: El administrador desea etiquetar una página.

Precondición:El administrador debe estar logueado en el sistema y la página debe existir.

Secuencia normal:

1. El administrador solicita etiquetar una página de la wiki.
2. El sistema solicita el identificador de la página.
3. El administrador introduce el identificador de la página.
4. El sistema comprueba que el identificador es correcto y muestra el contenido de la página.
5. El administrador crea la etiqueta y la asocia a la página.
6. El sistema realiza la acción y devuelve un mensaje de confirmación.

Excepciones:

4.a El identificador introducido por el usuario es incorrecto, se vuelve al paso 2.

*.a El administrador cancela la operación, quedando el caso de uso sin efecto.

Postcondición: La página queda etiquetada en el sistema.

Caso de Uso:Desetiquetar Página Administrador

Descripción: El administrador desea desetiquetar una página.

Precondición:El administrador debe estar logueado en el sistema y la página debe existir.

Secuencia normal:

1. El administrador solicita etiquetar una página de la wiki.
2. El sistema solicita el identificador de la página.
3. El administrador introduce el identificador de la página.
4. El sistema comprueba que el identificador es correcto y muestra el contenido de la página con sus etiquetas.
5. El administrador elige la etiqueta a eliminar.
6. El sistema realiza la acción y devuelve un mensaje de confirmación.

Excepciones:

*.a El administrador cancela la operación, quedando el caso de uso sin efecto.

Postcondición: La etiqueta de la página queda borrada del sistema.

Caso de Uso:Listar Etiquetas de una PáginasAdministrador

Descripción: El administrador desea listar las páginas de la wiki.

Precondición:El administrador debe estar logueado en el sistema y la página debe existir.

Secuencia normal:

1. El administrador solicita ver todas las etiquetas de una página de la wiki.
2. El sistema solicita el identificador de la página.
3. El administrador introduce el identificador de la página.
4. El sistema comprueba que el identificador es correcto y muestra el contenido de la página con sus etiquetas.

Excepciones:

*.a El administrador cancela la operación, quedando el caso de uso sin efecto.

Postcondición: Las etiquetas de la página quedan listadas.

Caso de Uso:Listar Etiquetas de una Página Administrador

Descripción: El administrador desea listar las páginas de la wiki.

Precondición:El administrador debe estar logueado en el sistema y la página debe existir.

Secuencia normal:

1. El administrador solicita ver todas las etiquetas de una página de la wiki.
2. El sistema solicita el identificador de la página.
3. El administrador introduce el identificador de la página.
4. El sistema comprueba que el identificador es correcto y muestra el contenido de la página con sus etiquetas.

Excepciones:

*.a El administrador cancela la operación, quedando el caso de uso sin efecto.

Postcondición: Las etiquetas de la página quedan listadas.

Caso de Uso: Modificar etiquetas de una Páginas Administrador

Descripción: El administrador desea modificar las etiquetas de una página de la wiki.

Precondición: El administrador debe estar logueado en el sistema y la página debe existir.

Secuencia normal:

1. El administrador solicita modificar todas las etiquetas de una página de la wiki.
2. El sistema solicita el identificador de la página.
3. El administrador introduce el identificador de la página.
4. El sistema comprueba que el identificador es correcto y muestra el contenido de la página con sus etiquetas.
5. El administrador elige la etiqueta y modifica los datos necesario y envía los cambios.
6. El sistema guarda los cambios de la etiqueta solicitada y envía confirmación al usuario.

Excepciones:

- *.a El administrador cancela la operación, quedando el caso de uso sin efecto.
- 4.a El identificador de la página es incorrecto, volviendo al paso 2.

Postcondición: Las etiquetas de la página quedan listadas.

Caso de Uso: Modificar etiqueta de una página

Descripción: El usuario registrado desea listar las páginas de la wiki.

Precondición: El usuario debe estar logueado en el sistema y la página debe existir y las páginas deben estar listadas.

Secuencia normal:

1. El usuario solicita modificar la etiqueta de una página.
2. El sistema muestra la etiqueta a modificar.
3. El usuario introduce los cambios en la etiqueta de la página.
4. El sistema comprueba los cambios y los guarda, devolviendo un mensaje de confirmación.

Excepciones:

- *.a El administrador cancela la operación, quedando el caso de uso sin efecto.

Postcondición: La etiqueta de la página queda modificada.

2.2. Arquitectura del sistema

Vamos a implementar el patrón de arquitectura MVC (Modelo-Vista-Controlador), dado que éste es el patrón de diseño de arquitectura recomendado para aplicaciones interactivas.

MVC, Modelo Vista Controlador, es un patrón que permite una separación entre las distintas capas de una aplicación Web:

- Capa de presentación (Vista)
- Receptor de las acciones (Controlador)

- Modelo de datos ya sean bases de datos u otros (Modelo).

Con esta separación, conseguimos aplicaciones más fáciles de mantener, con menos acoplamientos, más inteligibles e incluso más adaptables potenciando en gran medida la calidad del software.

Una de las ventajas más destacadas es que te permite desarrollar la aplicación por separado. Esto toma sentido cuando se trata de una aplicación desarrollada por un grupo de trabajo.

JSF (que es el framework elegido para el desarrollo de nuestro proyecto) encaja bien en la arquitectura de la capa de presentación basada en MVC. Ofrece una clara separación entre el comportamiento y la presentación. Une los familiares componentes UI con los conceptos de la capa-Web sin limitarnos a una tecnología de script o lenguaje de marcas particular.

En la siguiente imagen, podemos ver como JSF sigue el patrón Modelo-Vista-Controlador:

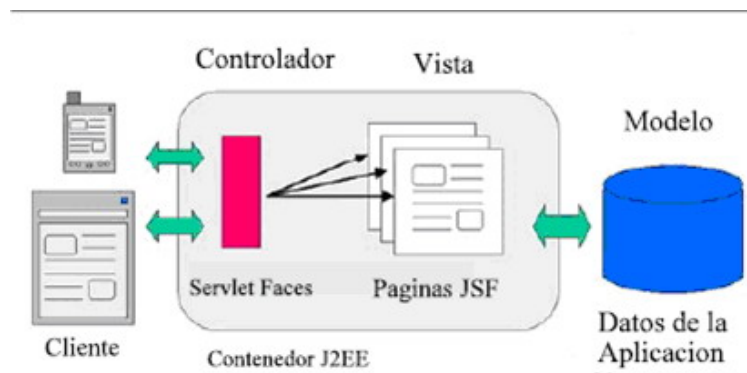


Figura 2.6: Modelo Vista Controlador en JSF

Vamos a describir como se representa cada componente del patrón en JSF:

2.2.1. Modelo

Objeto que representa y trabaja directamente gestionando los datos y controlando todas sus transformaciones. El modelo no tiene ninguna referencia de las vistas. JSF se comunica con el modelo de la aplicación a través del fichero faces-config.xml donde se detalla un objeto perteneciente a la clase del dominio Usuario, la clase a la que hace referencia y el ámbito de sesión.

```
<managed-bean>  
<managed-bean-name>usuario</managed-bean-name>  
<managed-bean-class>CapaDominio.Usuario</managed-bean-class>  
<managed-bean-scope>session</managed-bean-scope>  
</managed-bean>
```

Figura 2.7: MVC - Modelo

2.2.2. Vista

Objeto que maneja la presentación visual de los datos gestionados por el Modelo.

Genera una representación visual del modelo y muestra los datos al usuario. En nuestro proyecto vendrá determinado por cada una de las páginas JSF. La manera de conectar el modelo con la vista a través de componentes JSF es haciendo referencia a objetos del modelo existentes en el controlador.

```
<h:inputText value="#usuario.nombre"/>
```

2.2.3. Controlador

Objeto que actúa sobre los datos del modelo y que entra en acción cuando se realiza una petición por parte de la vista o una actualización por parte del modelo hacia la vista. En JSF opera como un gestor que reacciona ante los eventos provocados por el usuario, procesa sus acciones y los valores de estos eventos, ejecutando código para actualizar el modelo o la vista. Un ejemplo de petición sería:

```
<h:commandButton value="Aceptar" action="#login"/>
```

Donde la regla de navegación estará definida en el archivo faces-config.xml de la siguiente manera:

```
<navigation-rule>  
  <from-view-id>/index.jsp</from-view-id>  
  <navigation-case>  
    <from-outcome>login</from-outcome>  
    <to-view-id>/hola.jsp</to-view-id>  
  </navigation-case>  
</navigation-rule>
```

Figura 2.8: MVC - Controlador

3.1. Diagrama de Despliegue

Presentamos el diagrama de despliegue para modelar el Hardware del sistema a desarrollar.

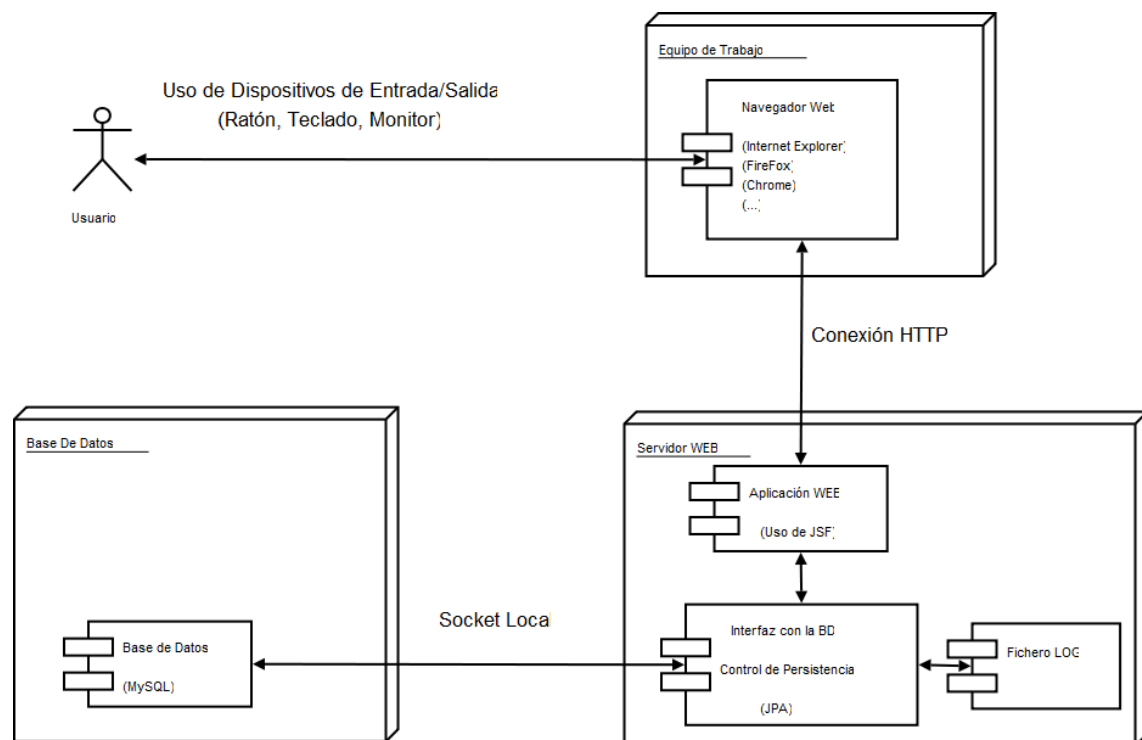


Figura 3.1: Diagrama de despliegue

3.2. Diagrama de Clases

Este es el diagrama de clases que hemos considerado para nuestra wiki. Se ven reflejados los diferentes métodos de las clases que vamos a emplear.

En el diagrama de clases, en las diferentes asociaciones que hay entre las clases, no hace falta ponerle un nombre a la asociación, ya que se relaciona el nombre de dicha asociación según la clase con la que se asocia. Por ejemplo, "Usuario Registrado" cuando se asocia con "Etiqueta", la asociación tendrá los nombres de "Crear etiqueta", "Modificar etiqueta"... cuando se asocia con la "Página" tendrá la asociación los nombres de "Crear página", "Modificar página"... Las anotaciones que aparecen en el diagrama sirven para aclarar aquello que no se puede reflejar en el diagrama de clases, por ejemplo el que un "Usuario Registrado" sólo pueda modificar una "Página" propia.

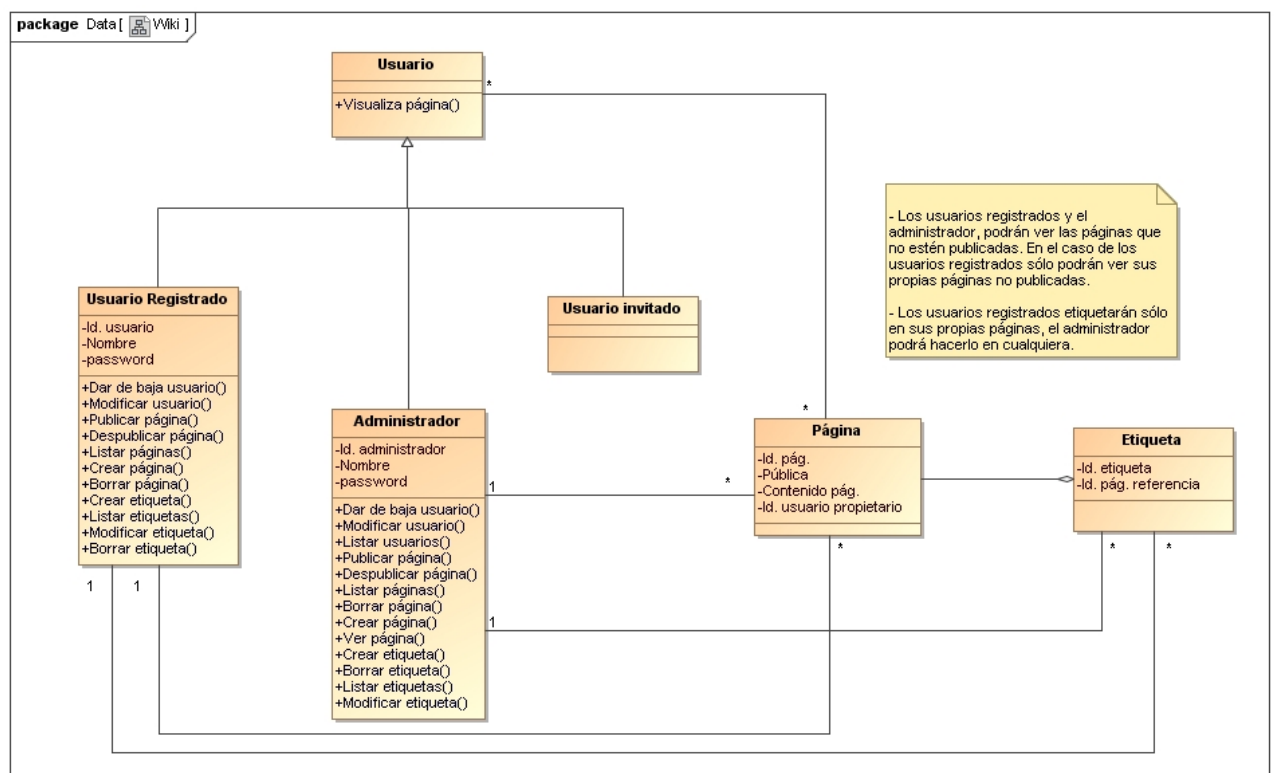


Figura 3.2: Diagrama de clases

3.3. Diagrama de navegabilidad

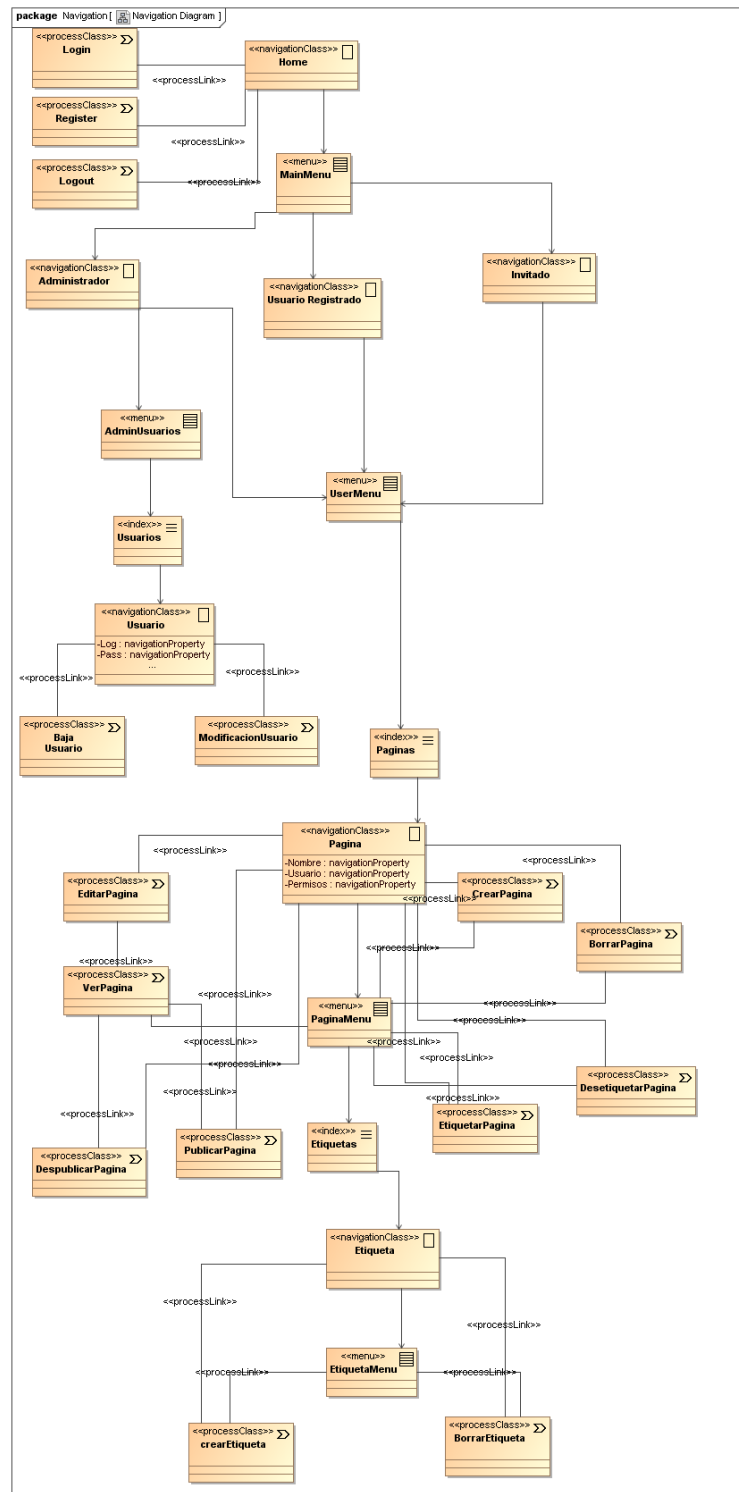


Figura 3.3: Diagrama de navegabilidad

3.4. Diagrama Entidad/Relación

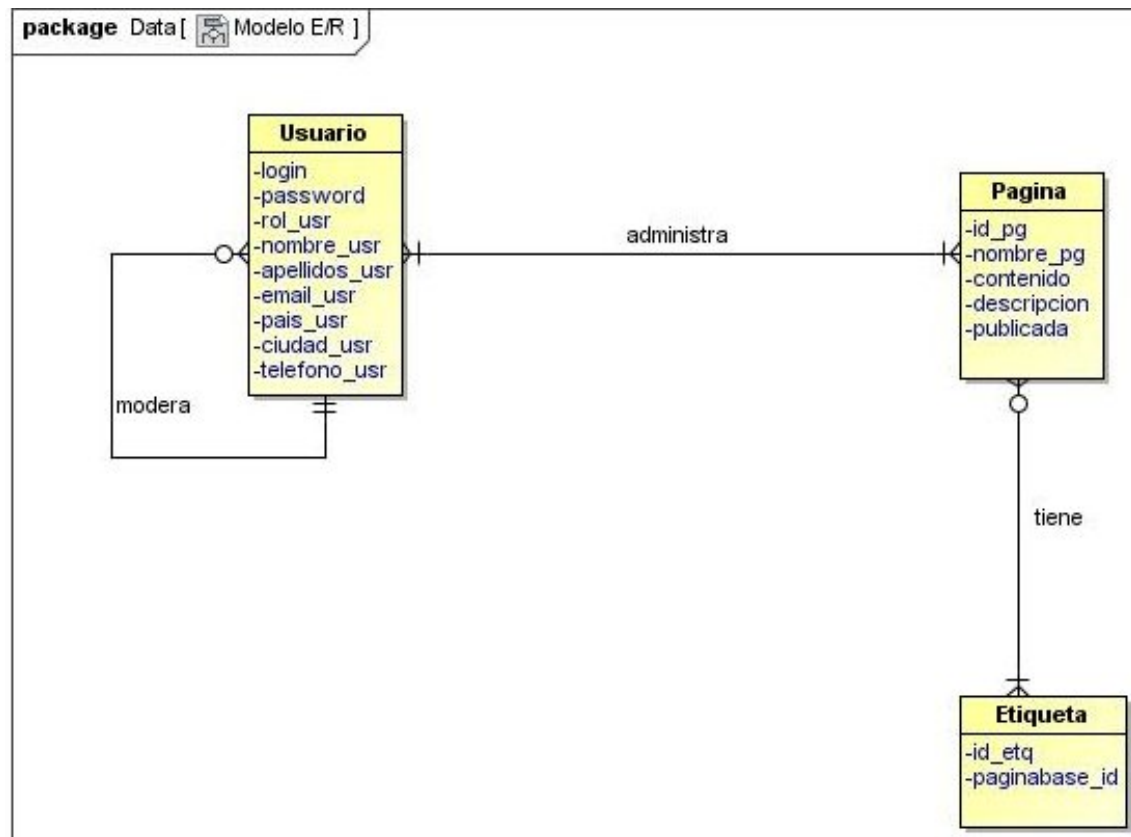


Figura 3.4: Diagrama Entidad/Relación

4

Capítulo

Documentos de diseño

4.1. Diagramas Robustness

Estos son los diagramas de Robustness, estos diagramas sirven para ver como funciona la interacción de cada de caso de uso. Podemos ver las interacciones de los distintos usuarios con el sistema, pero el sistema ya está dividido en sus partes funcionales, existen 3 tipos de componentes de las cuales se compone el sistema:

1. Componentes Boundary(Limitadores): Estos componentes son los que se comunican con los distintos actores y con los controladores.
2. Componentes Controlador: Estos llevan la lógica del sistema e interactuan tanto con los componentes Boundary como con los componentes Entidad.
3. Componentes Entidad: Son los tipos de entidad que nos encontramos en los modelos conceptuales

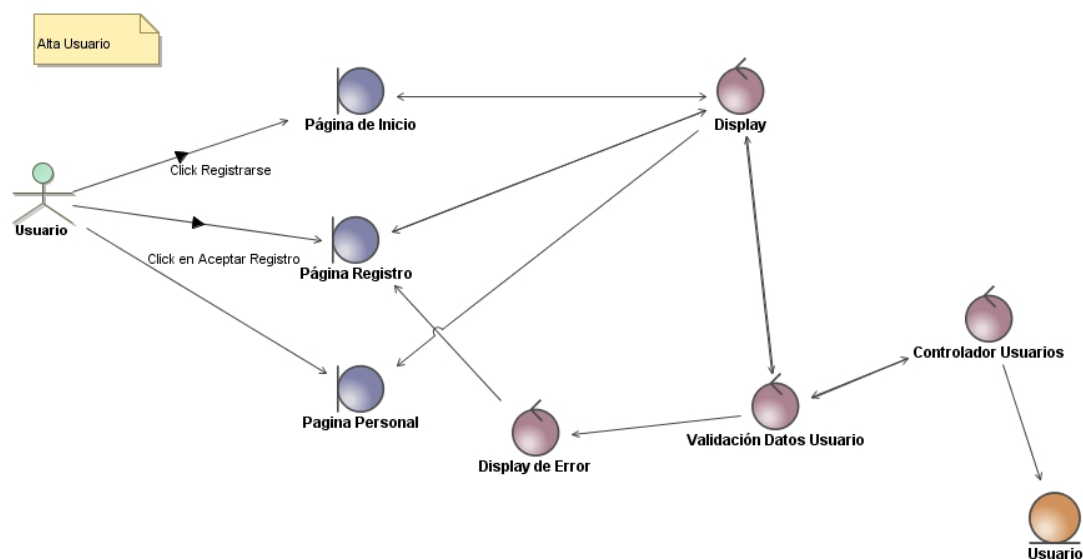


Figura 4.1: Diagrama Robustness Alta usuario

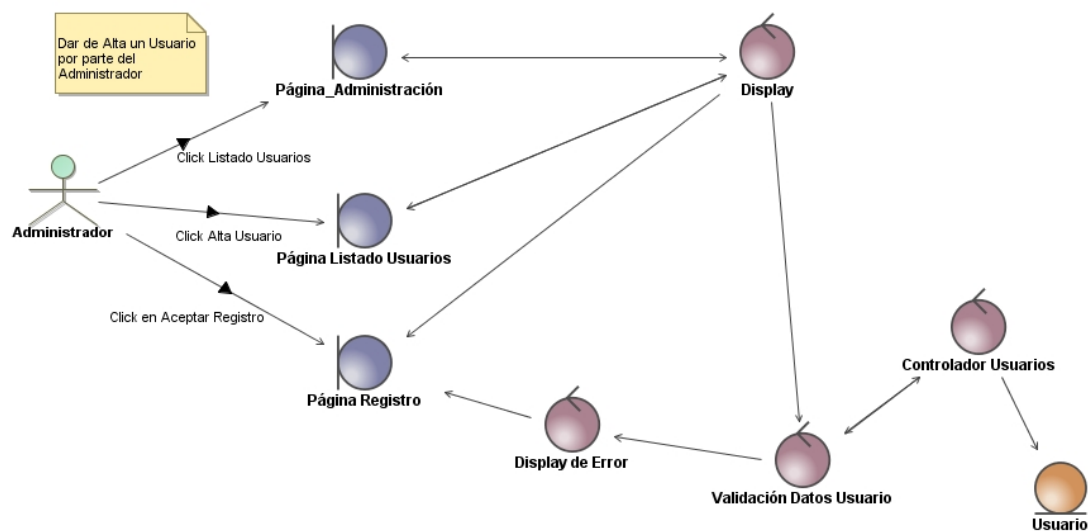


Figura 4.2: Diagrama Robustness Alta usuario por parte del administrador

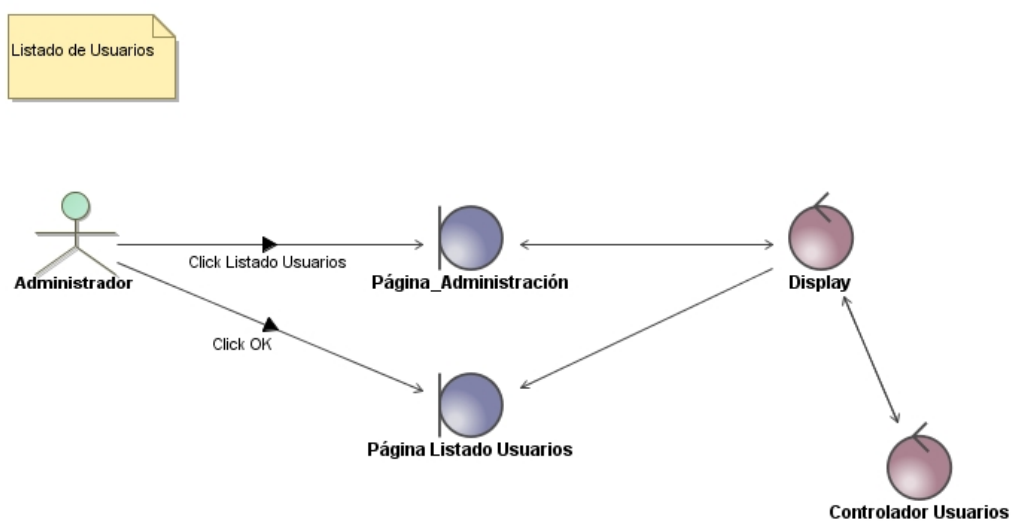


Figura 4.3: Diagrama Robustness Listado de usuarios

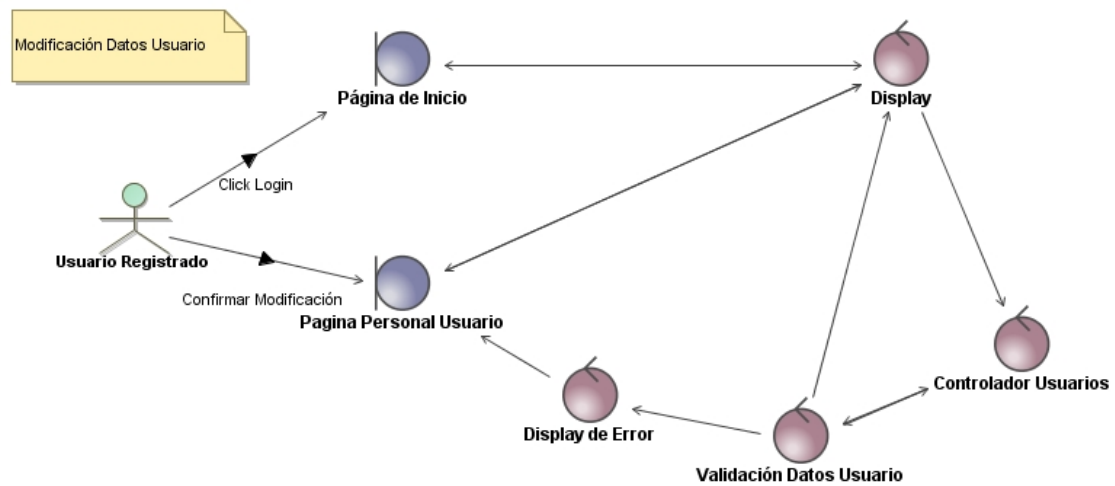


Figura 4.4: Diagrama Robustness Listado de usuarios

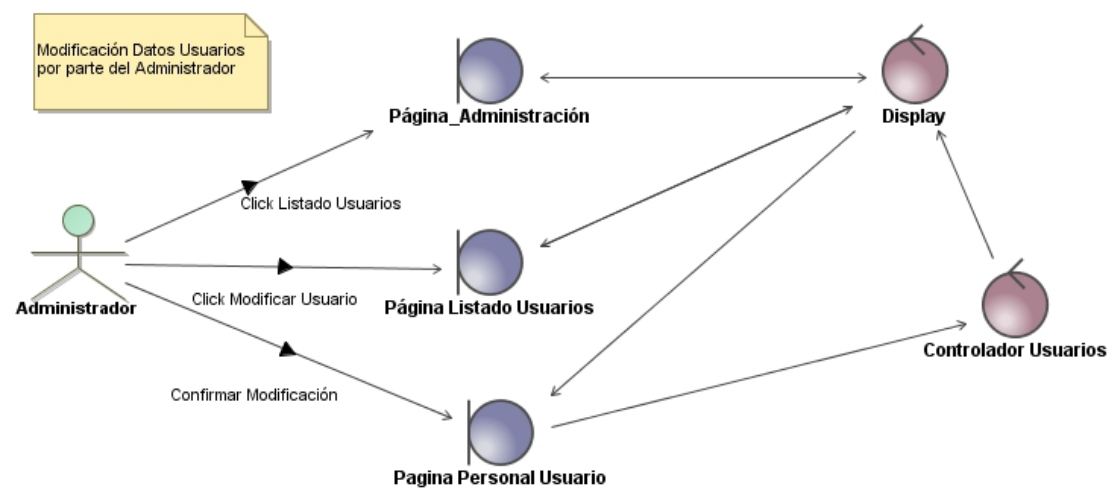


Figura 4.5: Diagrama Robustness Listado de usuarios

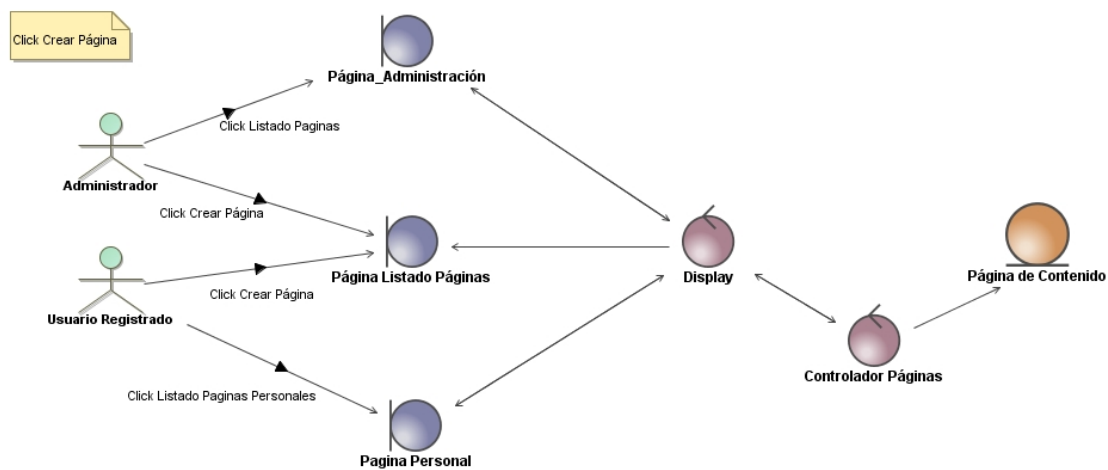


Figura 4.6: Diagrama Robustness Crear página

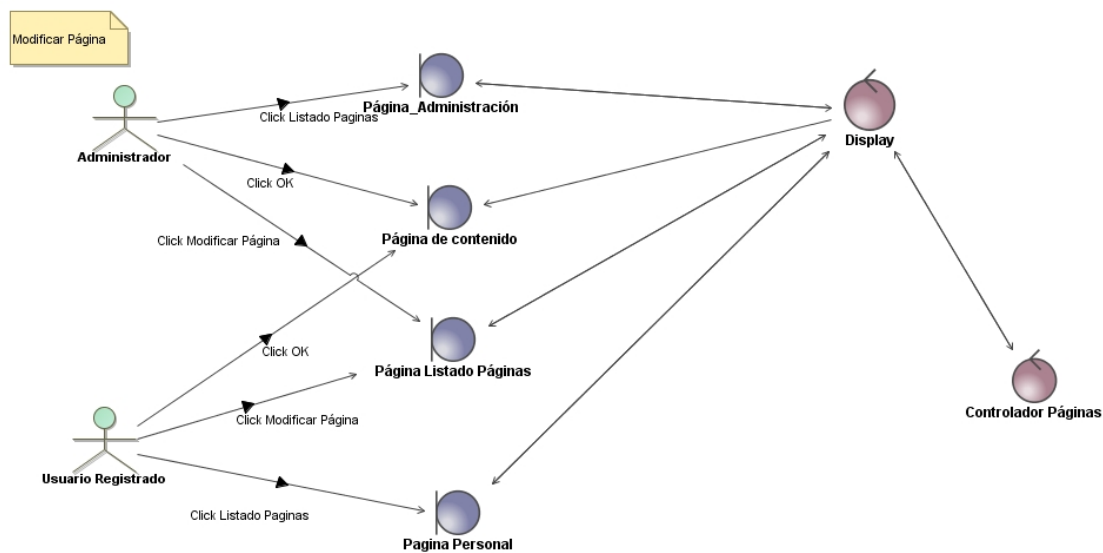


Figura 4.7: Diagrama Robustness Modificar página

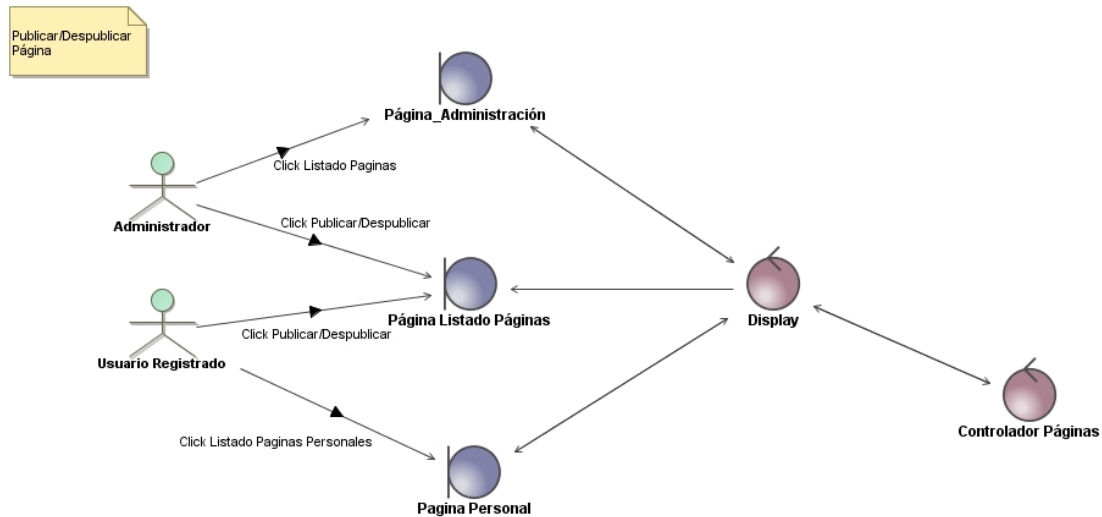


Figura 4.8: Diagrama Robustness Publicar/Despublicar página

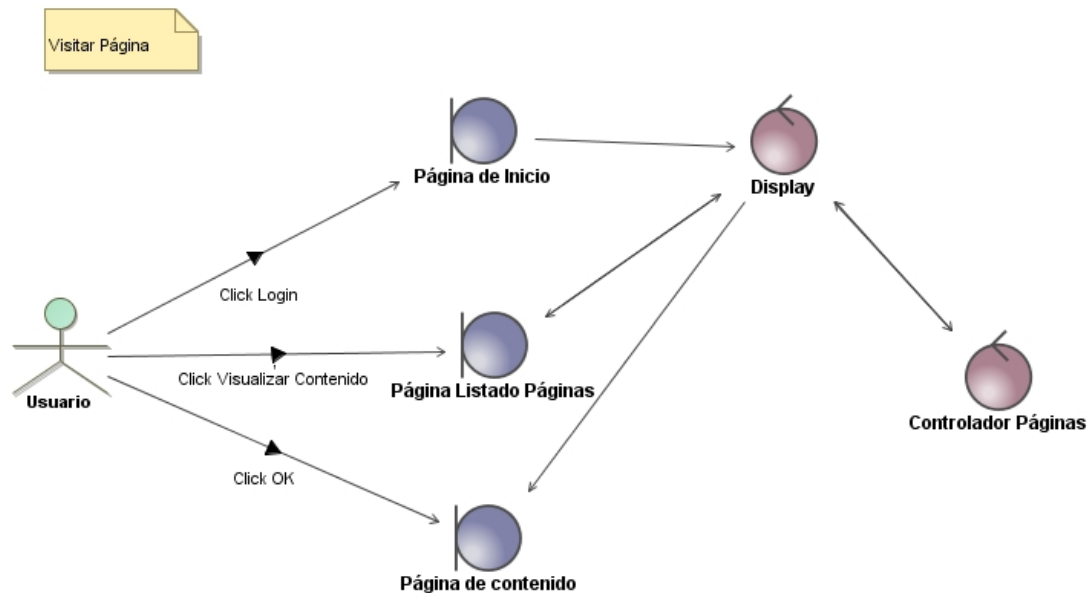


Figura 4.9: Diagrama Robustness Visitar página

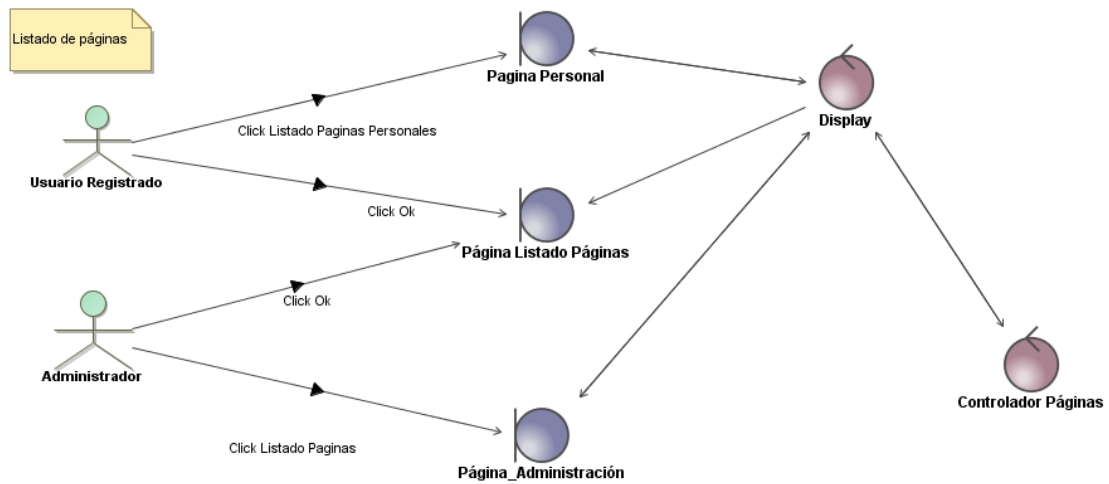


Figura 4.10: Diagrama Robustness Listado de páginas

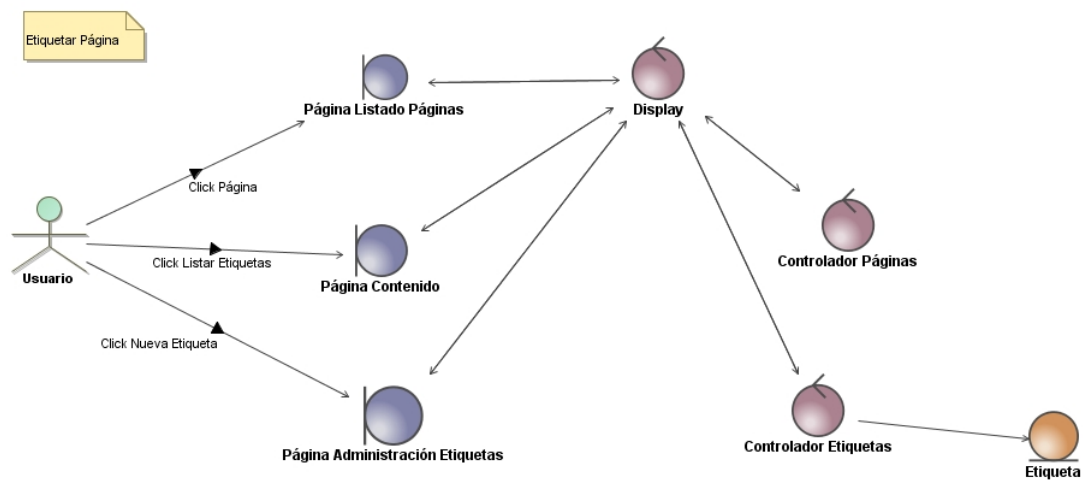


Figura 4.11: Diagrama Robustness Etiquetar página

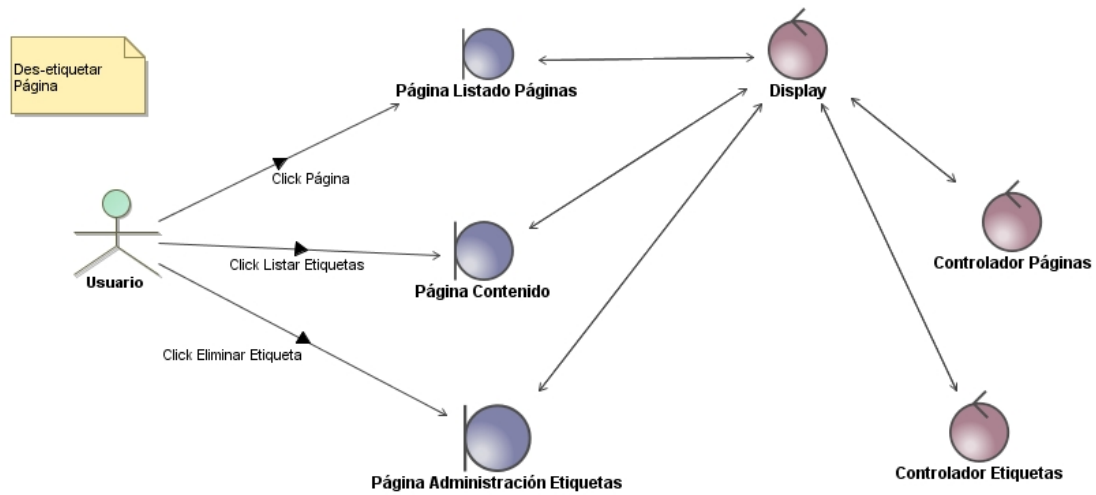


Figura 4.12: Diagrama Robustness Des-etiquetar página

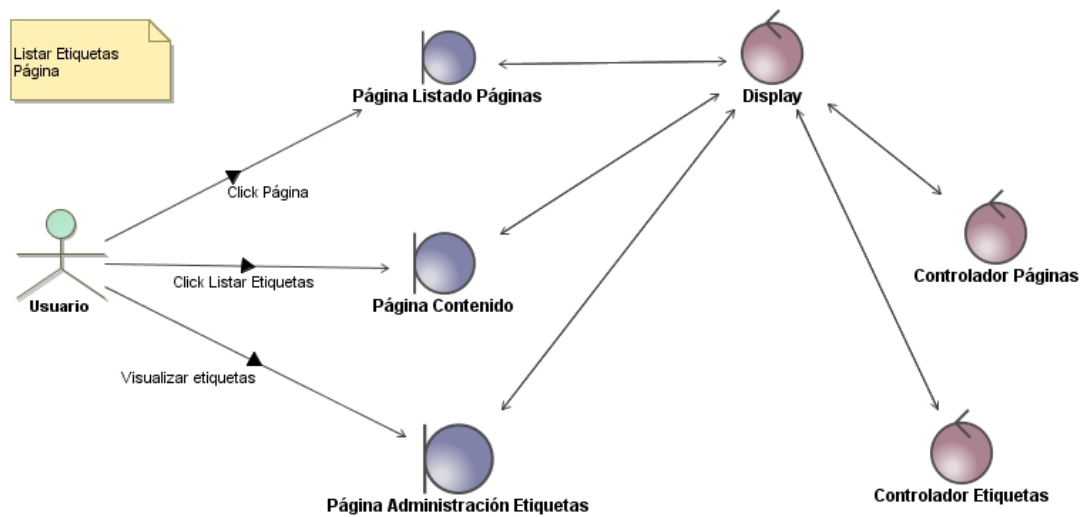


Figura 4.13: Diagrama Robustness Listar etiquetas de una página

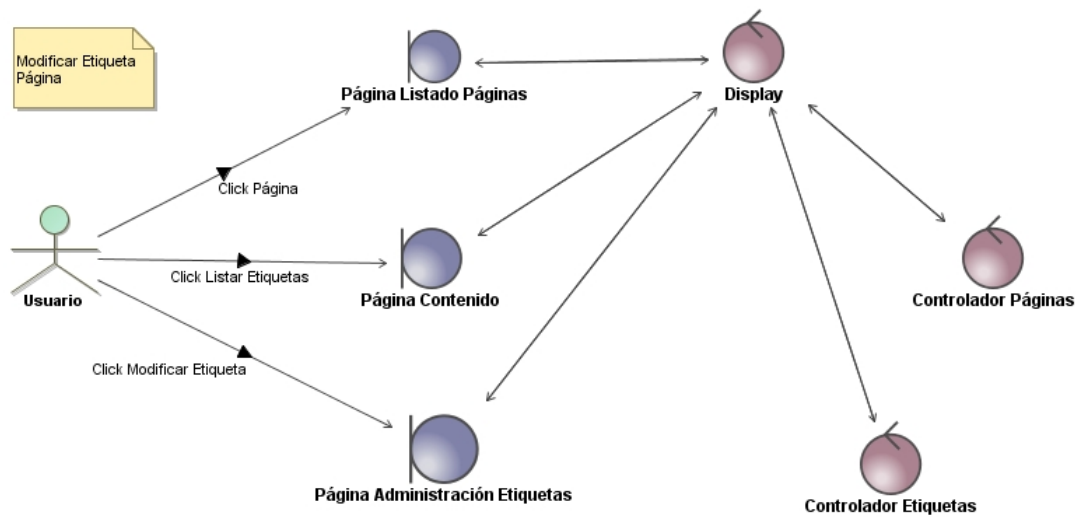


Figura 4.14: Diagrama Robustness Modificar etiqueta de una página

4.2. Diagramas de actividad

Presentamos los diferentes diagramas de actividad correspondientes al paquete de trabajo de gestión de usuarios: login usuario, listado usuarios, alta usuario, alta usuario por el administrador, modificar usuario y modificar usuario por el administrador.

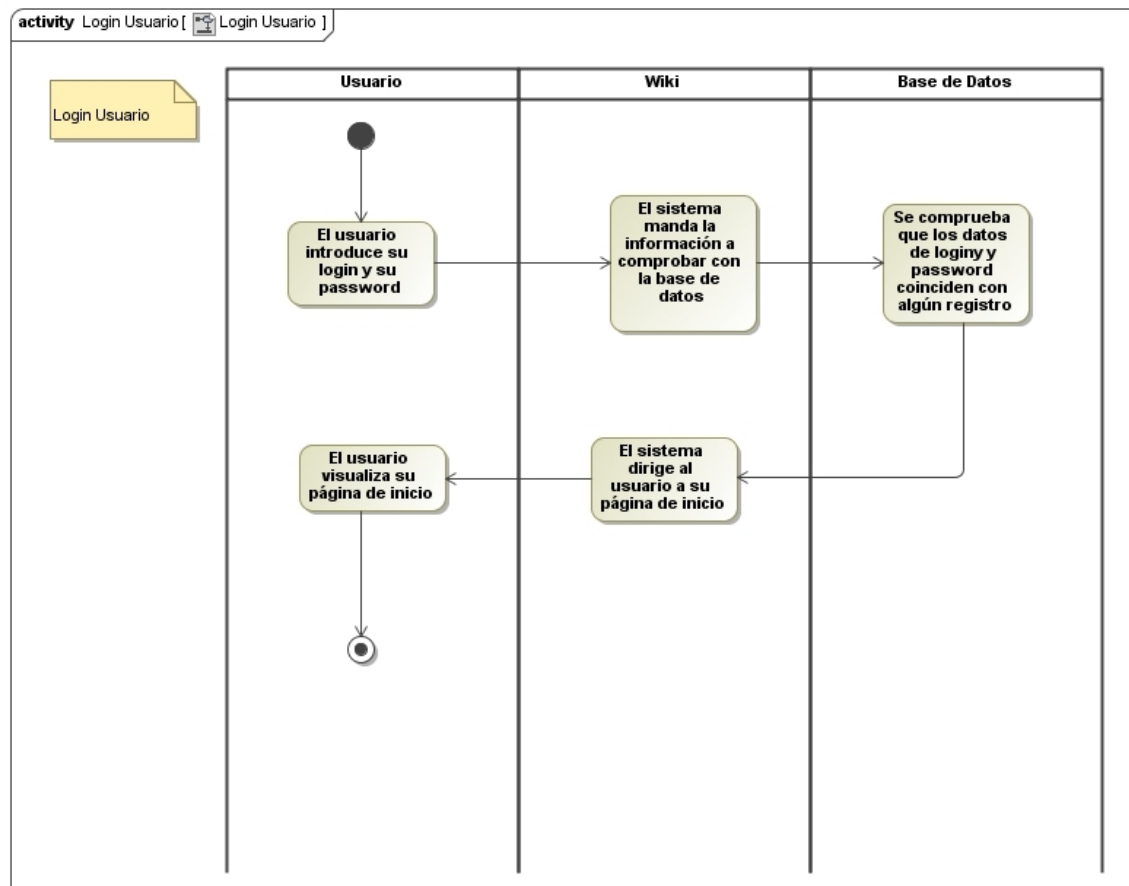


Figura 4.15: Diagrama de actividad login usuario

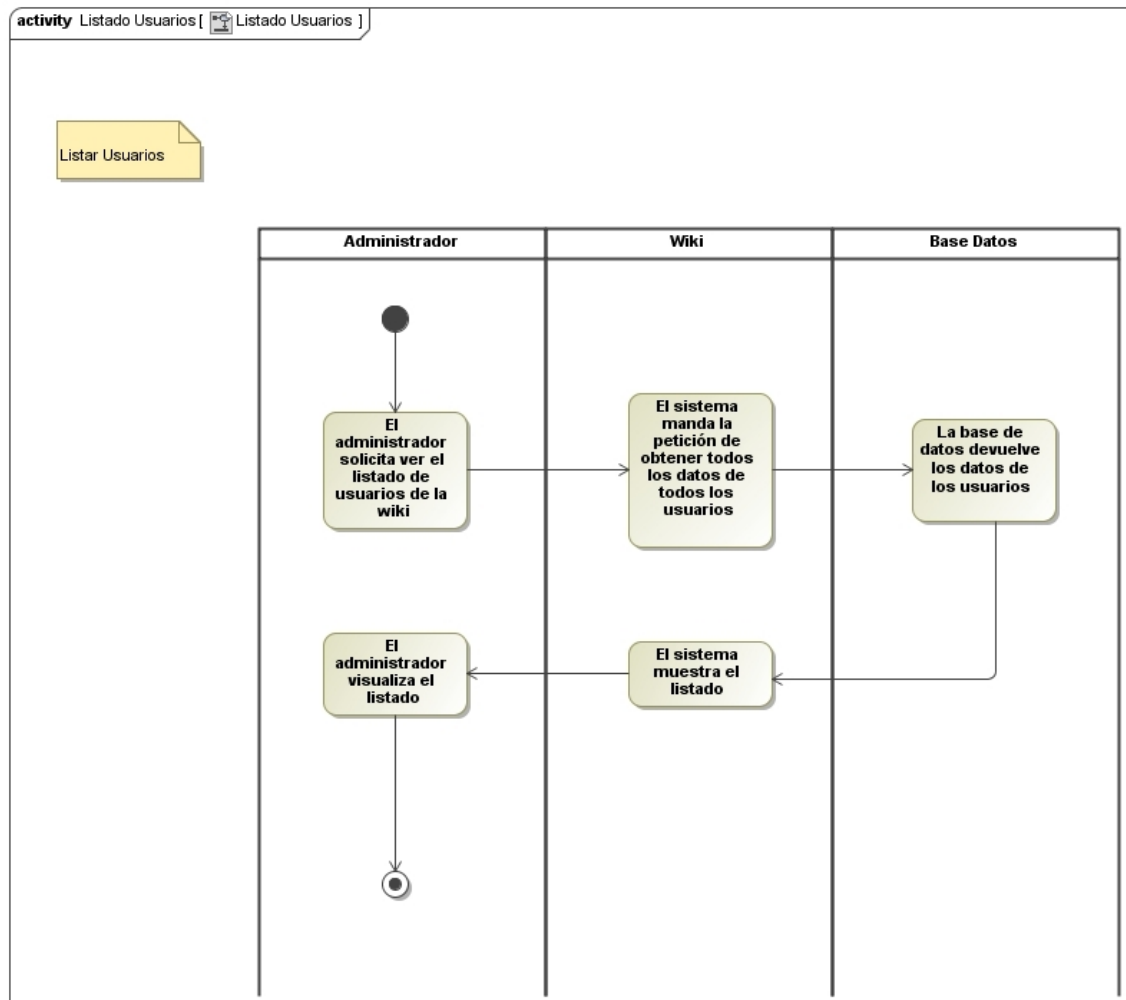


Figura 4.16: Diagrama de actividad listado usuarios

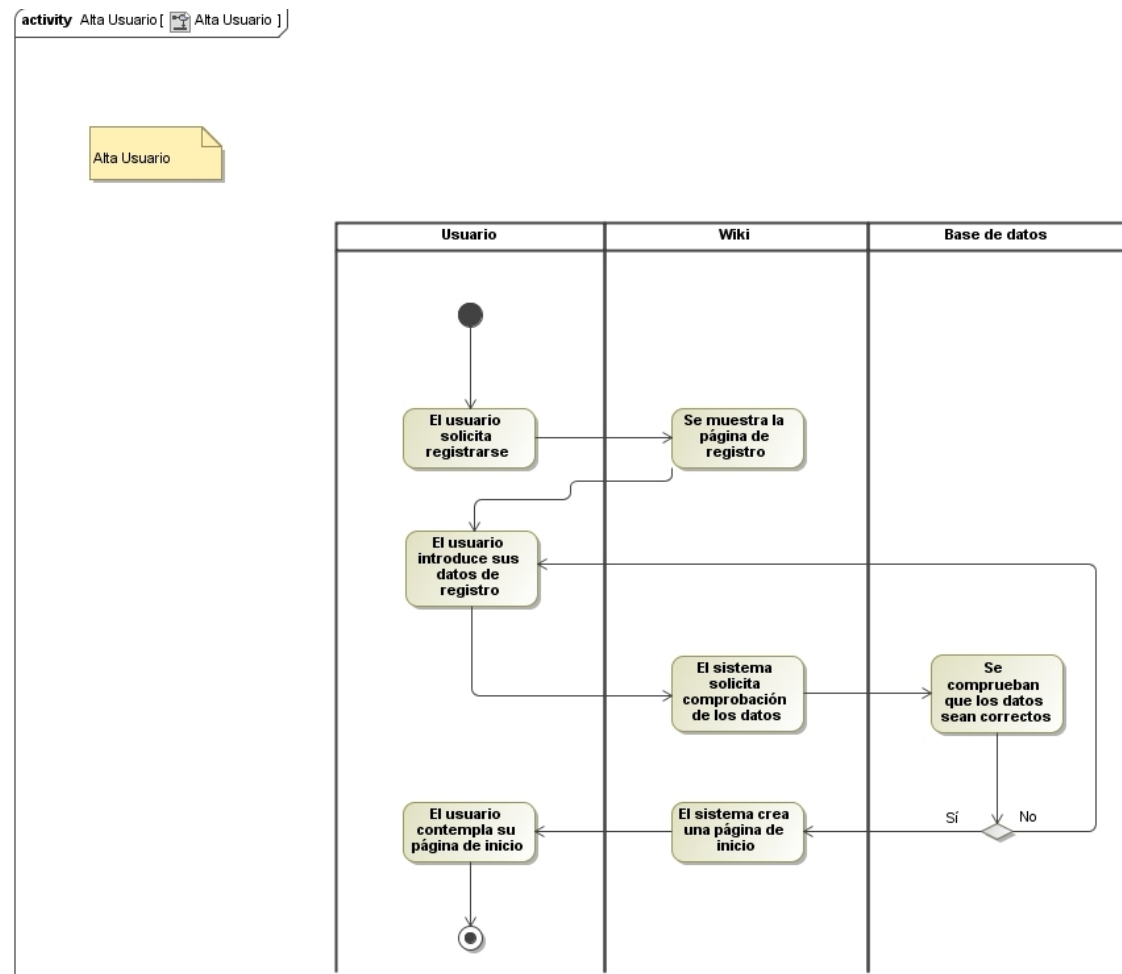


Figura 4.17: Diagrama de actividad alta usuario

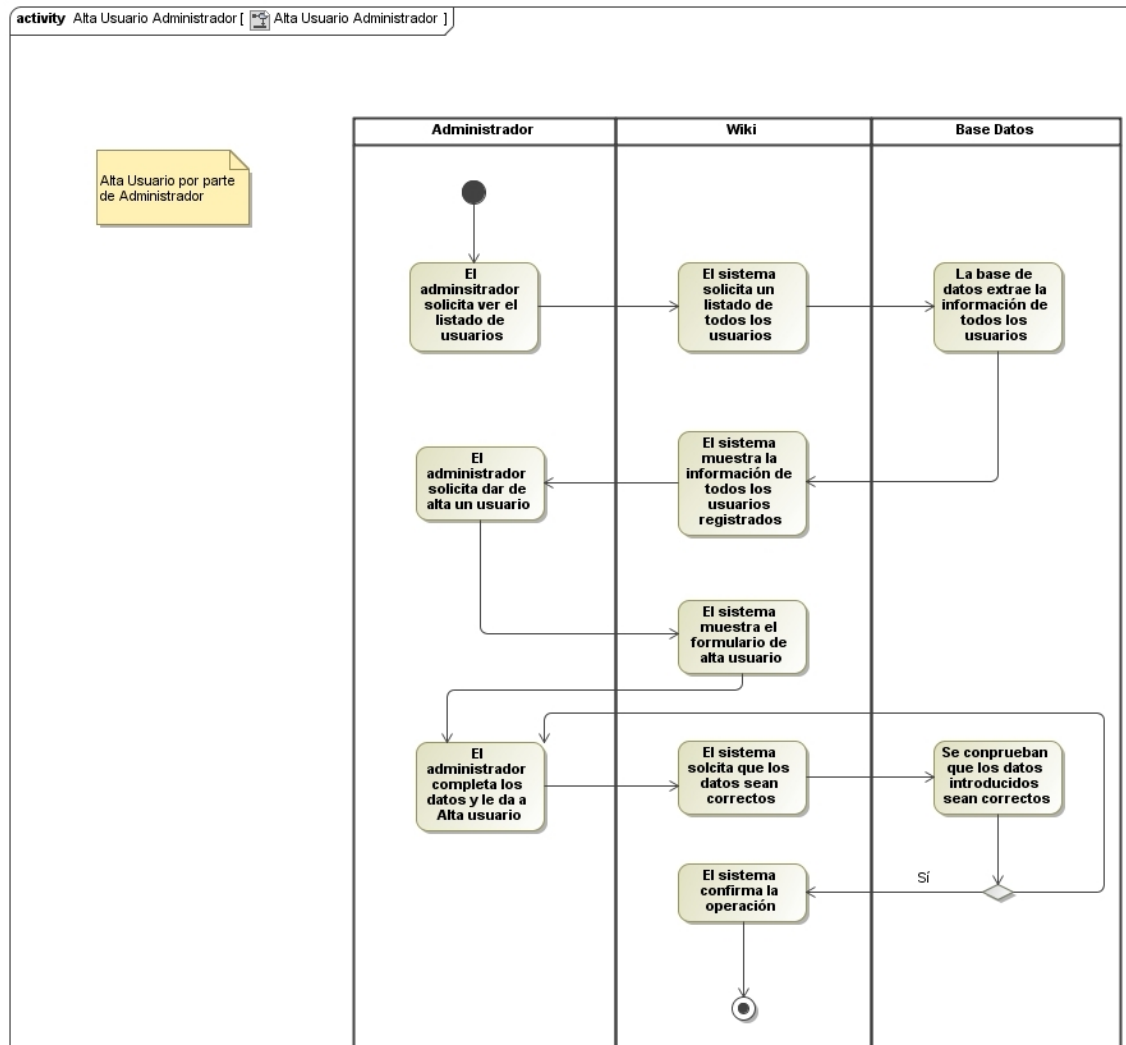


Figura 4.18: Diagrama de actividad alta usuario por administrador

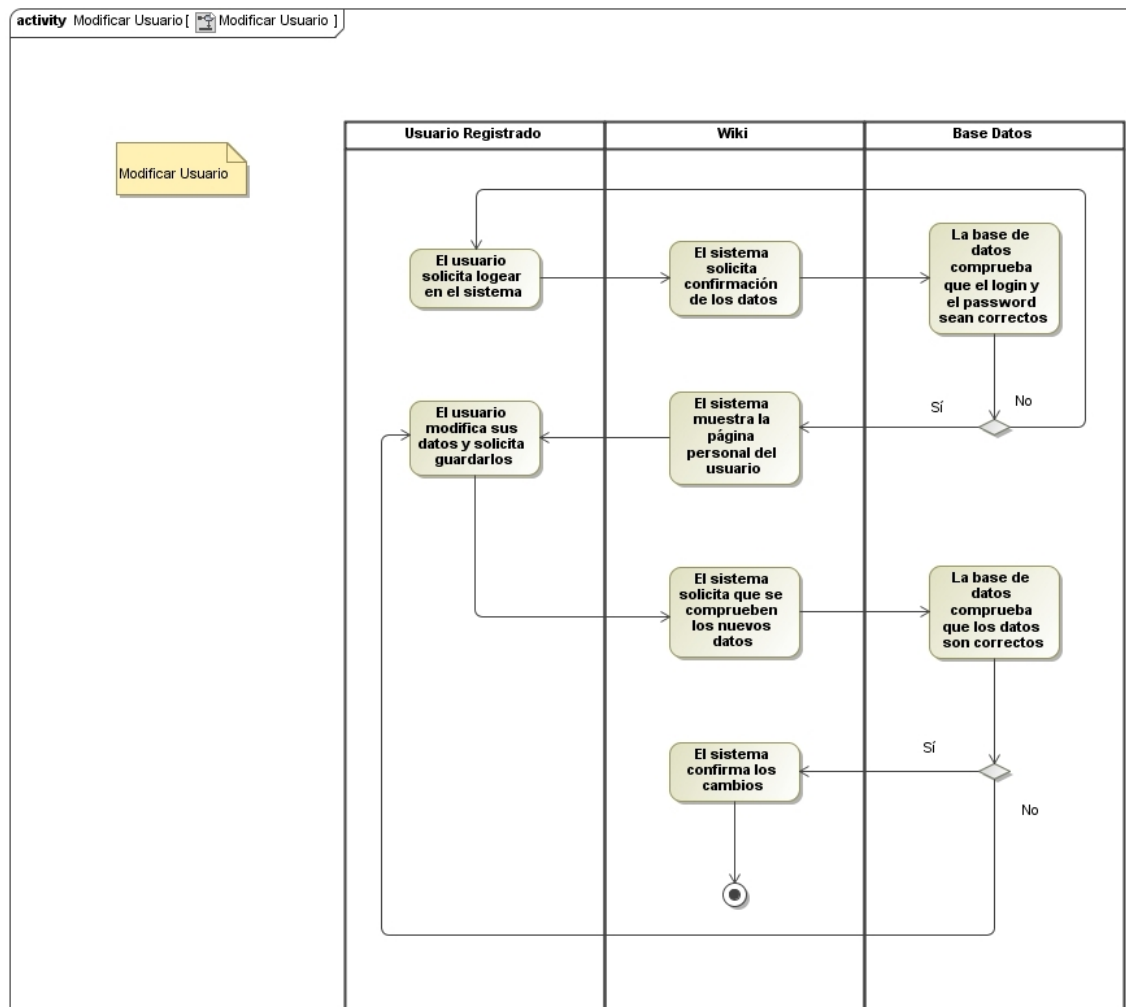


Figura 4.19: Diagrama de actividad modificar usuario

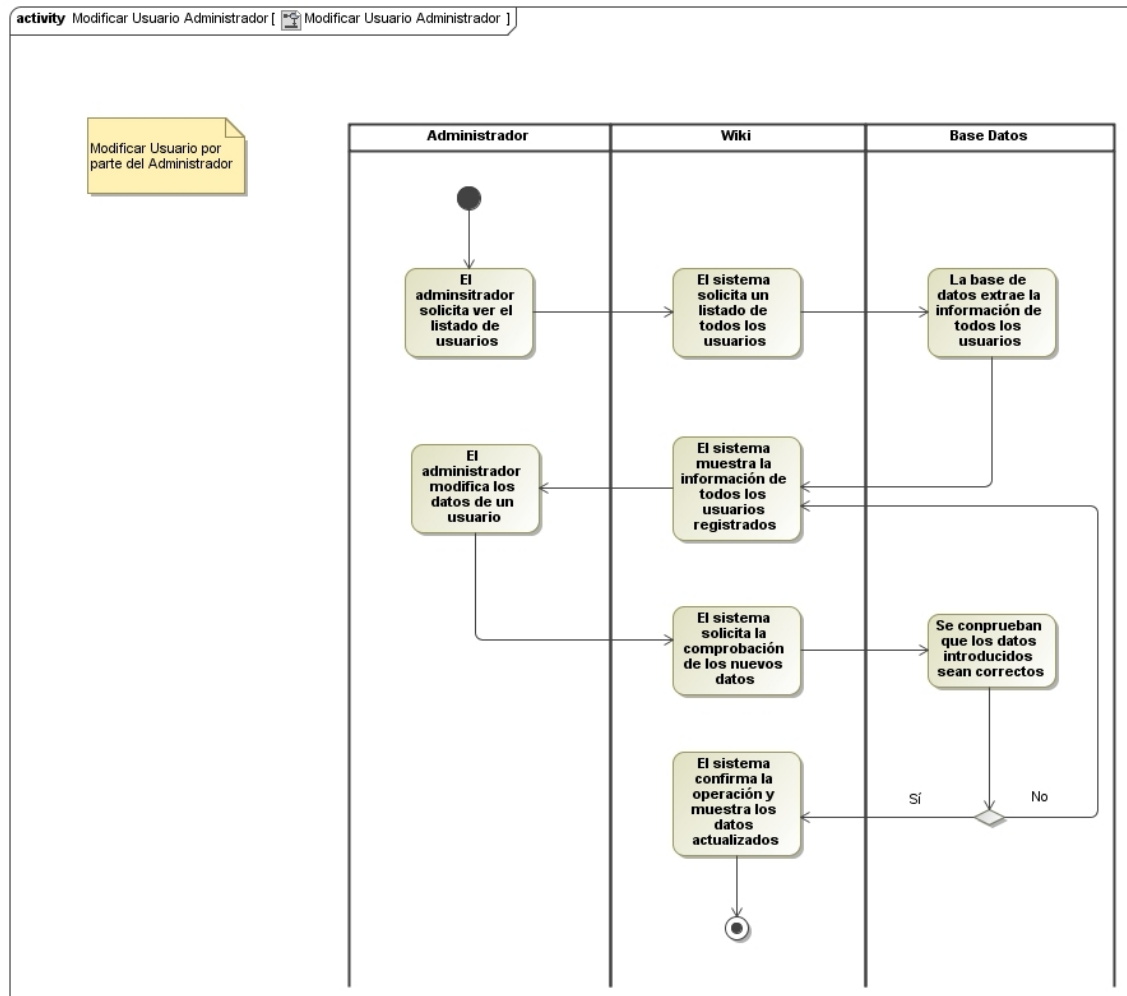


Figura 4.20: Diagrama de actividad modificar usuario por administrador

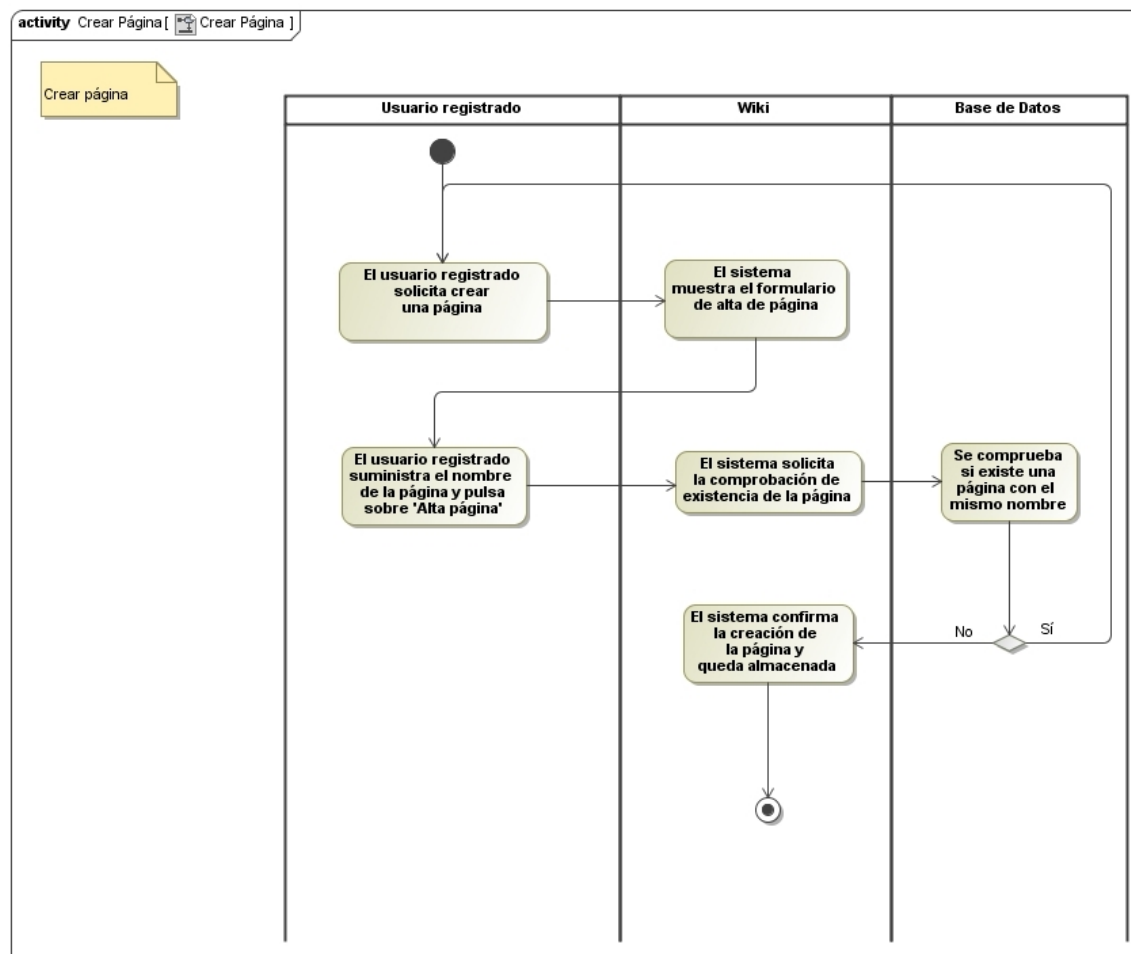


Figura 4.21: Diagrama de actividad crear página

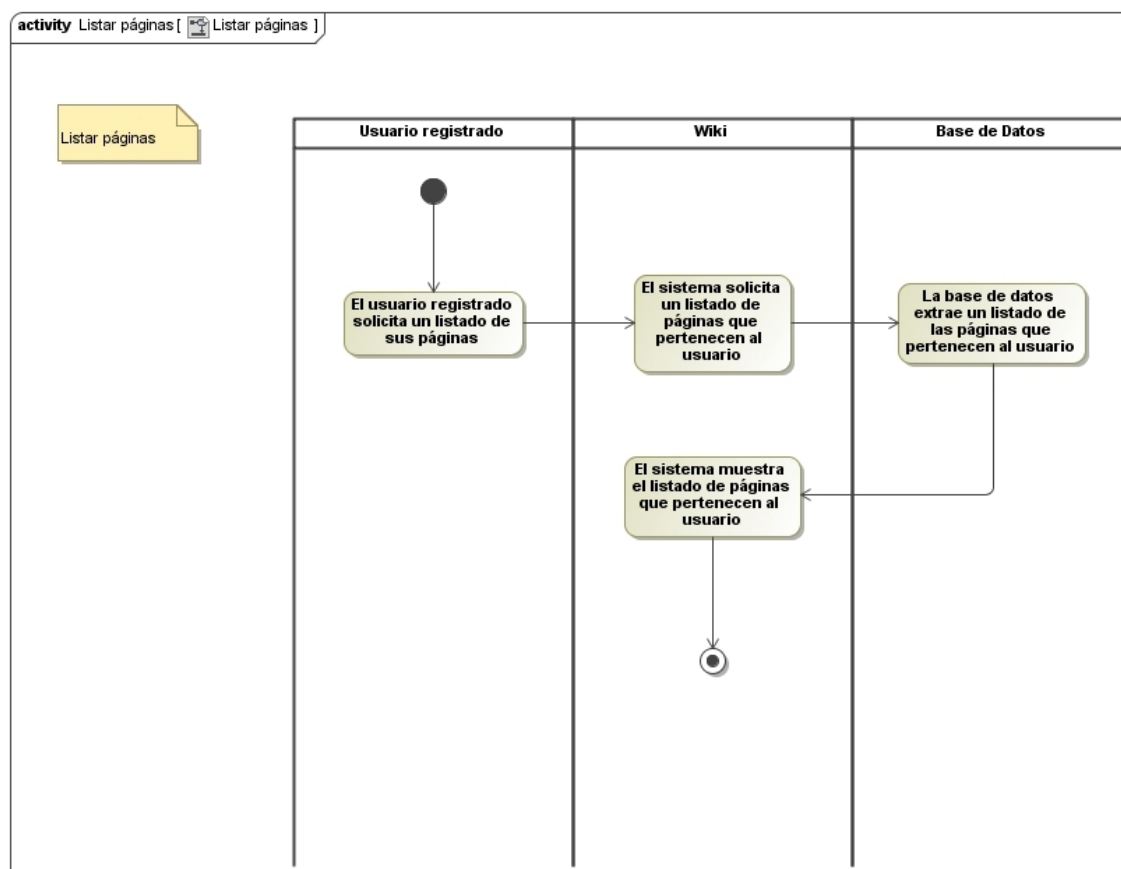


Figura 4.22: Diagrama de actividad listar páginas

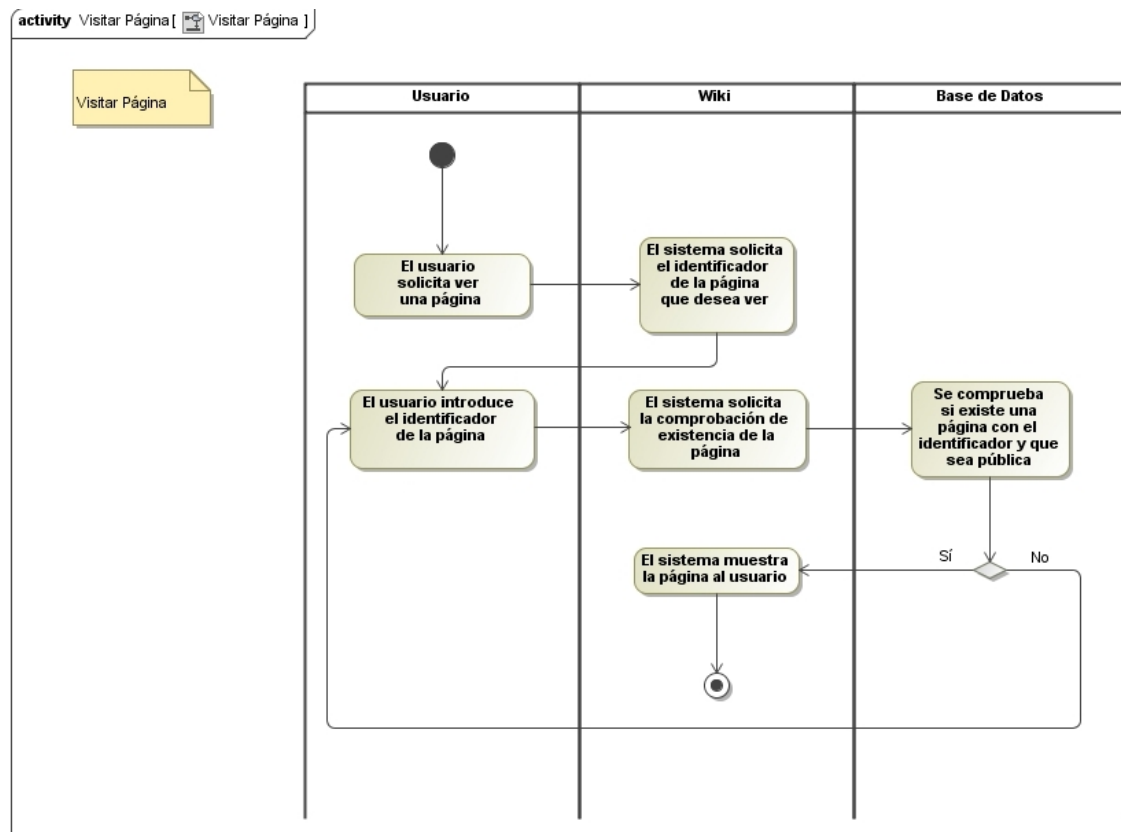


Figura 4.23: Diagrama de actividad visitar página

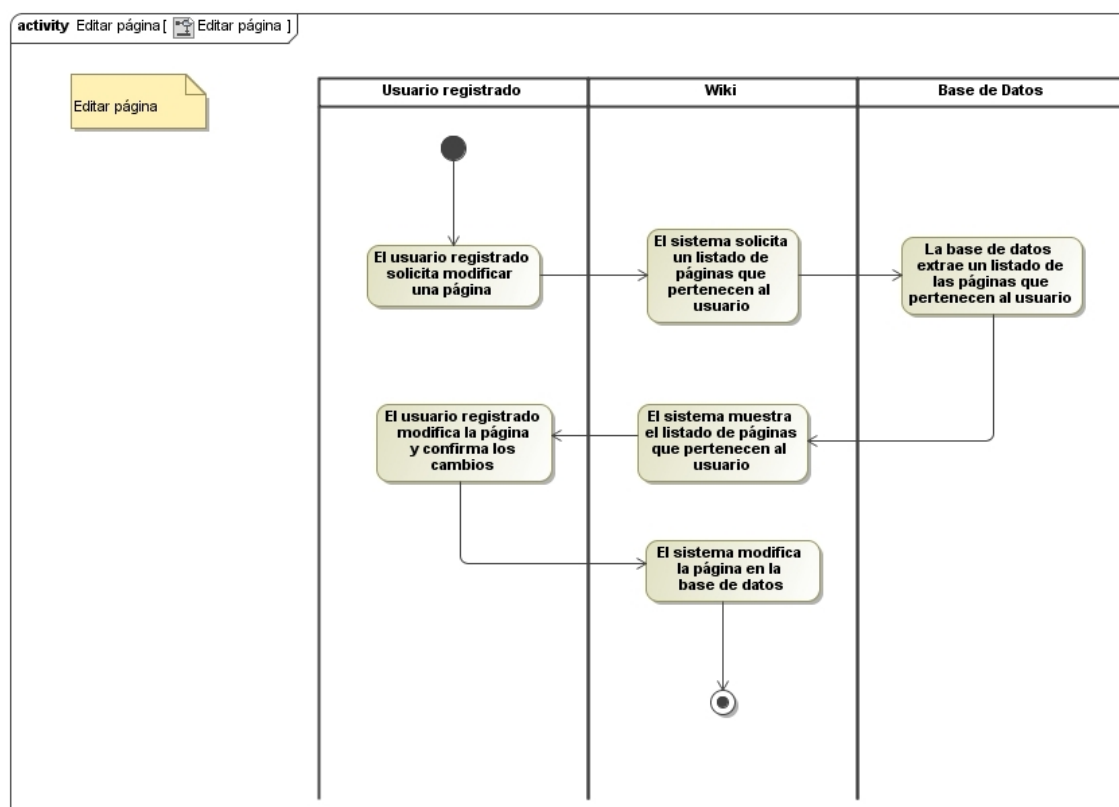


Figura 4.24: Diagrama de actividad editar página

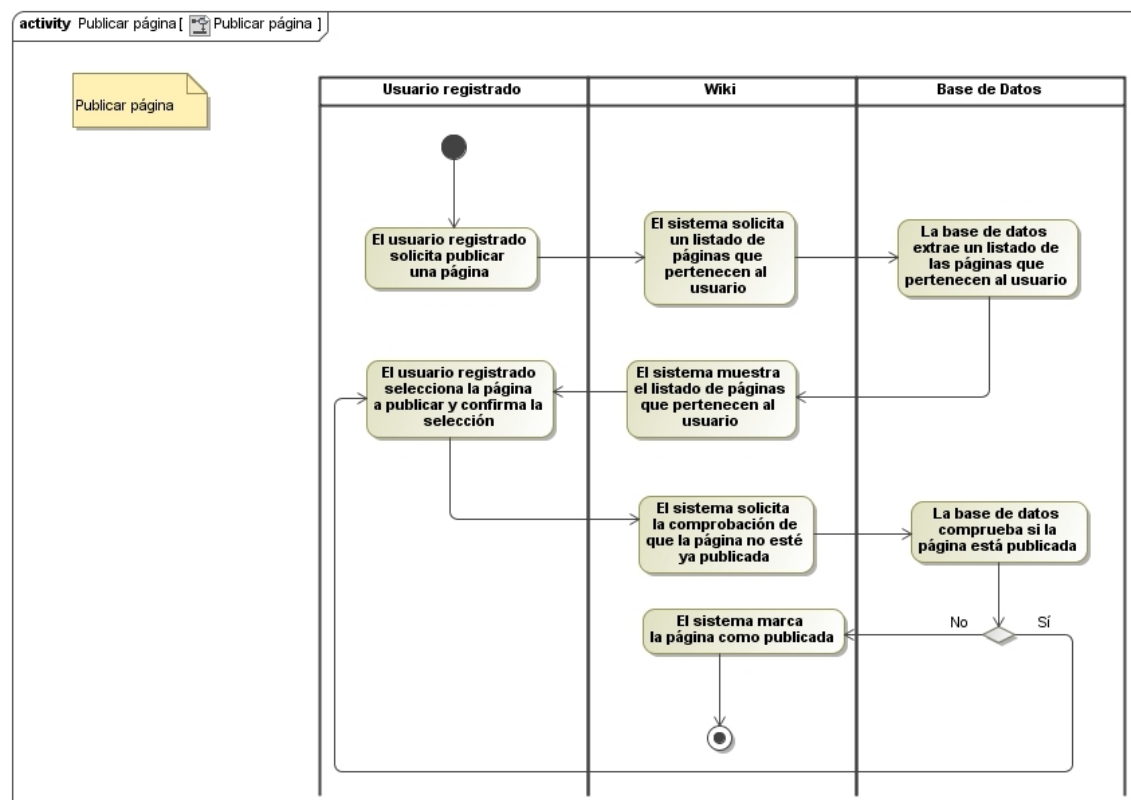


Figura 4.25: Diagrama de actividad publicar página

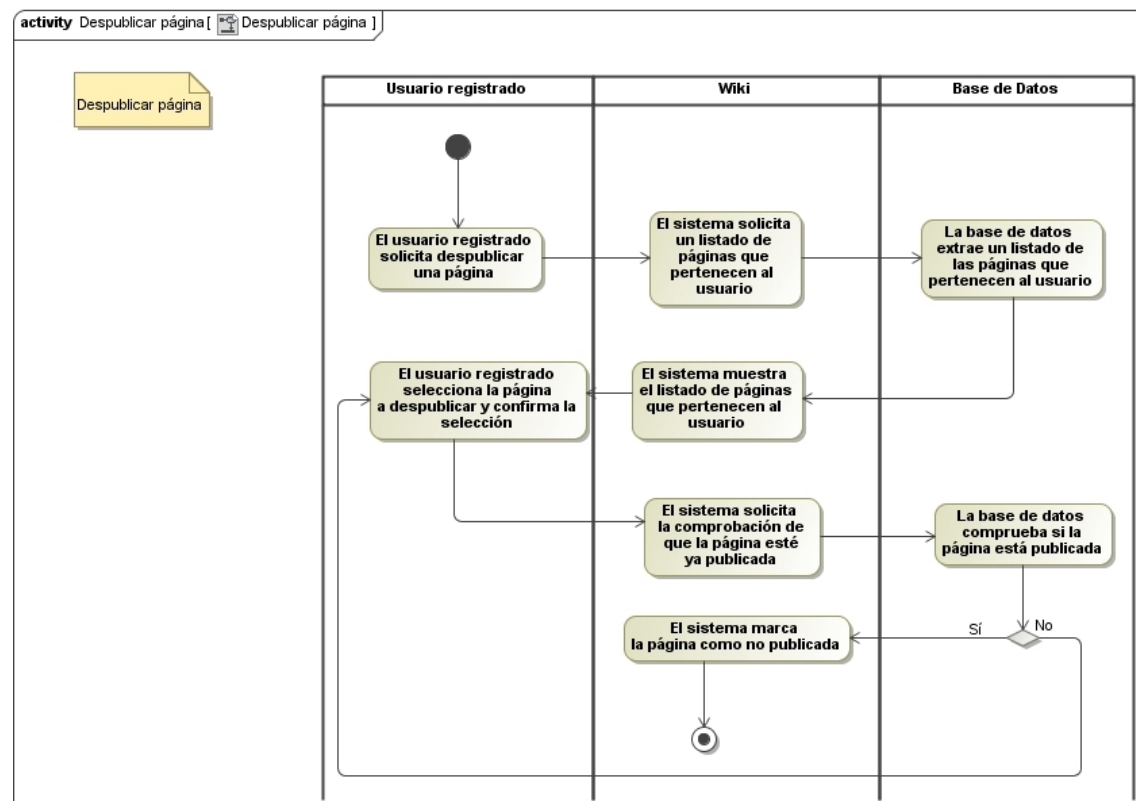


Figura 4.26: Diagrama de actividad despublicar página

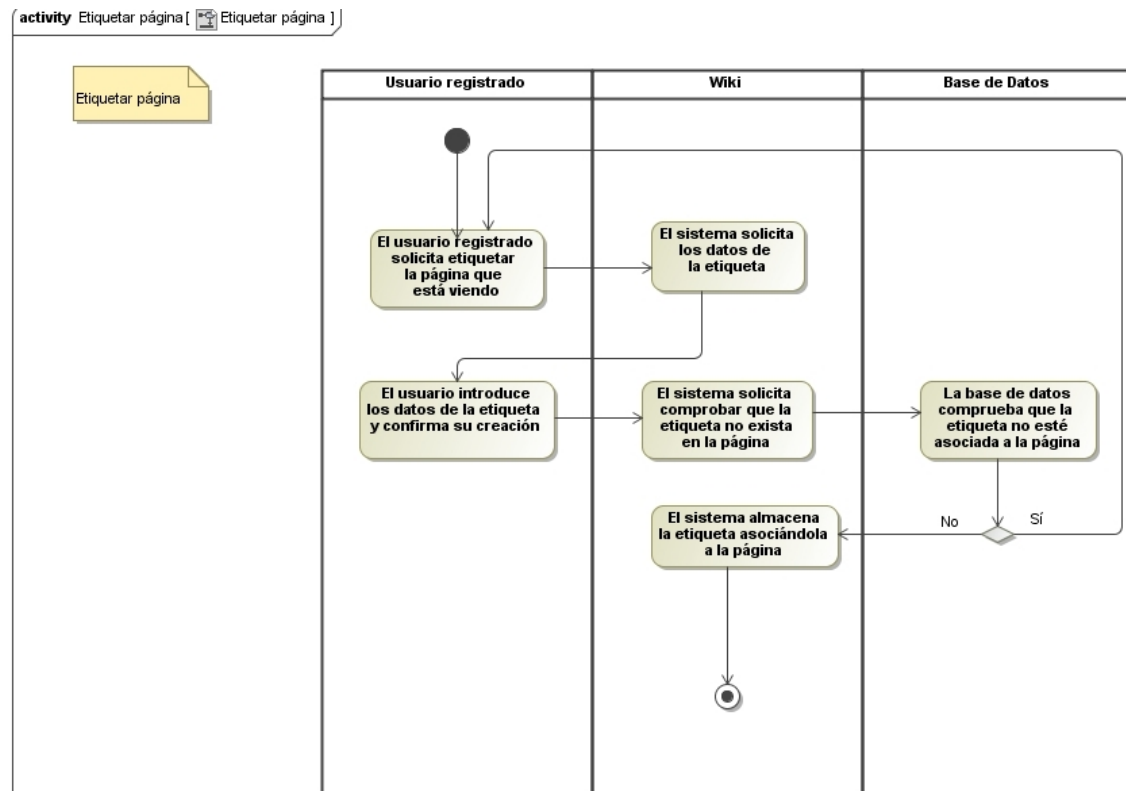


Figura 4.27: Diagrama de actividad etiquetar página

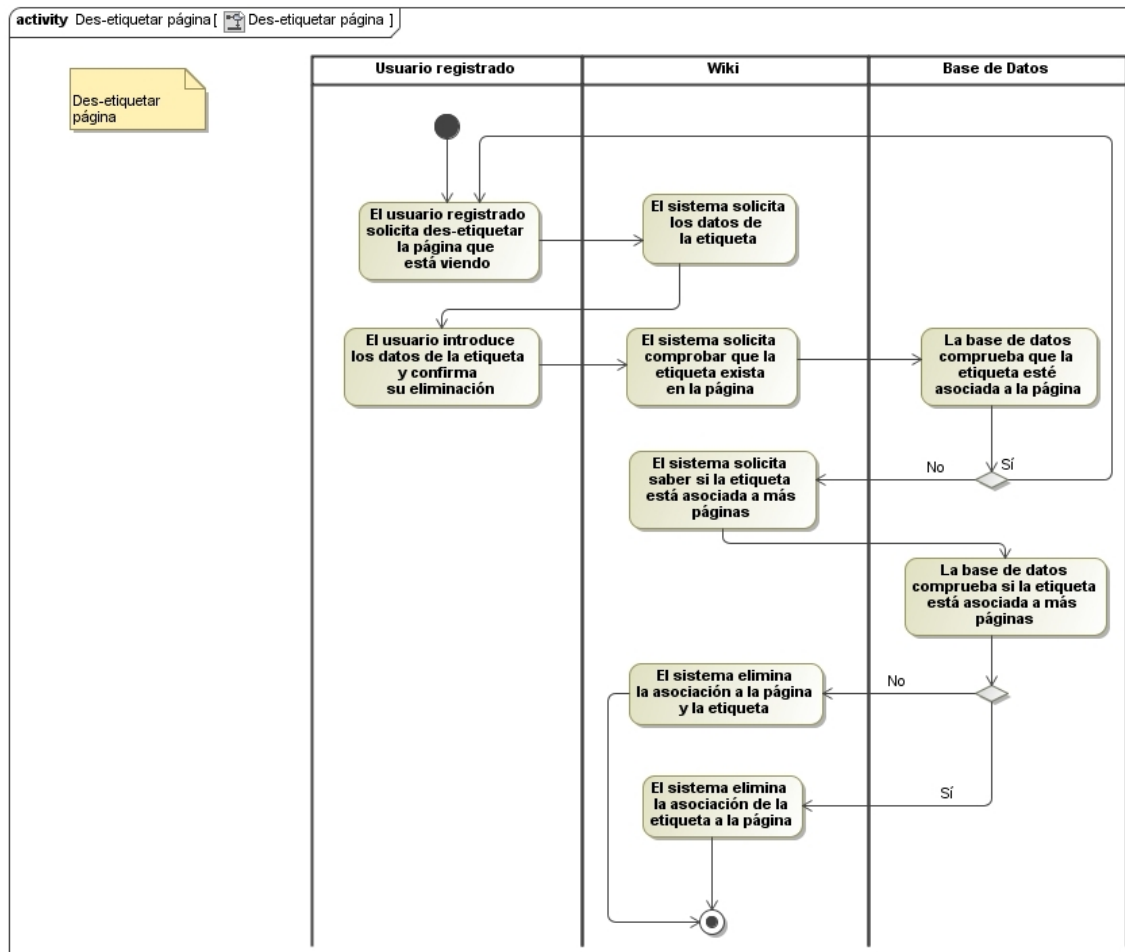


Figura 4.28: Diagrama de actividad des-etiquetar página

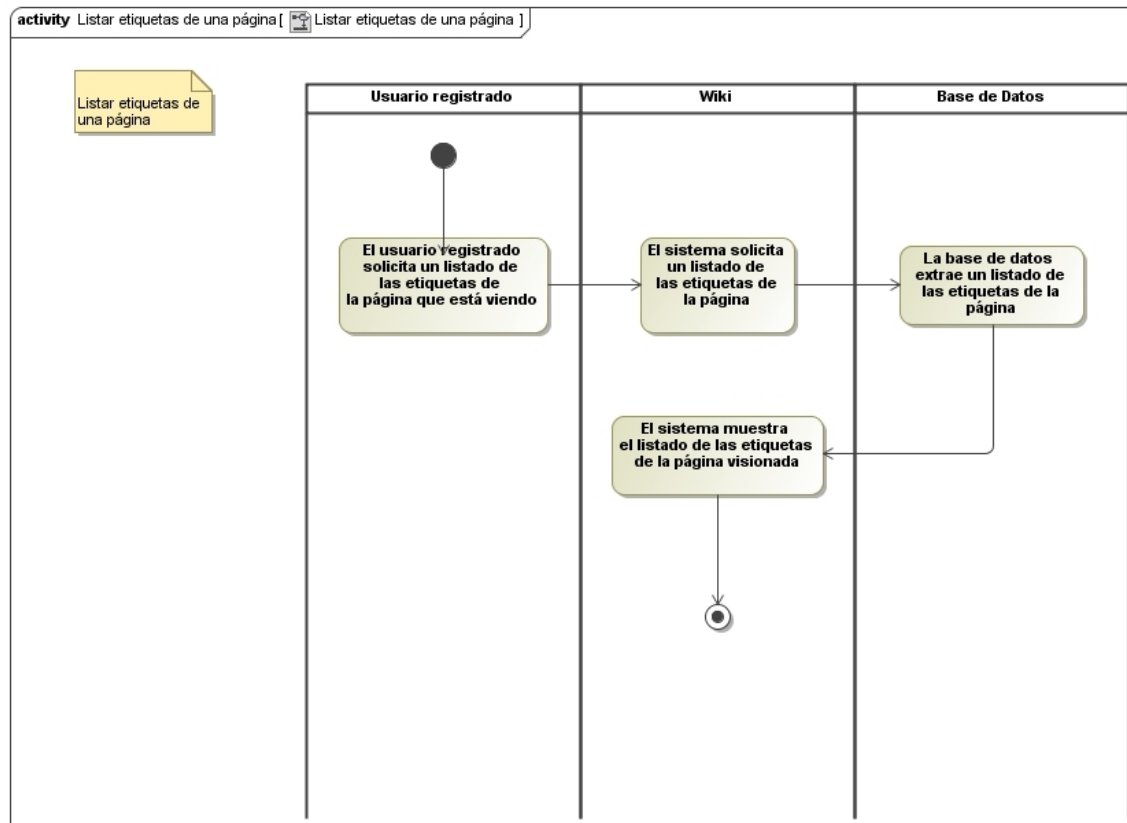


Figura 4.29: Diagrama de actividad listar etiquetas de una página

activity Publicar página administrador [ Publicar página administrador]

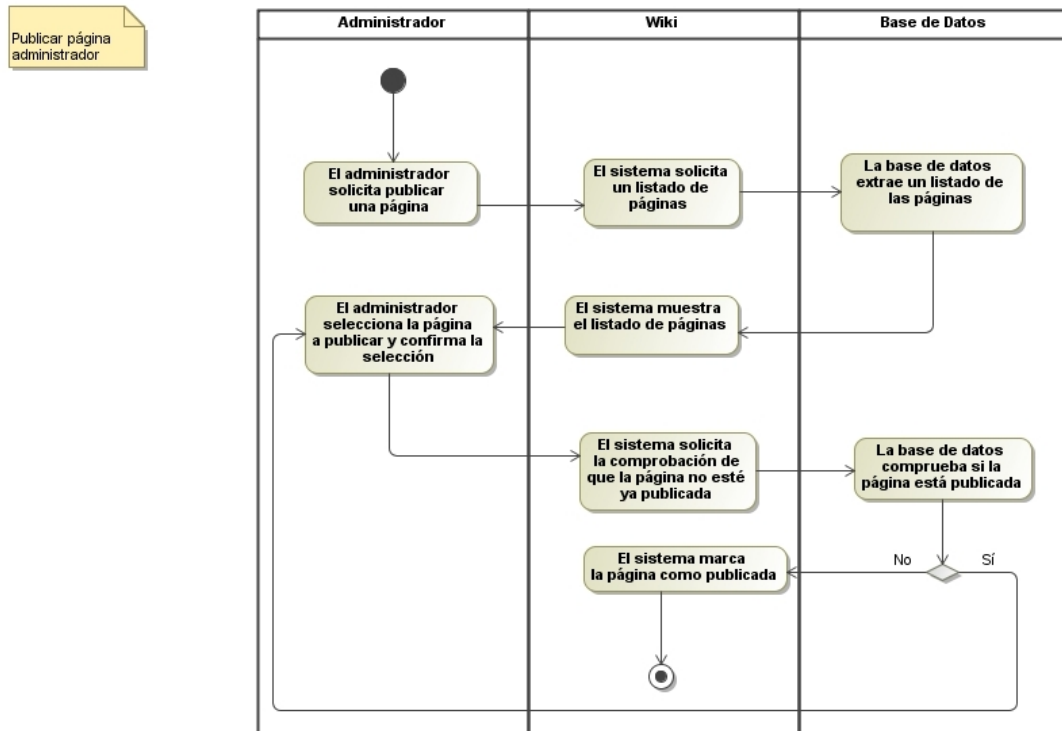


Figura 4.30: Diagrama de actividad publicar página administrador

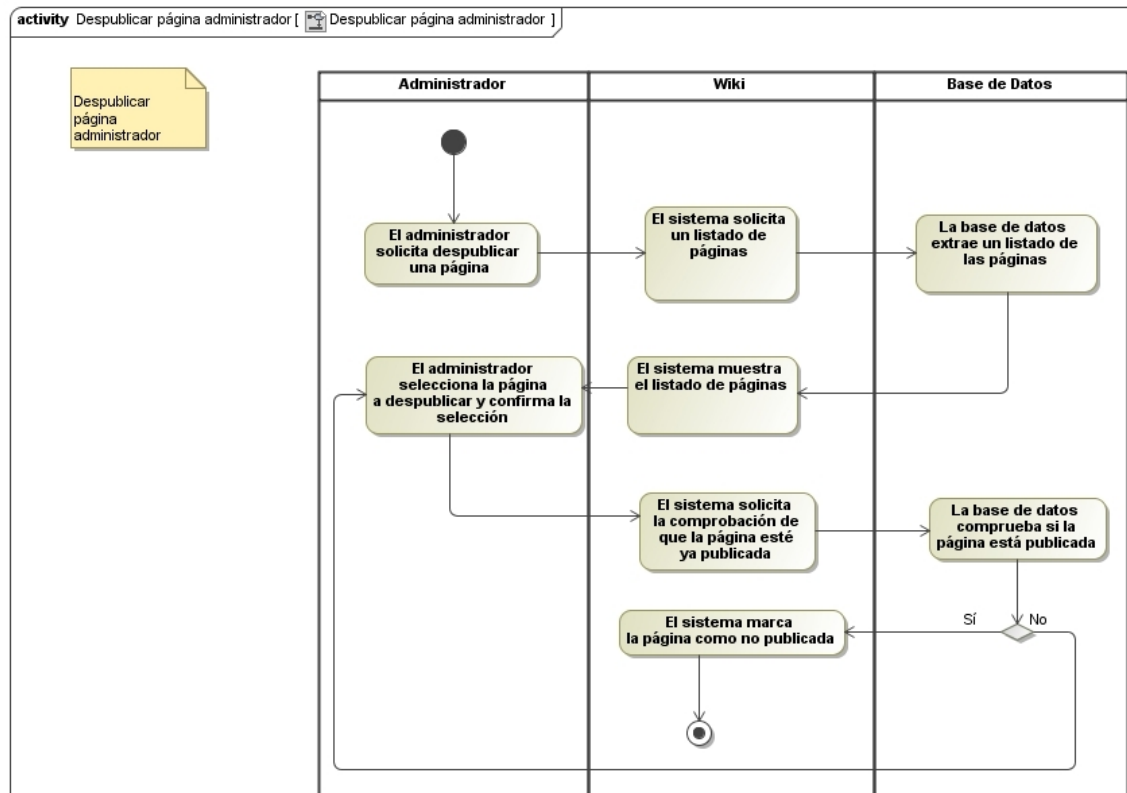


Figura 4.31: Diagrama de actividad despublicar página administrador

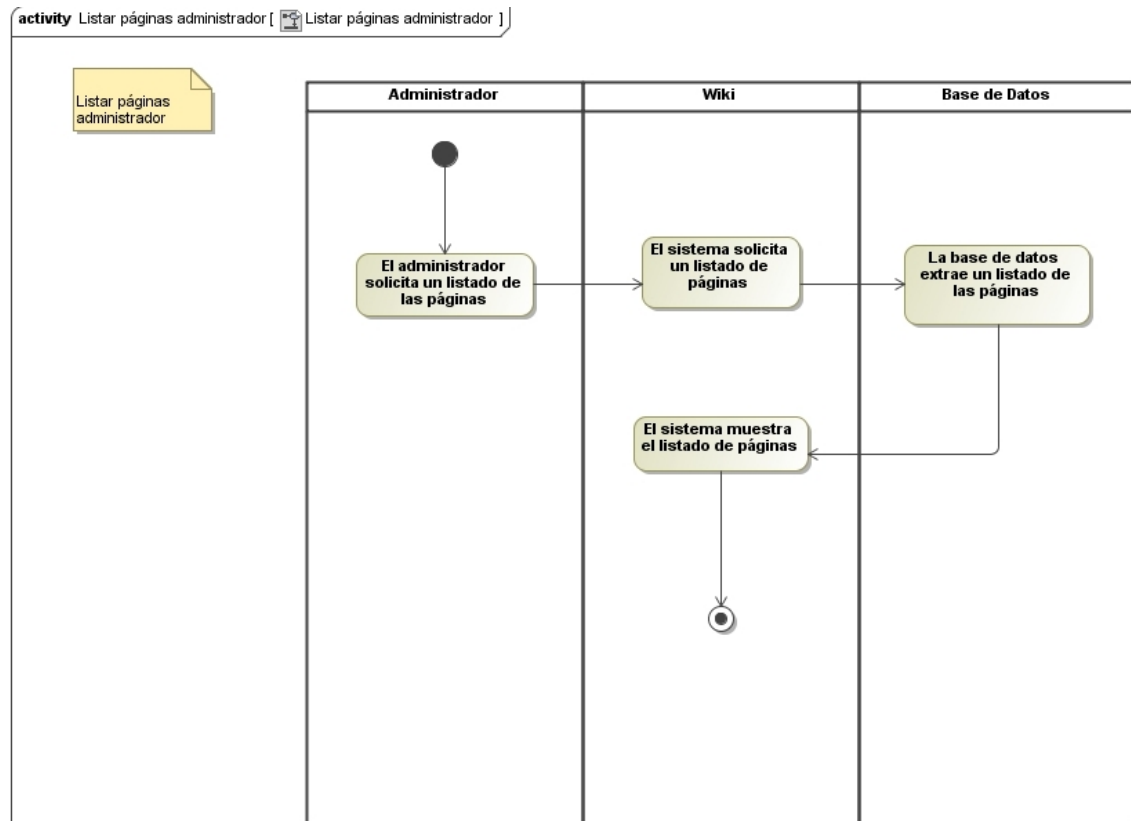


Figura 4.32: Diagrama de actividad listar páginas administrador

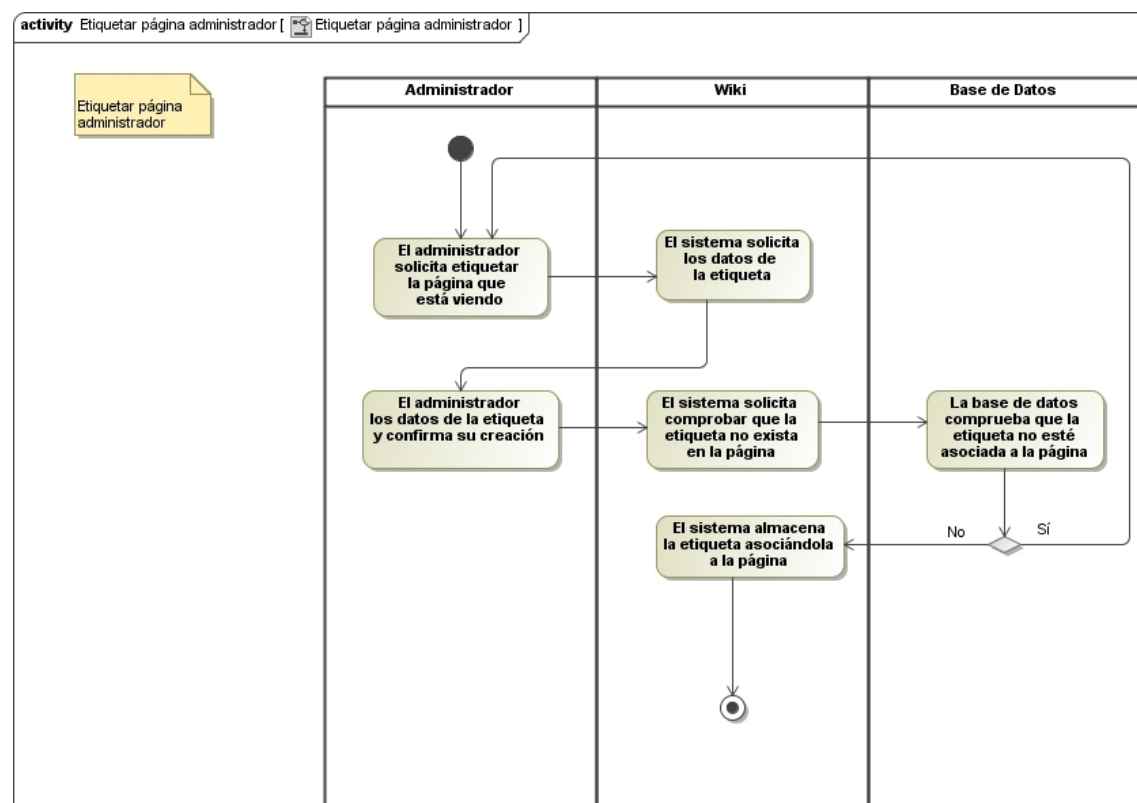


Figura 4.33: Diagrama de actividad etiquetar página administrador

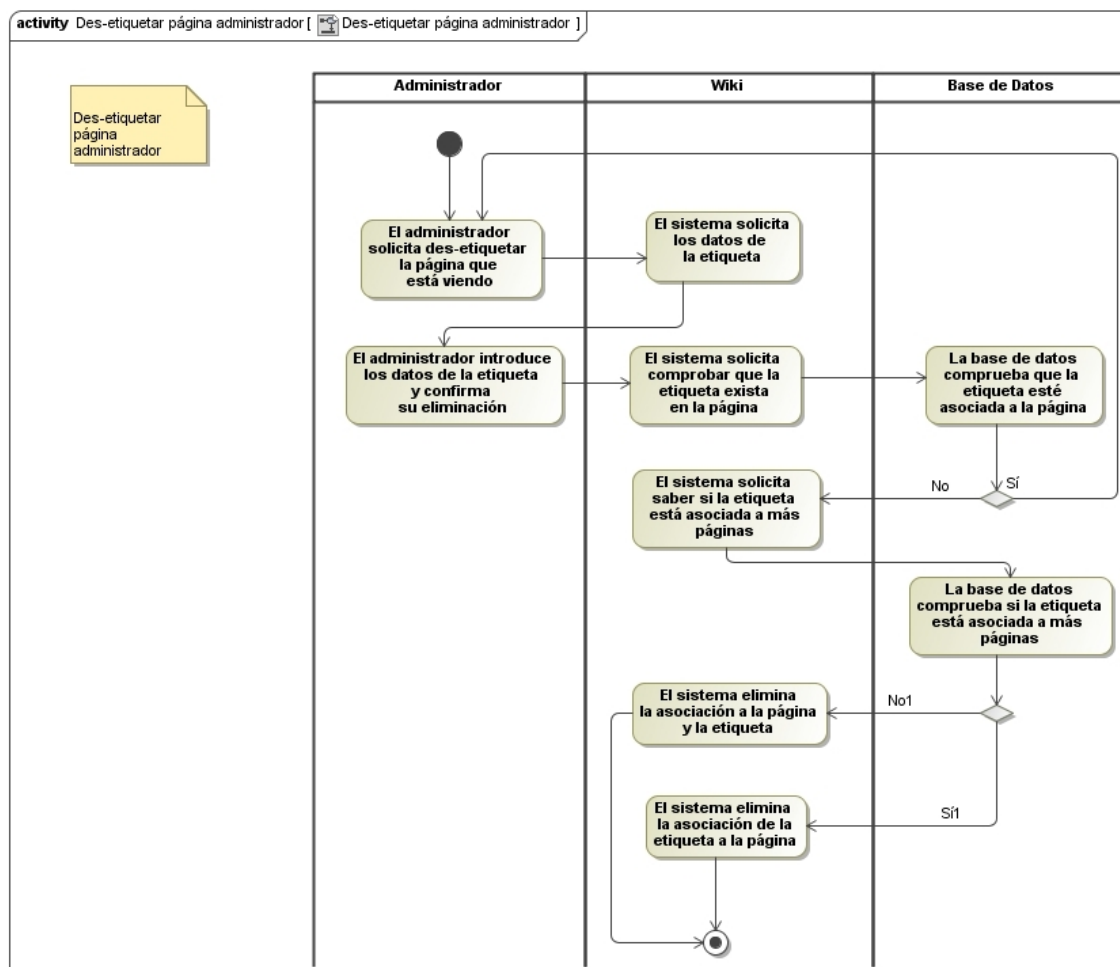


Figura 4.34: Diagrama de actividad des-etiquetar página administrador

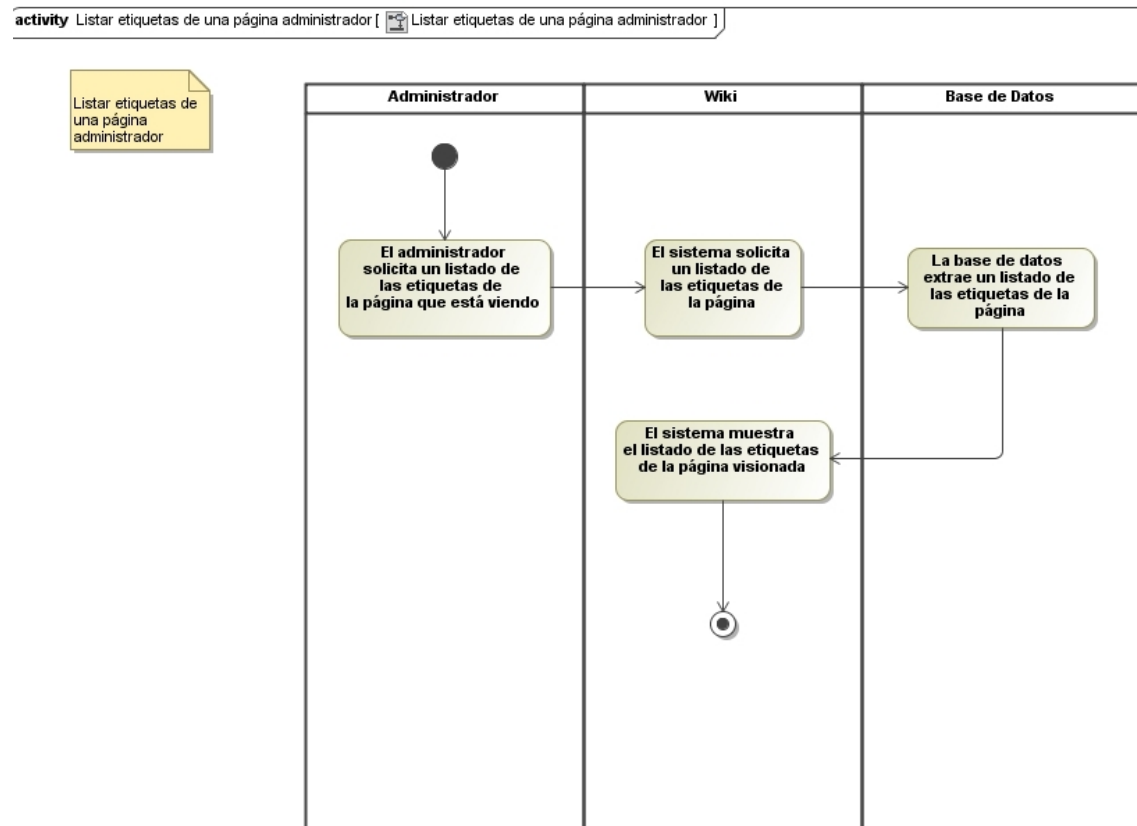


Figura 4.35: Diagrama de actividad listar etiquetas de una página administrador

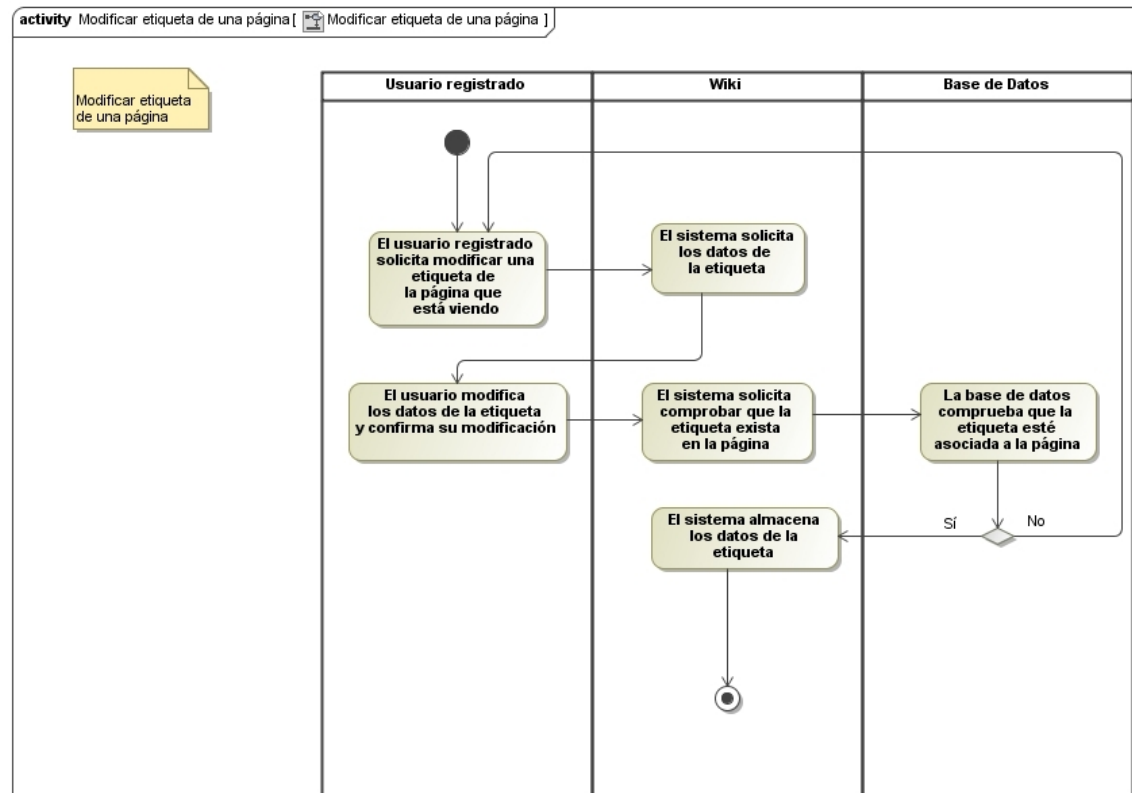


Figura 4.36: Diagrama de actividad modificar etiqueta de una página

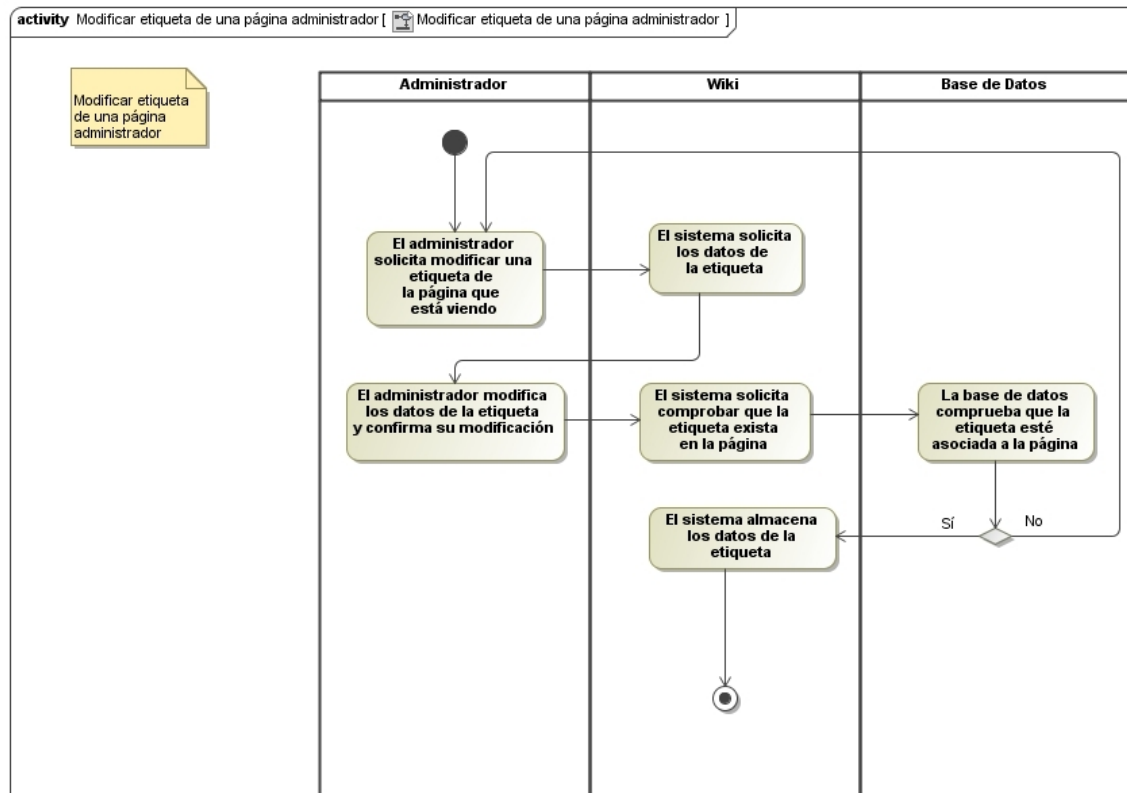


Figura 4.37: Diagrama de actividad modificar etiqueta de una página

4.3. Capturas de pantalla

Aquí unas capturas de pantalla de nuestra wiki.

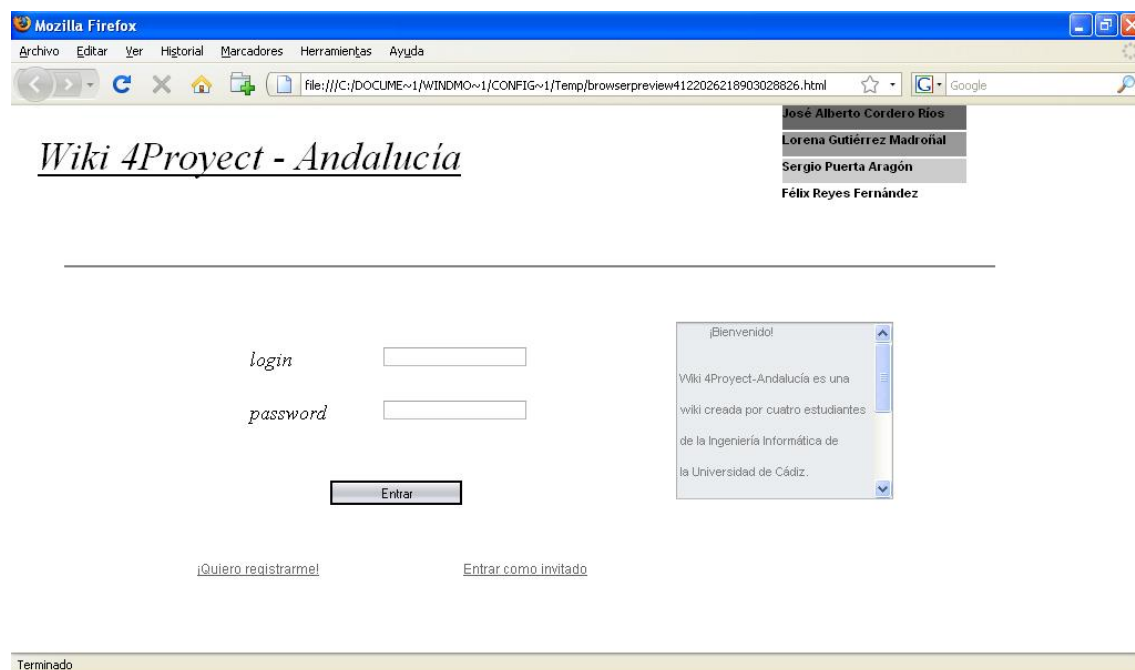


Figura 4.38: Pantalla de bienvenida a la Wiki

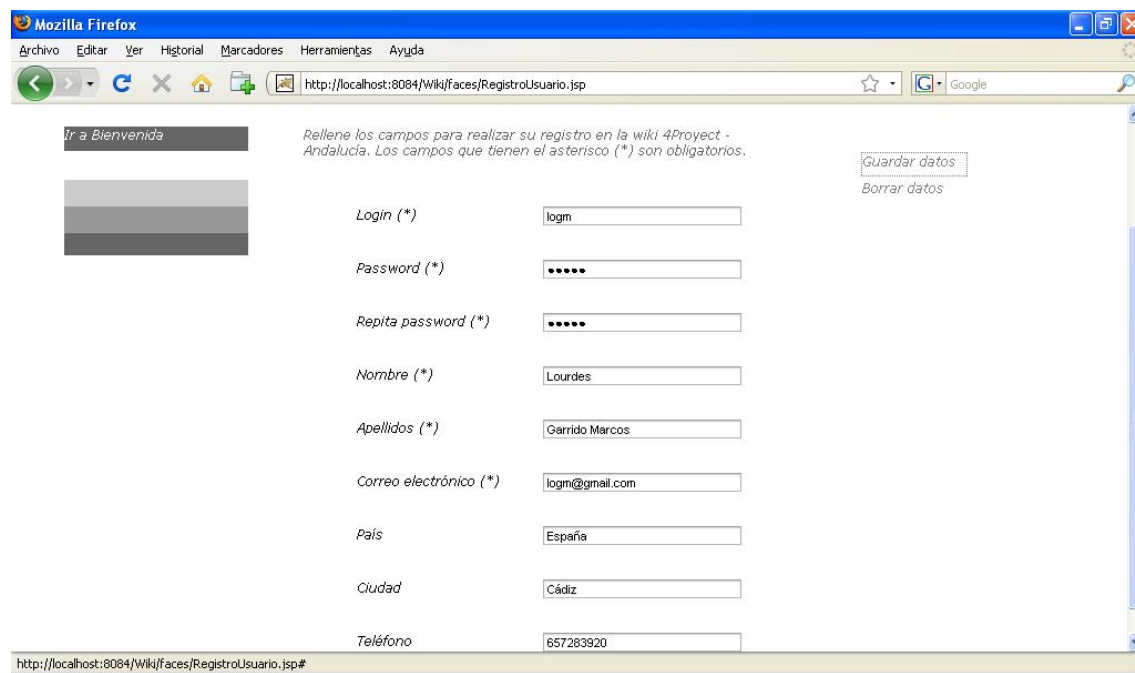


Figura 4.39: Registro en la wiki por parte de un usuario

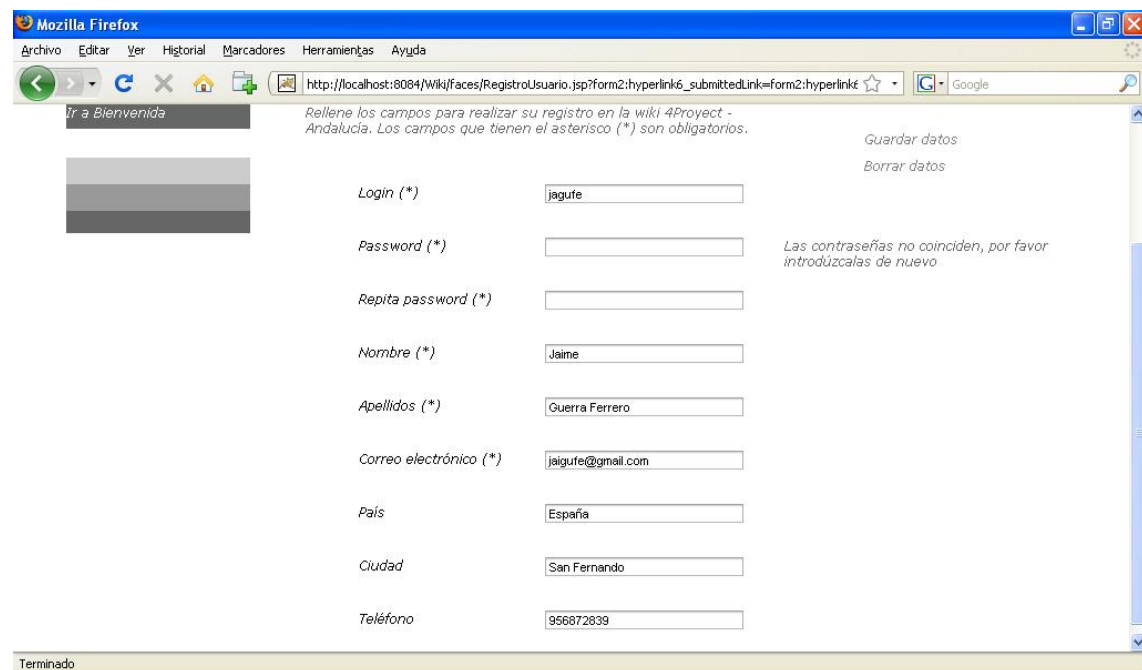


Figura 4.40: Registro fallido por parte de un usuario

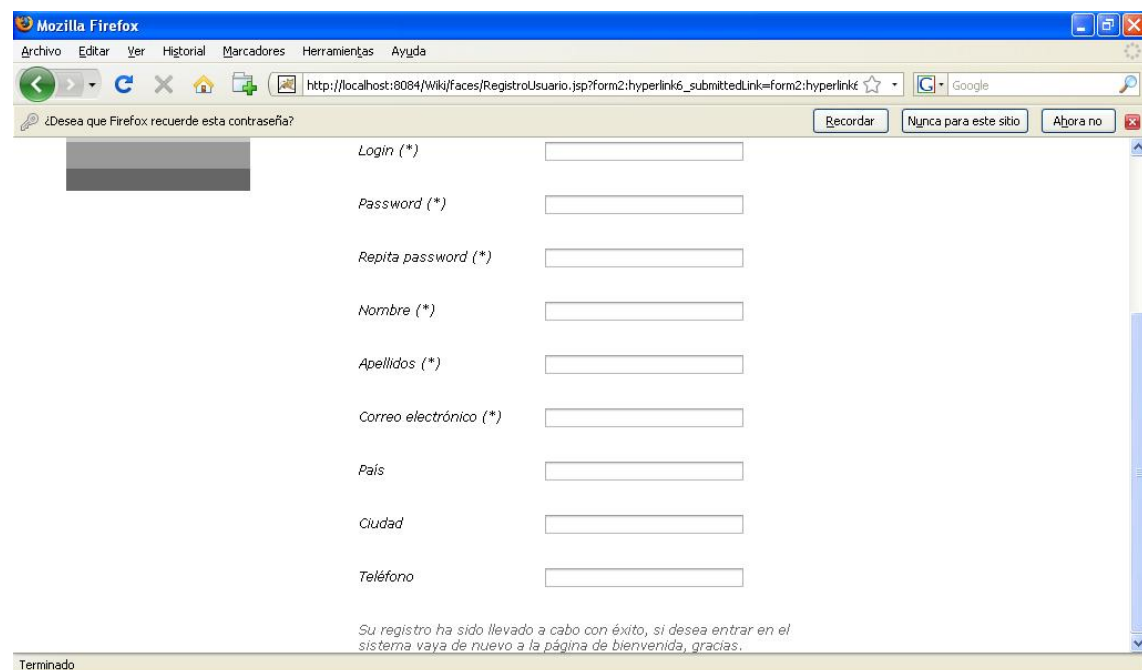


Figura 4.41: Registro correcto por parte de un usuario

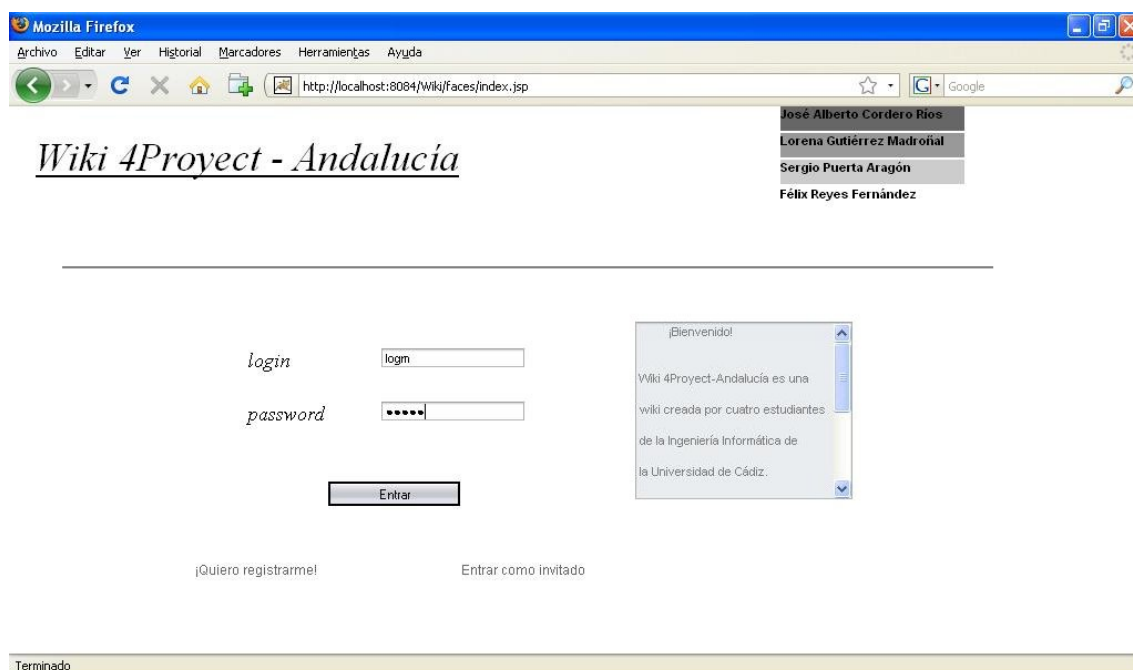


Figura 4.42: Login por parte de un usuario



Figura 4.43: Login fallido por parte de un usuario

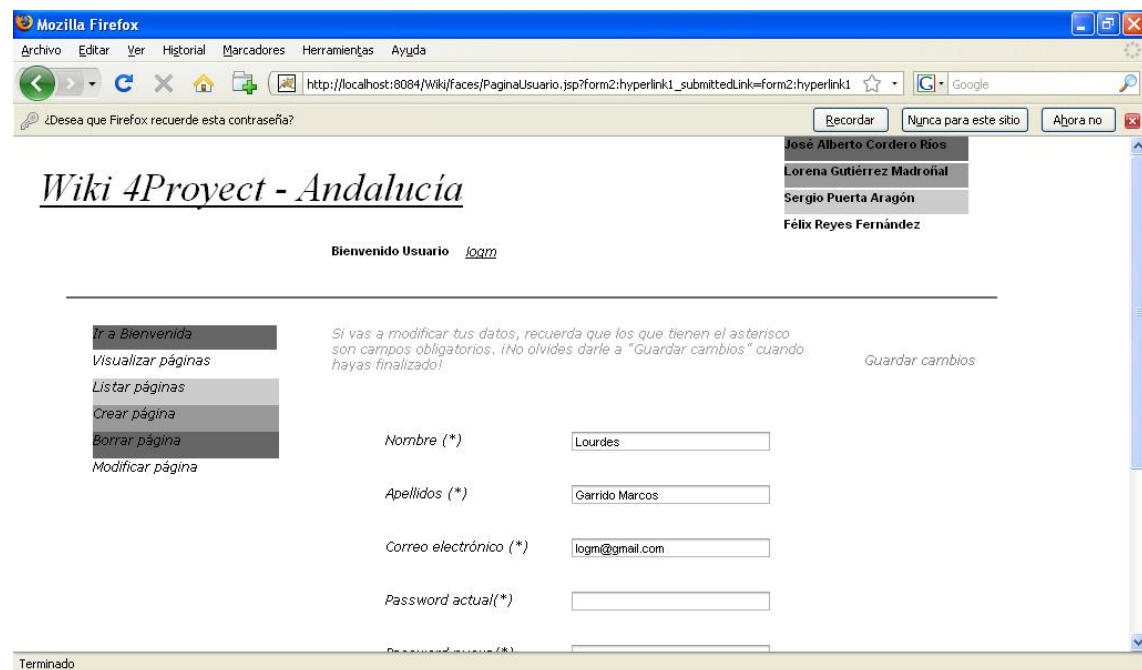


Figura 4.44: Página de inicio de un usuario registrado, cabecera

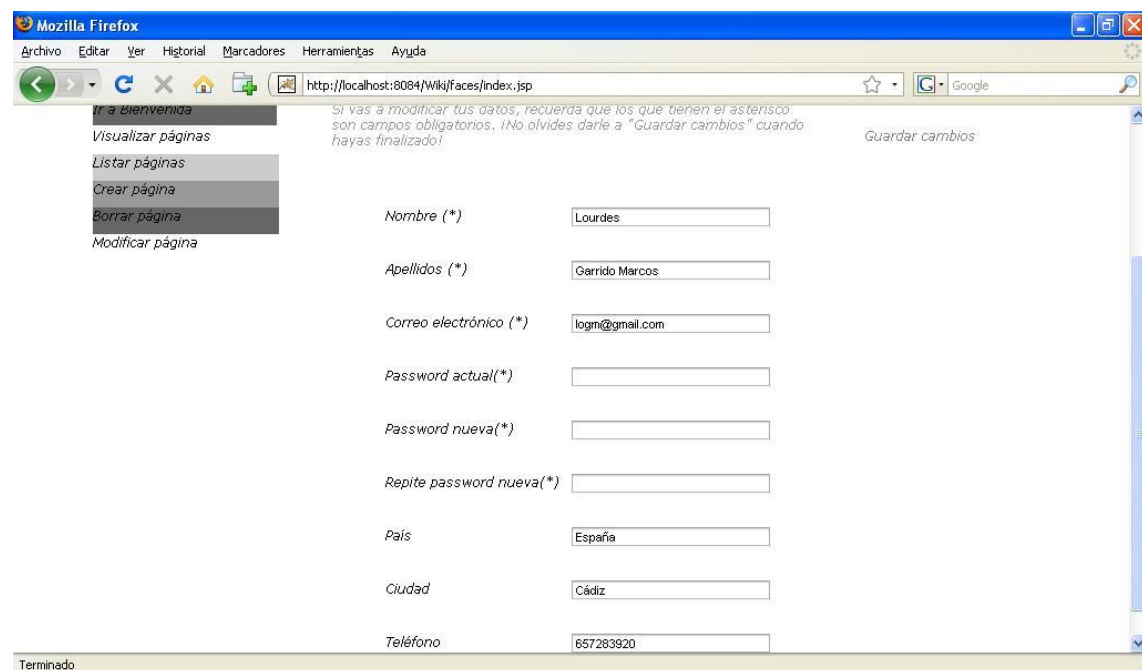


Figura 4.45: Página de inicio de un usuario registrado

Wiki 4Proyect - Andalucía

Bienvenido administrador de la wiki

José Alberto Cordero Ríos

Lorena Gutiérrez Madroñal

Sergio Puerta Aragón

Félix Reyes Fernández

[Ir a Bienvenida](#)

[Visualizar páginas](#)

[Listar páginas](#)

[Crear página](#)

[Borrar página](#)

[Modificar página](#)

[Listar usuarios](#)

Si vas a modificar tus datos, recuerda que los que tienen el asterisco son campos obligatorios. ¡No olvides darle a "Guardar cambios" cuando hayas finalizado!

[Guardar cambios](#)

Nombre (*)

Administrador

Apellidos (*)

Administrador

Password actual(*)

Password nueva(*)

Repite password nueva(*)

Correo electrónico (*)

administrador@wiki.com

País

Ciudad

Teléfono

Figura 4.46: Página de inicio del administrador

Si vas a modificar tus datos, recuerda que los que tienen el asterisco son campos obligatorios. ¡No olvides darle a "Guardar cambios" cuando hayas finalizado!

[Guardar cambios](#)

Nombre (*)

Lourdes

Apellidos (*)

Garrido Marcos

Correo electrónico (*)

logm@gmail.com

La clave introducida no coincide con la de esta cuenta

Password actual(*)

Password nueva(*)

Repite password nueva(*)

País

España

Ciudad

Cádiz

Teléfono

657283920

Terminado

Figura 4.47: Modificación por parte del usuario fallida

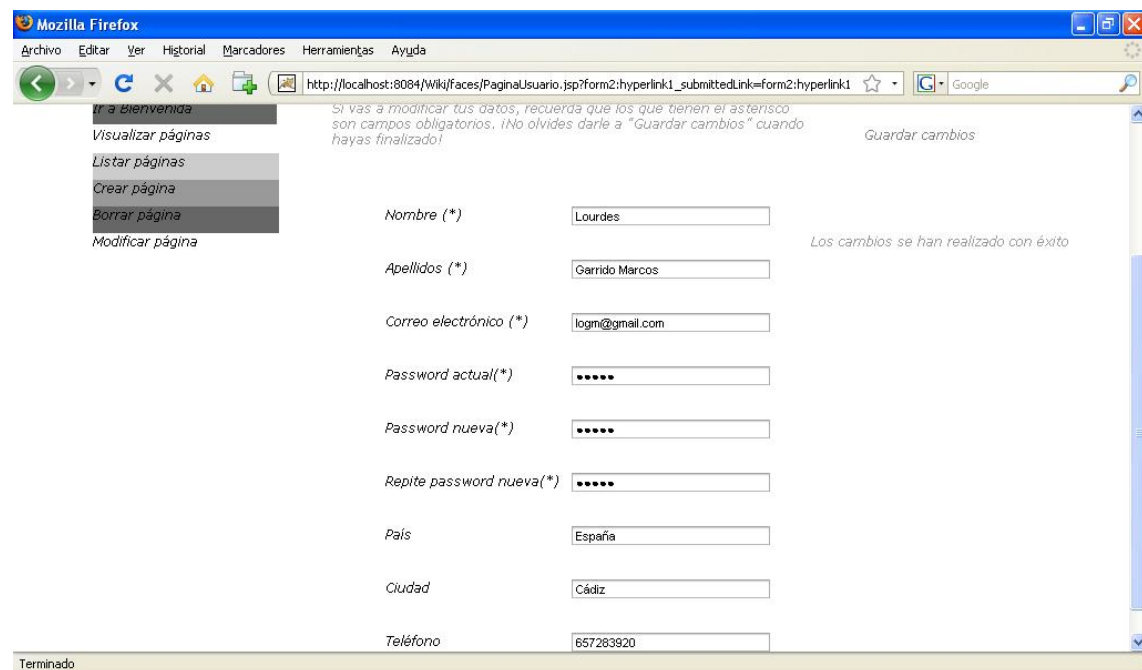


Figura 4.48: Modificación por parte del usuario satisfactoria

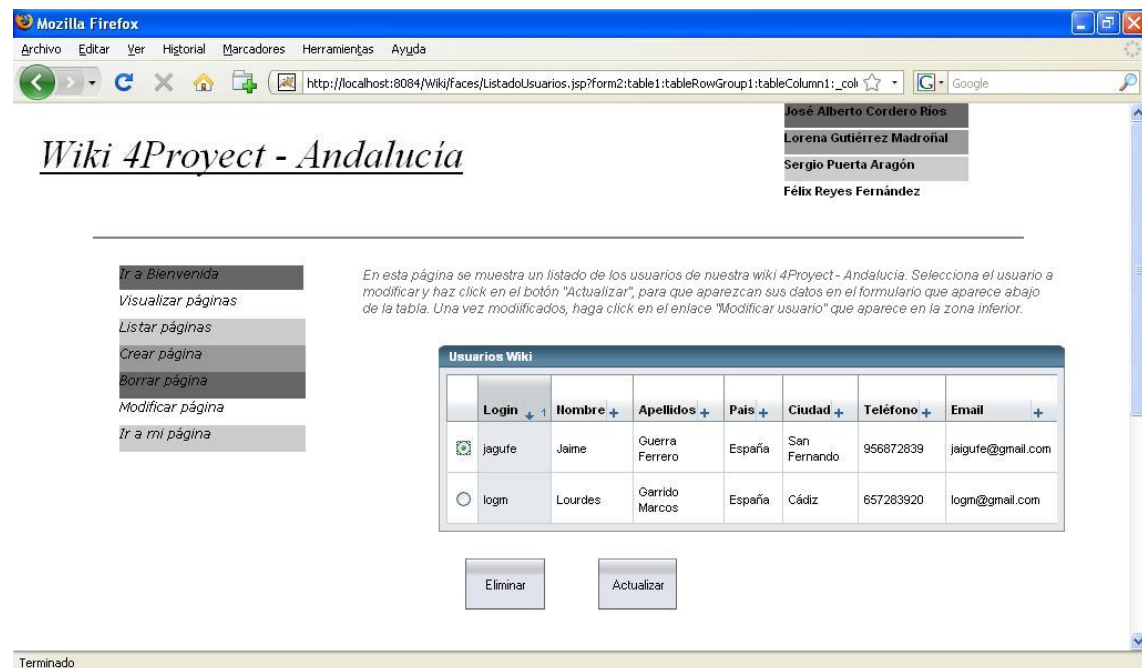


Figura 4.49: Listado de los usuarios registrados

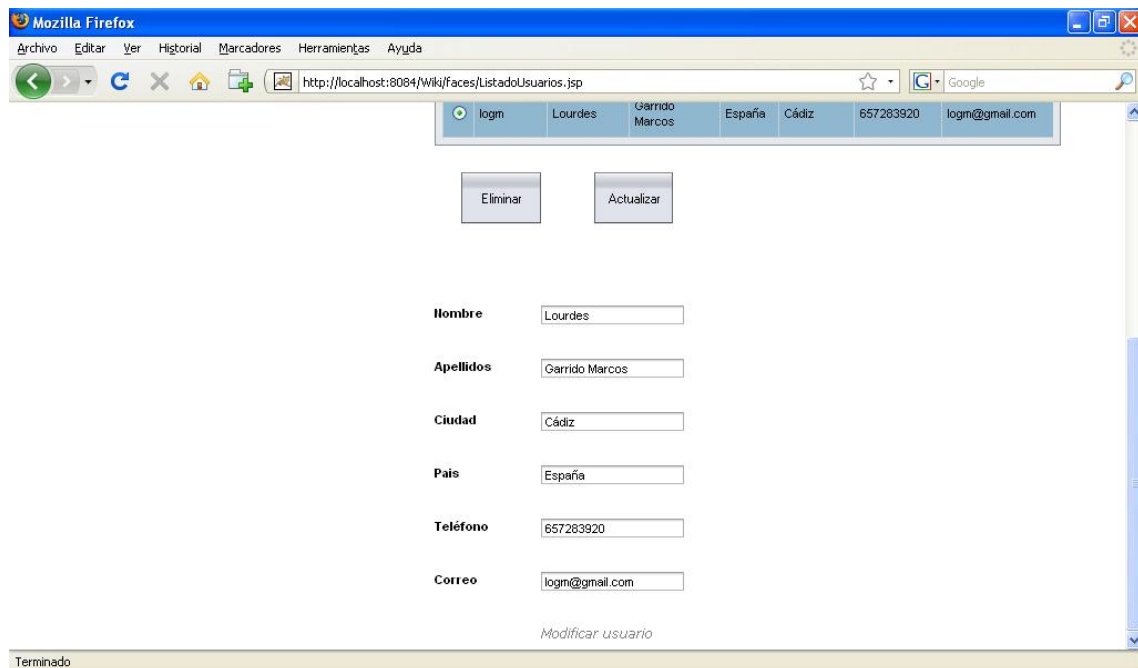


Figura 4.50: Listado de los usuarios registrados para modificar

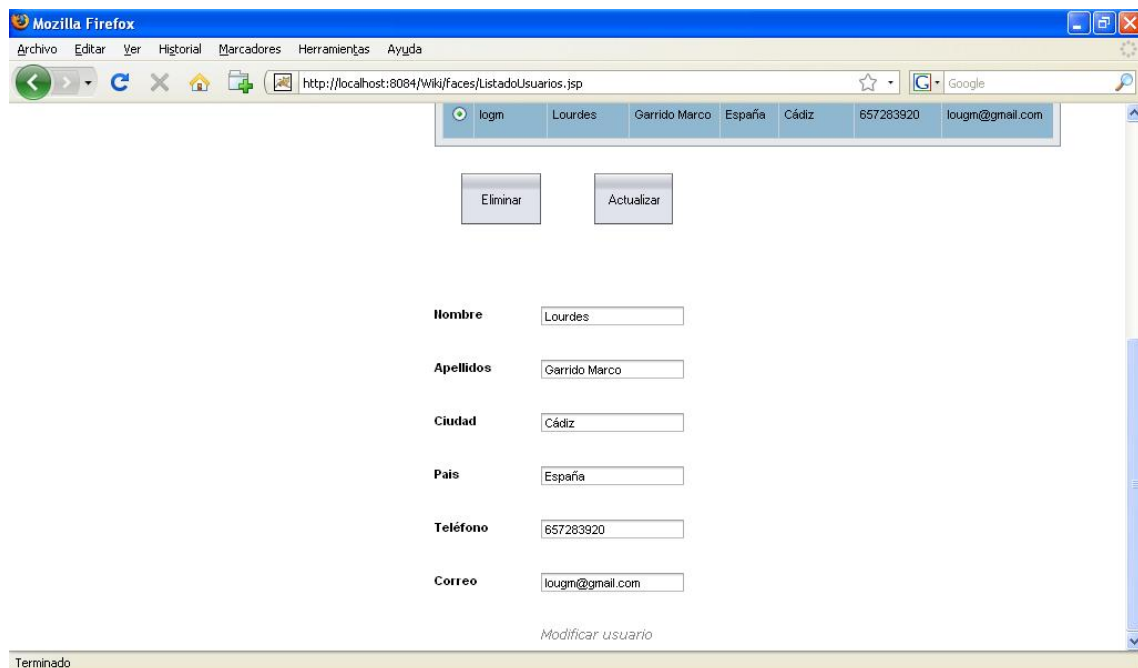


Figura 4.51: Listado de los usuarios registrados, modificando datos

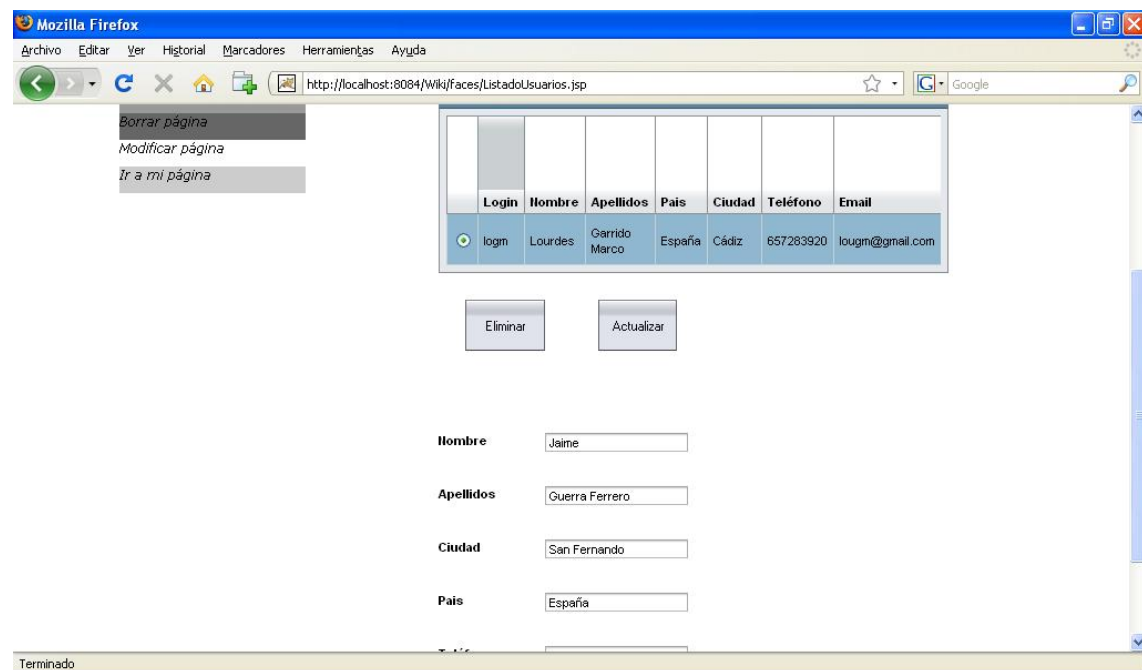


Figura 4.52: Listado de los usuarios registrados, eliminar usuario

5

Capítulo

Casos de prueba

6

Capítulo

Persistencia de datos

6.0.1. Esquema Entidad-Relación de la Base de Datos

A continuación se muestra el esquema E/R de nuestra Base de Datos:

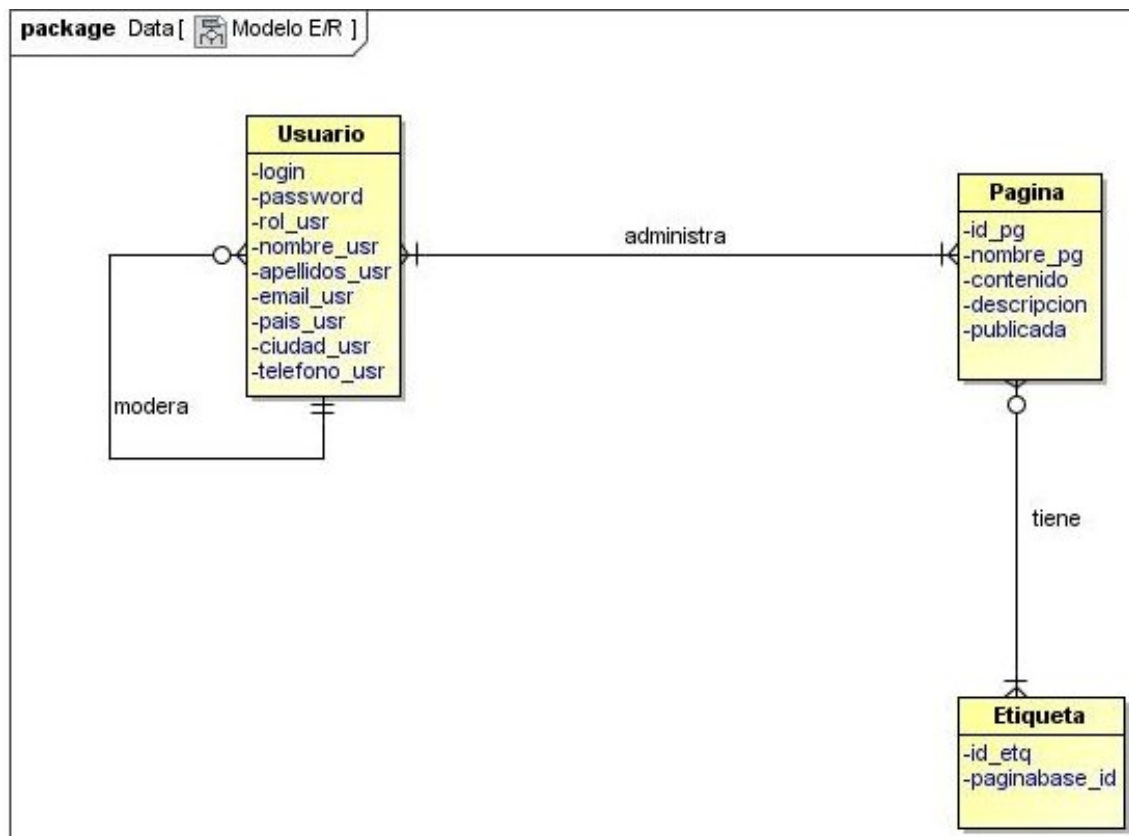


Figura 6.1: Diagrama E/R de nuestra Base de Datos

Este diseño genera el siguiente diseño arquitectónico del Mapeo Objeto-Relacional

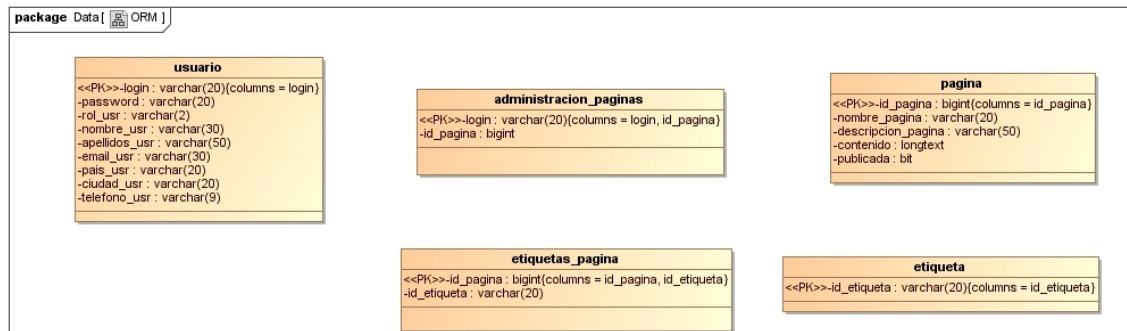


Figura 6.2: Mapeo Objeto-Relacional

Podemos ver la composición de cada una de las clases, que representan a cada tabla física de la base de datos. También se observa el tipo de cada atributo, así como las claves primarias de cada tabla.

6.1. Hibernate

6.1.1. Descripción

Como información previa para entender que es Hibernate, explicamos el concepto de mapeador objeto-relacional (ORM). La programación orientada a objetos y bases de datos relacionales son dos paradigmas diferentes. El modelo relacional trata con relaciones y conjuntos. Sin embargo, el paradigma orientado a objetos trata con objetos, sus atributos y asociaciones de unos a otros.

Un ORM es un sistema que permite almacenar objetos de aplicaciones Java en tablas de sistemas de bases de datos relacionales usando metadatos que describen la relación entre los objetos y la base de datos, y lo hace de una manera transparente y autónoma.

Es una tecnología que simplifica el acceso a base de datos, para ello permite establecer una correspondencia entre el modelo de la Base de Datos Relacional y una serie de clases que modelan los objetos de la aplicación. Con esta tecnología disponemos de un sistema de acceso a bases de datos relacionales de manera transparente, no es necesario estudiar un nuevo lenguaje, únicamente se deben escribir sentencias java, y de manera independiente, el código escrito con Hibernate funcionará en cualquier motor de datos al que se dé soporte. Hibernate se distribuye como una herramienta de software libre, distribuida bajo los términos de la licencia GNU LGPL.

Hibernate parte de una filosofía de mapear objetos Java, también conocidos como "POJOs" (Plain Old Java Objects). Con Hibernate no es necesario escribir código específico en nuestros objetos ni hacer que hereden de clases determinadas. En vez de eso trabajamos con archivos XML y objetos que proporciona la librería. Una de las principales características de Hibernate es su flexibilidad, envolviéndolo todo bajo un marco de trabajo común.

6.1.2. Características

Entre las principales características técnicas que aporta Hibernate están las siguientes:

- Modelo de programación natural. Hibernate es una capa de persistencia objeto/relacional que e permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos.
- Gran escalabilidad. Hibernate es muy eficiente, tiene una arquitectura de caché de doble capa y podría ser usado en un cluster.
- Hibernate admite los contextos de persistencia de larga vida denominados detach/reattach objetos. Además se ocupa del bloqueo optimista automáticamente.
- Software libre. Está bajo licencia LGPL (Lesser GNU Public License).
- EJB 3.0. Hibernate implementa la gestión de la API de la persistencia Java y el mapeado objeto-relacional.
- Persistencia transparente. Ofrece soporte para un amplio conjunto de las colecciones de Java, propiedades del estilo de persistencia de JavaBeans, etc. que abstraen al usuario.
- Mapeado flexible gracias a las asociaciones bidireccionales, la persistencia transitiva, colecciones de tipos básicos, etc. el mapeado resulta mucho más flexible.

- Facilidades en consultas. Debido en parte a que se realizan en un potente lenguaje de consultas orientado a objetos.
- Facilidades en metadatos. Soporta el formato del mapeado de XML, diseñado para ser editado a mano y el mapeado basado en anotaciones. Además de Validación basada en anotaciones.

Hibernate en su funcionamiento genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todas las bases de datos con un ligero incremento en el tiempo de ejecución. A la hora de realizar una optimización en las consultas, el hecho de generar el código en caliente hace que sea más fácil modificarlo y ajustarlo, cambiando un parámetro de true a false, pasamos de ejecutar todas las sentencias con y sin uniones de tablas, comparase esto con la otra posibilidad, que es cambiar todo a mano.

La influencia de Hibernate en la comunidad java ha sido tal, que gran parte de la especificación EJB3 está basada en él.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como *criteria*). Existe una ramificación de Hibernate conocida como Nhibernate que permite utilizar la misma filosofía en lenguaje C para su integración en la plataforma .NET.

Es una herramienta que funciona muy bien, gratuita, con una excelente documentación y con gran número de utilidades.

6.1.3. Consideraciones generales del funcionamiento

Para almacenar y recuperar objetos de la base de datos, el desarrollador debe mantener una conversación con el motor de Hibernate mediante un objeto especial, quizás el concepto clave más importante dentro de Hibernate, llamado Sesión (clase *Session*). Se puede equiparar a grandes rasgos al concepto de conexión de JDBC y cumple un papel muy parecido, es decir, sirve para delimitar una o varias operaciones relacionadas dentro de un proceso de negocio, demarcar una transacción y aporta algunos servicios adicionales como manejo de cache para evitar interacciones innecesarias contra la BD. En este sentido veremos que la clase *Session* ofrece métodos como *save* (*Object object*), *createQuery* (*String queryString*), *beginTransaction()*, *close()*, entre otros.

Para interactuar con la base de datos tal como estamos acostumbrados a hacerlo con una conexión JDBC, pero con una diferencia: mayor simplicidad, es decir, guardar un objeto, por ejemplo, consiste en algo así como *session.save(miObjeto)*, sin necesidad de especificar una sentencia SQL, y esto es sólo un ejemplo muy trivial en el que ganamos relativamente poco utilizando Hibernate.

Hibernate distingue entre objetos tipo *transient* y tipo *persistent*: los primeros son objetos que sólo existen en memoria y no en un almacén de datos, en algunos casos no serán almacenados jamás en la base de datos y en otros es un estado en el que se encuentran hasta ser almacenados en ella. Los segundos se caracterizan por haber sido ya almacenados y ser por tanto objetos persistentes. Dicho de otra manera los objetos *transient* han sido instanciados por el desarrollador sin haberlos almacenado mediante una sesión, los objetos persistentes han sido creados y almacenados en una sesión o bien devueltos en una consulta realizada con la sesión.

Igual que con las conexiones JDBC, tenemos que crear y cerrar sesiones, aunque no hay una relación 1:1 entre sesiones y conexiones, es decir, no tenemos que abrir y cerrar simultáneamente sesiones y conexiones JDBC, la política a seguir dependerá del contexto del proceso de negocio

de cada situación dándonos Hibernate amplias posibilidades para la implementación de nuestras políticas (conexiones JDBC gestionadas por la aplicación, por Hibernate, por un posible servidor de aplicaciones, etc), siendo solamente necesario en la práctica crear y cerrar explícitamente las sesiones de Hibernate.

6.1.4. Ventajas e inconvenientes

Hay un conjunto de factores que hacen de Hibernate una potente herramienta:

Productividad: Usado conjuntamente con otras herramientas reducirán significativamente el tiempo de desarrollo.

No obstante, donde verdaderamente Hibernate es potente, es en las herramientas que pone a disposición. Es posible, mediante tareas de ant, partiendo de los objetos, y ficheros de configuración, crear todo el modelo de datos, y lo que es mejor, con las sentencias adecuadas a cada motor. Incluso en la situación contraria, aún mejor, partiendo del modelo de datos, Hibernate es capaz de generar todos los ficheros de configuración y los objetos asociados. Y si hacemos las cosas bien y diseñamos nuestros objetos de negocio de la manera correcta y concienzuda, con Xdoclet, podemos generar los ficheros de configuración, y de ahí la base de datos. En caso de utilizar java 5 no serían necesarios ningún tipo de fichero de configuración, bastaría con anotaciones. Utilizar un framework de ORM simplifica enormemente la programación de lógica de persistencia. Se trata de una idea completamente madura que cada vez se vuelve más popular. En aplicaciones donde la lógica de negocios trabaja contra un modelo de dominio completamente orientado a objetos la generación de código se reduce en entre un 30 % y un 40 %. Además el código generado suele ser mucho más sencillo y mantenible.

Mantenibilidad: Al tener menos líneas de código hace que el sistema sea más comprensible y nos centramos en la lógica de negocio en lugar de en la conexión. Un sistema con menos código es más fácil de refactorizar.

Rendimiento: La mayoría de las optimizaciones son más fáciles disponiendo de un ORM así algunos aspectos como queries son más fáciles de optimizar que con código manejado vía SQL/JDBC

Independencia del Vendedor: El motor abstrae a la aplicación de la Base de Datos subyacente y del dialecto SQL. Si la herramienta soporta un número distinto de base de datos, esto proporciona un cierto nivel de portabilidad. Hibernate es agnóstico respecto al pool de conexiones, si no se especifica ninguno, usará uno interno, que para los casos de desarrollo es más que suficiente. Por defecto, Hibernate usa el pool C3P0, pero el pool de jakarta Commons DBCP, funciona perfectamente. Y si es agnóstico respecto a el pool, también lo es respecto a la base de datos, con una propiedad, cambiando una simple propiedad, tenemos nuestra aplicación lista para funcionar en PostgreSQL, MySQL, HSQL, Derby, etc.

Flexibilidad: Hibernate puede integrarse en una arquitectura J2EE. Su flexibilidad permite configurarlo no sólo con JDBC, sino también con JNDI, JTA y JMX. Es tan flexible su configuración que es válida para su uso como aplicación independiente o usada dentro de un servidor de aplicaciones. Si este fuera el caso, podríamos usar el motor de transacciones del servidor, y si como aplicación independiente usaríamos el del controlador jdbc.

Versatilidad en el acceso a BD relacionales: Hibernate, sobre cualquier otro sistema de acceso a base de datos relacionales, ofrece distintas formas de hacer la misma cosa. Por ejemplo, las consultas se pueden hacer de cuatro formas:

- **HQL.** Hibernate Query Language. Es un lenguaje de acceso a base de datos en donde las consultas se hacen a los objetos, no a las tablas. No es complicado de aprender y ofrece una gran potencia y simplifica los accesos. Aunque es un lenguaje propietario, sólo vale para Hibernate, el próximo EJB-QL está, en gran parte, basado en él, por lo que merece la pena aprenderlo.

- QBE. Query by Example. Simplemente, crea un objeto, rellena un campo con datos, y se crea una sentencia SQL con ese criterio.
- QBC. Query by Criteria. Parecido al anterior, pero para criterios no sólo de igualdad. Con código se crean comparaciones, sentencias tipo like. Ideal para buscadores. La cantidad de tiempo y código que ahorra es digna de mención.
- SQL. En Hibernate es posible ejecutar sentencias SQL, por si alguna vez se tuviera la imperiosa necesidad de hacerlo. En este sentido, podríamos trabajar con Hibernate como si fuera iBatis.

En el caso de decidirse por HQL, o SQL, para acceder a los datos, no es necesario escribir las sentencias en el código, podemos materializarlas en los ficheros de configuración, darles un identificador y referirnos a ellas por este nombre que le acabamos de otorgar. También proporciona herramientas para ejecutar sentencias HQL, y ver su traducción a SQL, por si alguien fuera escéptico respecto a las bondades de Hibernate.

Hibernate es menos invasivo que otros marcos de trabajo de mapeo O/R. Se utilizan la reflexión y la generación de bytecodes en tiempo de ejecución, y la generación del SQL ocurre en el momento del arranque. Esto nos permite desarrollar objetos persistentes siguiendo el lenguaje común de Java: incluyendo asociación, herencia, polimorfismo, composición y las Colecciones de Java.

Hibernate soporta la mayoría de los sistemas de bases de datos SQL. El Hibernate Query Language, diseñado como una extensión mínima, orientada a objetos, de SQL, proporciona un puente elegante entre los mundos objeto y relacional. Hibernate ofrece facilidades para recuperación y actualización de datos, control de transacciones, repositorios de conexiones a bases de datos, consultas programáticas y declarativas y un control de relaciones de entidades declarativas.

6.1.5. Requisitos e incompatibilidades

Hibernate sólo impone una condición para poder usarse, las tablas deben tener una clave primaria, preferentemente una clave no natural, pero debe poder identificar los registros de alguna manera.

Si bien Hibernate reúne un conjunto de características que lo hacen ser muy llamativo, no es una solución óptima en el caso de proyectos de migración de datos, y si los criterios con que se crearon la base de datos no tienen un mínimo de calidad.

6.1.6. Configuración de Hibernate

El primer paso es añadir a nuestro proyecto el Framework Hibernate. Podemos ver en la siguiente imagen como se realiza esto:

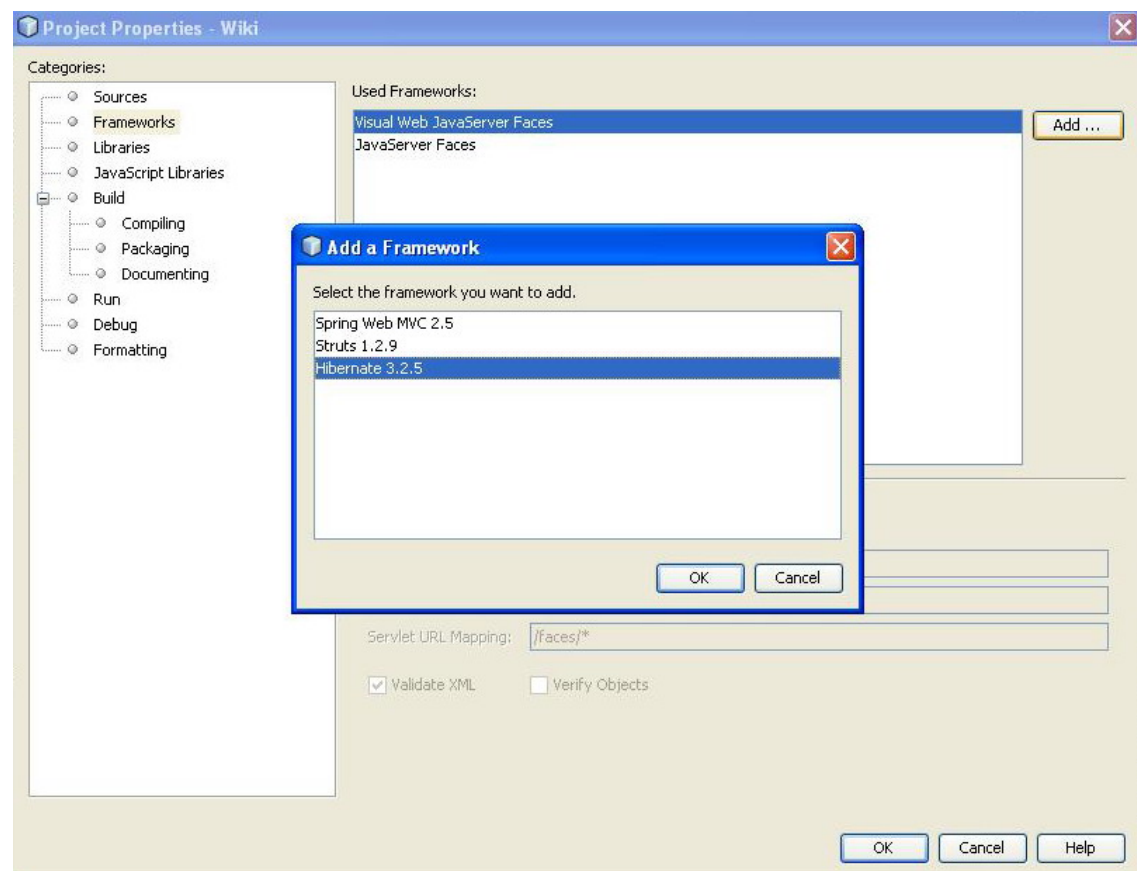


Figura 6.3: Configuración de Hibernate paso 1

Ahora seleccionamos la conexión a nuestra base de datos:

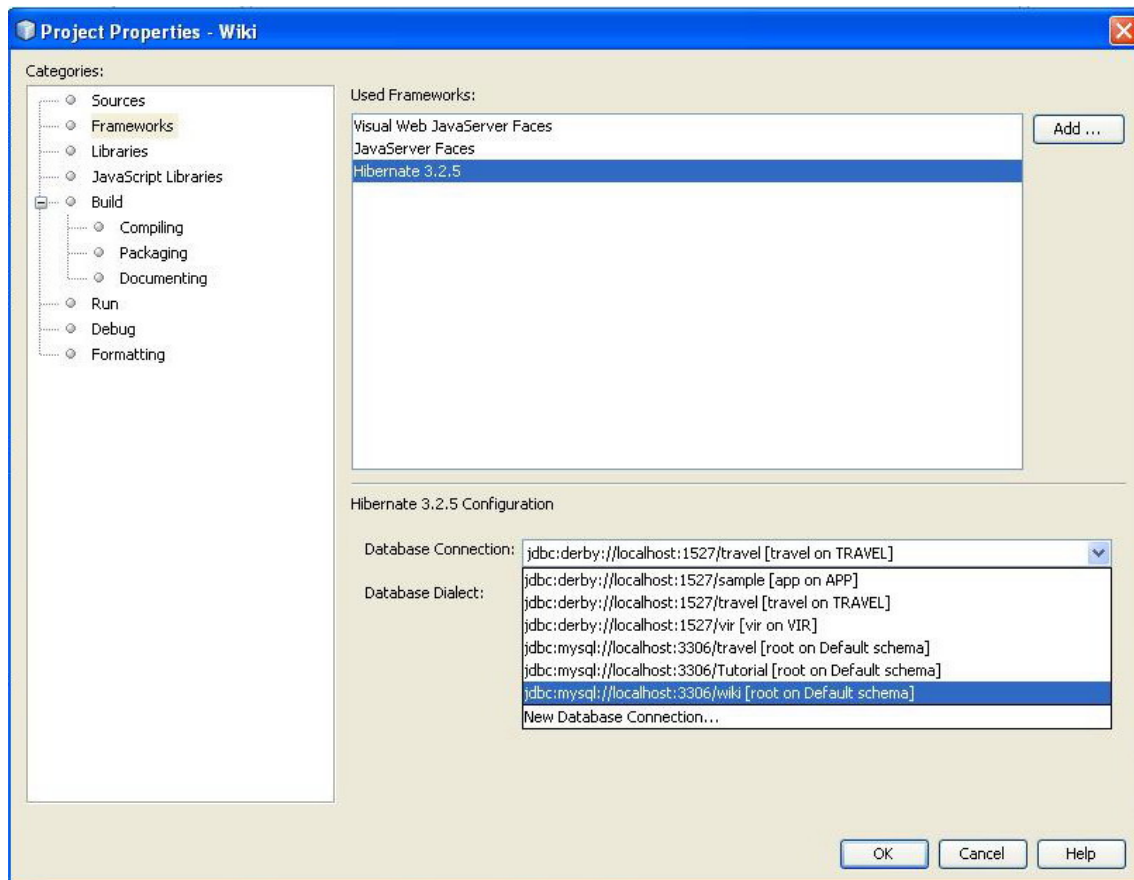


Figura 6.4: Configuración de Hibernate paso 2

Ahora abrimos el fichero hibernate.cfg.xml para ver su contenido:

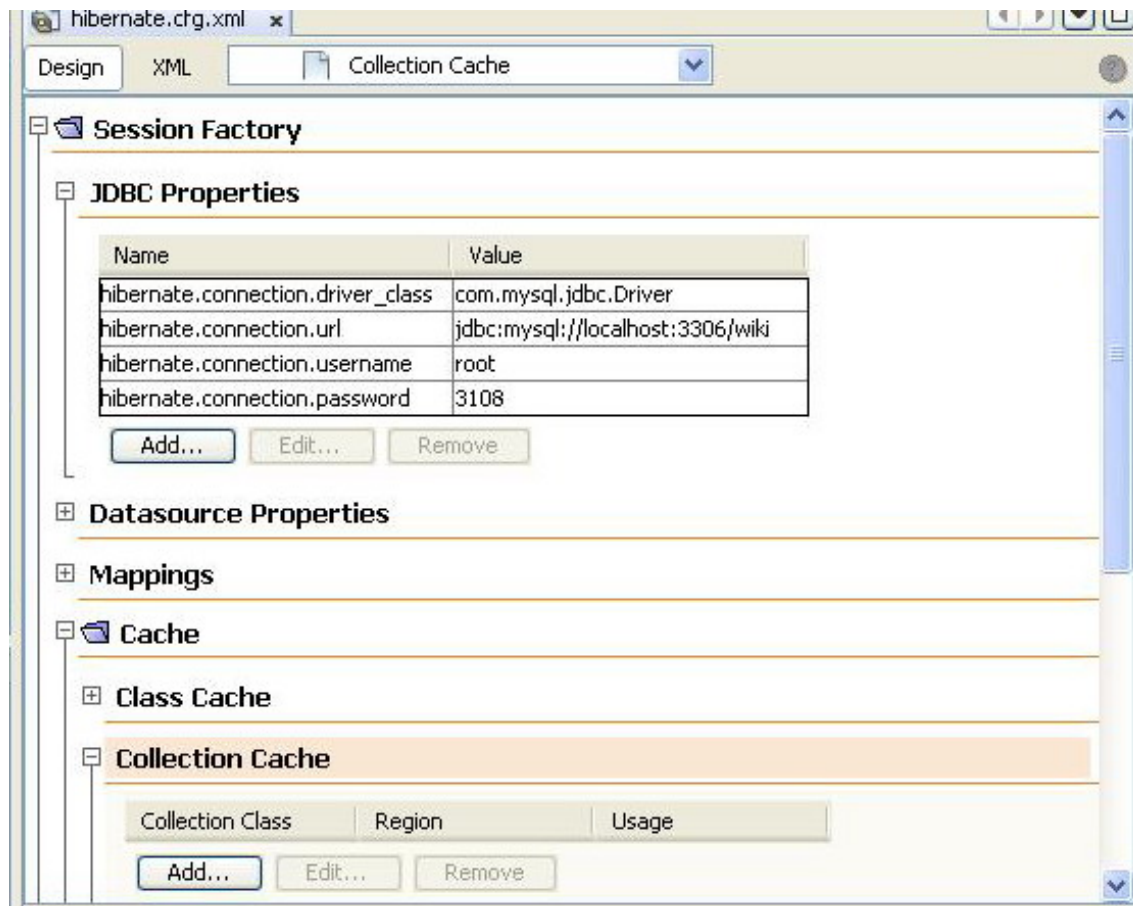


Figura 6.5: Configuración de Hibernate paso 3

El siguiente paso es modificar este fichero de configuración de Hibernate. Este fichero de configuración contiene información sobre la conexión de la base de datos y otras propiedades.

Ahora vamos a configurar el fichero para permitir logearse para la depuración para sentencias SQL y habilitar la gestión del contexto de sesión de Hibernate. Una vez abierto el fichero de configuración, expandimos el nodo de Configuración de Propiedades.

Clickea para añadir propiedad Hibernate:

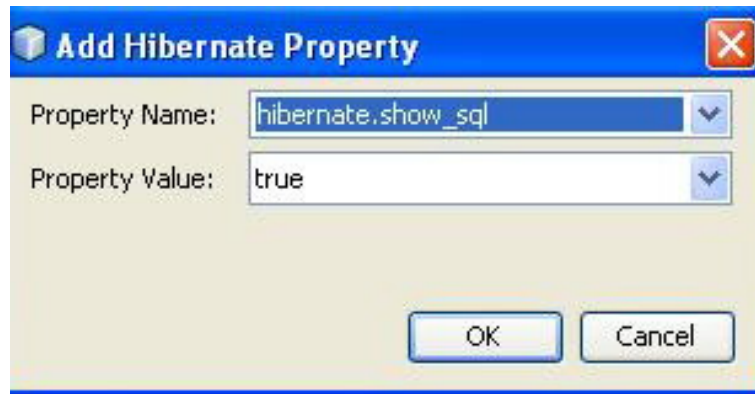


Figura 6.6: Configuración de Hibernate paso 4

Esto permite el logueo de depuración para sentencias SQL. Ahora expande el nodo Miscellaneous Properties y clickea Añadir.



Figura 6.7: Configuración de Hibernate paso 5

6.1.7. Creando una clase Ayuda con Hibernate

Para usar Hibernate necesitas crear una clase de ayuda que maneje el comienzo y que accede a SessionFactory de Hibernate para obtener un objeto Sesión. La clase llama a configure () y carga el fichero de configuración y entonces construye la SessionFactory para obtener el objeto de Sesión.

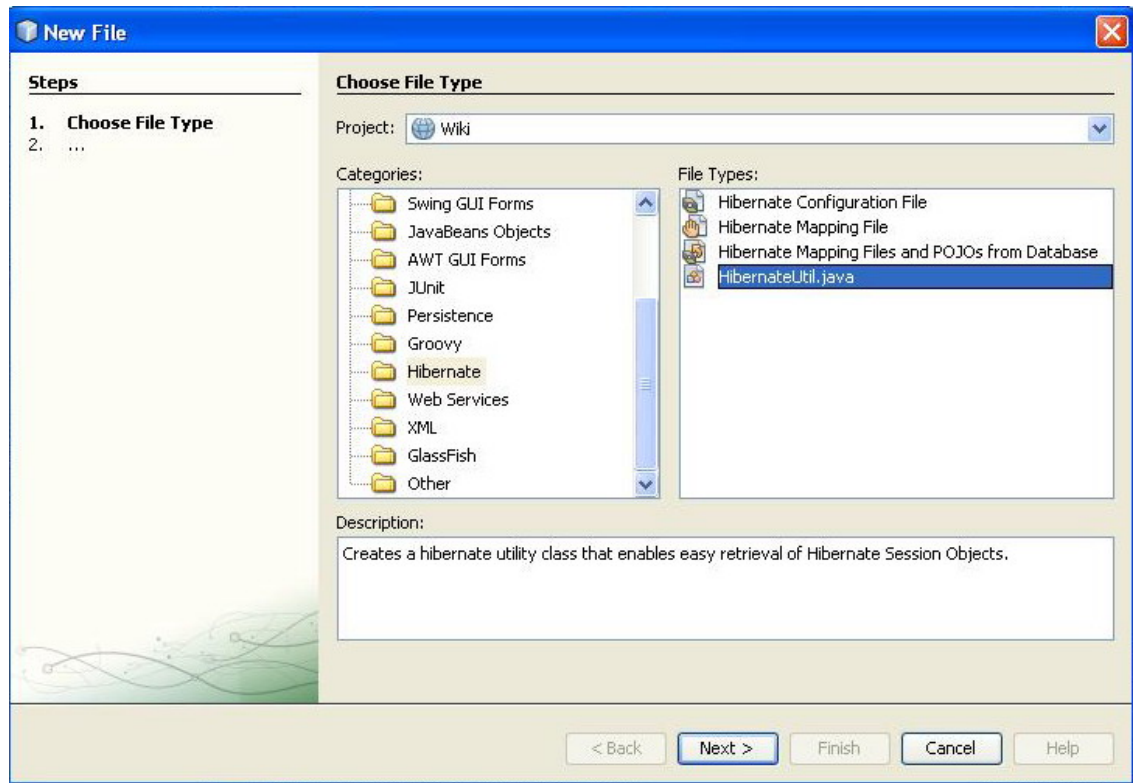


Figura 6.8: Creando clase ayuda con Hibernate

6.1.8. Generando ficheros de mapeo Hibernate y Clases Java

Vamos a seguir los siguientes pasos:

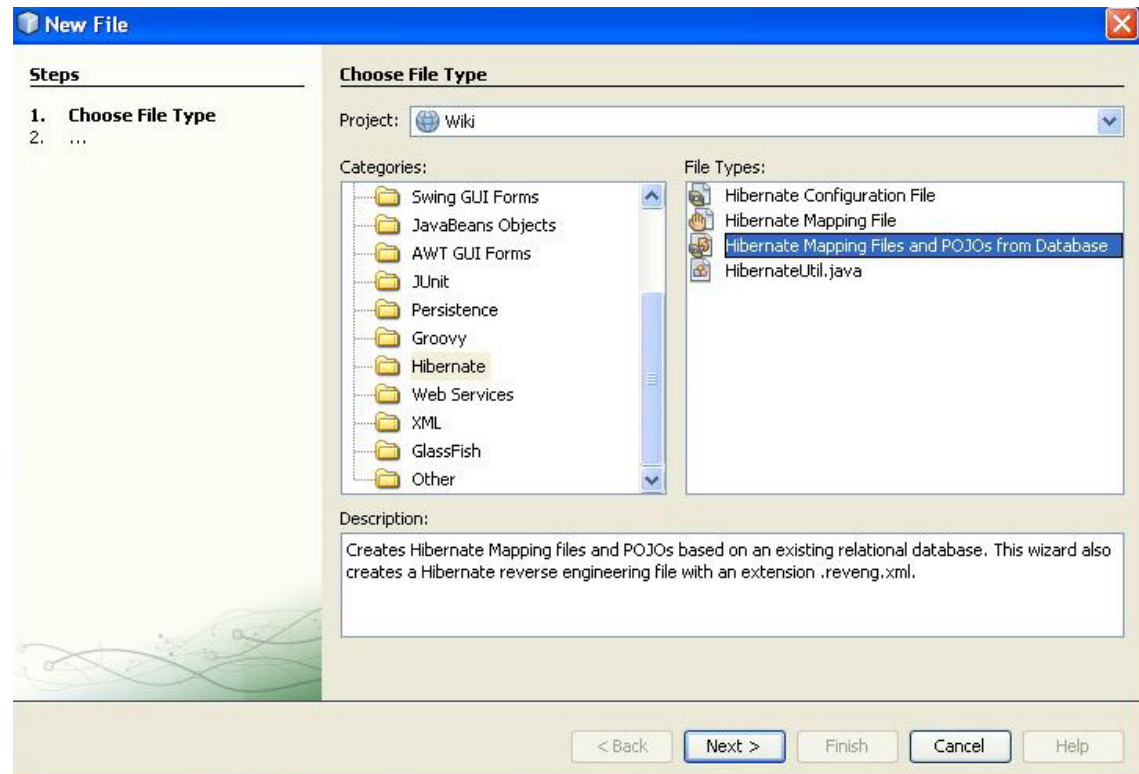


Figura 6.9: Ficheros de mapeo Hibernate y Clases Java, paso 1

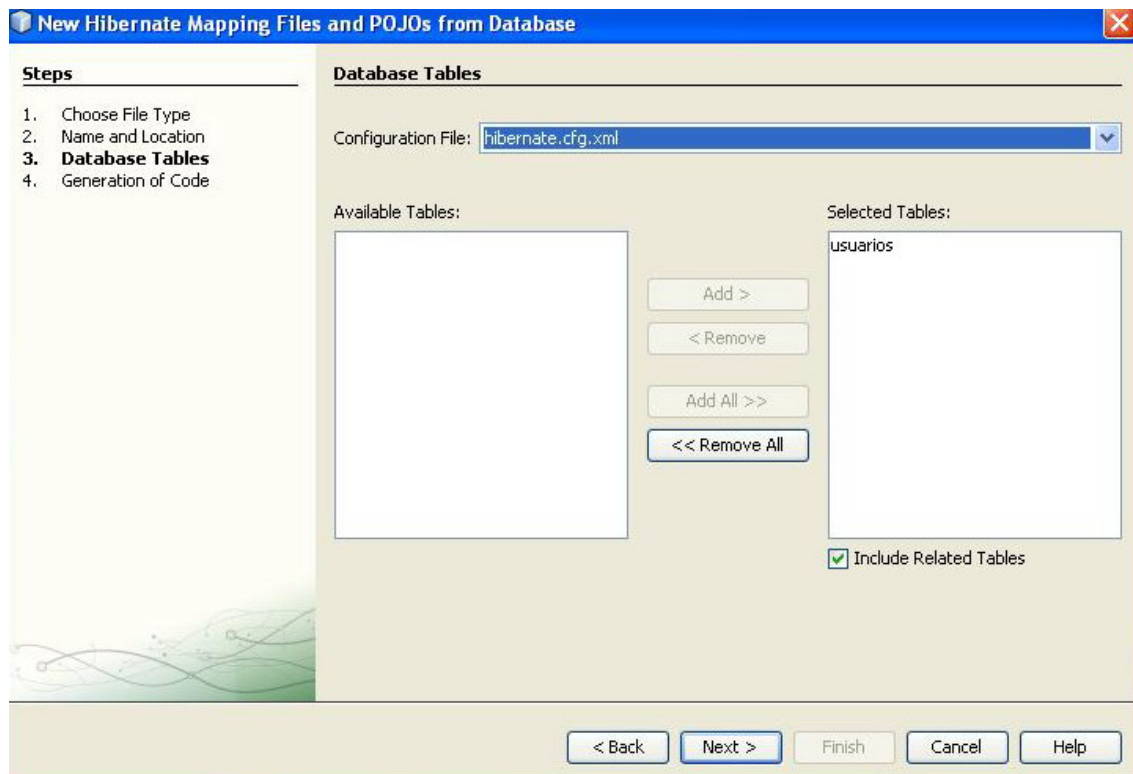


Figura 6.10: Ficheros de mapeo Hibernate y Clases Java, paso 2

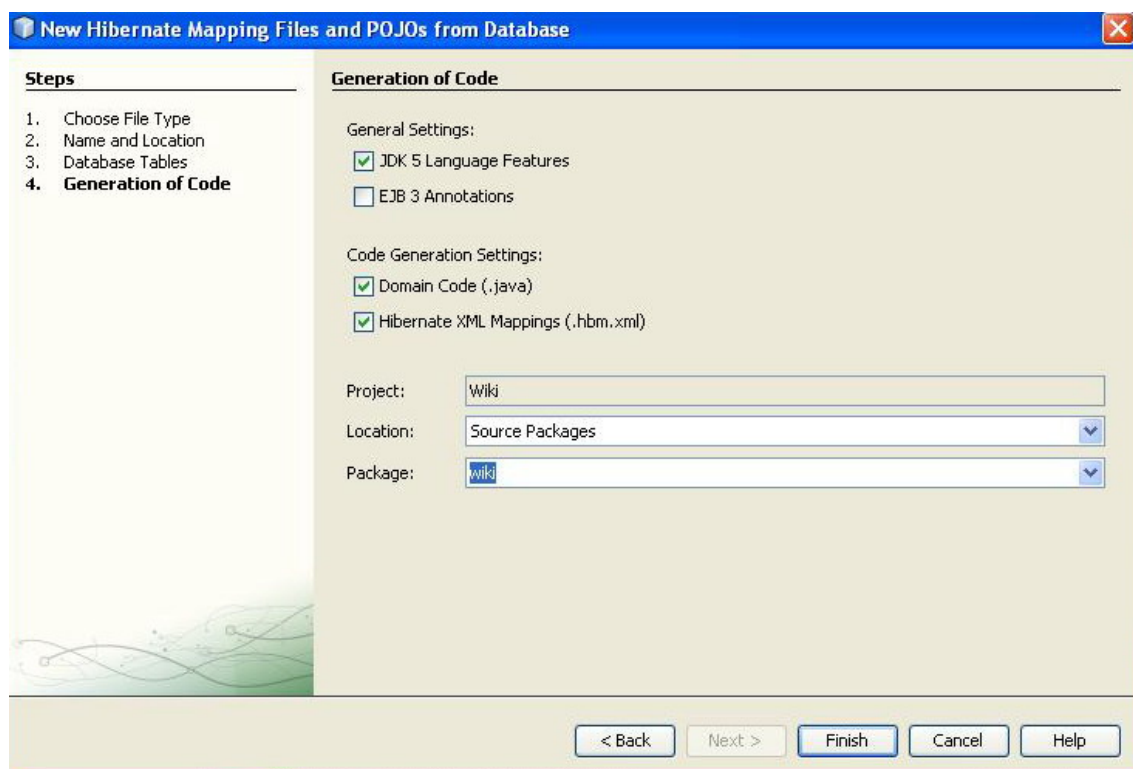


Figura 6.11: Ficheros de mapeo Hibernate y Clases Java, paso 3

Cuando clickeas Finish el IDE genera los ficheros de mapeo POJOs e Hibernate con los campos mapeados a las columnas en las tablas seleccionadas. El IDE también añade entradas de mapeo a hibernate.cfg.xml.

6.1.9. Creando la clase de ayuda UsuarioAyuda.java

Necesitaremos crear una clase de ayuda en el paquete Wiki que se usará para desempeñar las consultas Hibernate en la base de datos. Usaremos HQL editor (Lenguaje de Consulta Hibernate) para construir y probar las consultas de recuperación de datos.

Creando la clase

En esta sección crearemos la clase UsuarioAyuda.java en el paquete Wiki. Se creará una sesión Hibernate para llamar a getSessionFactory en HibernateUtil.java y creamos algunos métodos de ayuda para crear consultas y recuperar datos de la base de datos.

Dentro de nuestro paquete Wiki pulsamos sobre New Java Class. Damos el nombre de la clase y ahora tenemos que crear un atributo dentro de la clase que se corresponda con una sesión Hibernate que inicializaremos en el constructor.

```
1 package wiki;
2
3 import org.hibernate.Session;
4
5 public class UsuarioAyuda {
6     Session session = null;
7
8     public UsuarioAyuda() {
9
10         this.session = HibernateUtil.getSessionFactory().getCurrentSession();
11     }
12 }
```

Luego modificaremos esta clase con los métodos necesarios para realizar las consultas.

Por ejemplo, recuperaremos los usuarios que tenemos en nuestra tabla usuarios de la base de datos. Para ello crearemos una consulta HQL que consulta la base de datos para recuperar una lista de usuarios de la tabla Usuarios. En nuestro caso de ejemplo como solo tenemos un usuario hasta el momento, mostrará únicamente ese usuario.

Usaremos el editor HQL para construir y probar la consulta HQL. Pulsamos con el botón derecho sobre hibernate.cfg.xml en la ventana Projects y elegimos Arrancar consulta HQL para abrir el editor de consultas HQL. Seleccionamos hibernate.cfg de la lista desplegable en la barra de herramientas.

Probamos la conexión escribiendo lo siguiente en el editor y clickeando el botón de consulta Run HQL en la barra de herramientas.

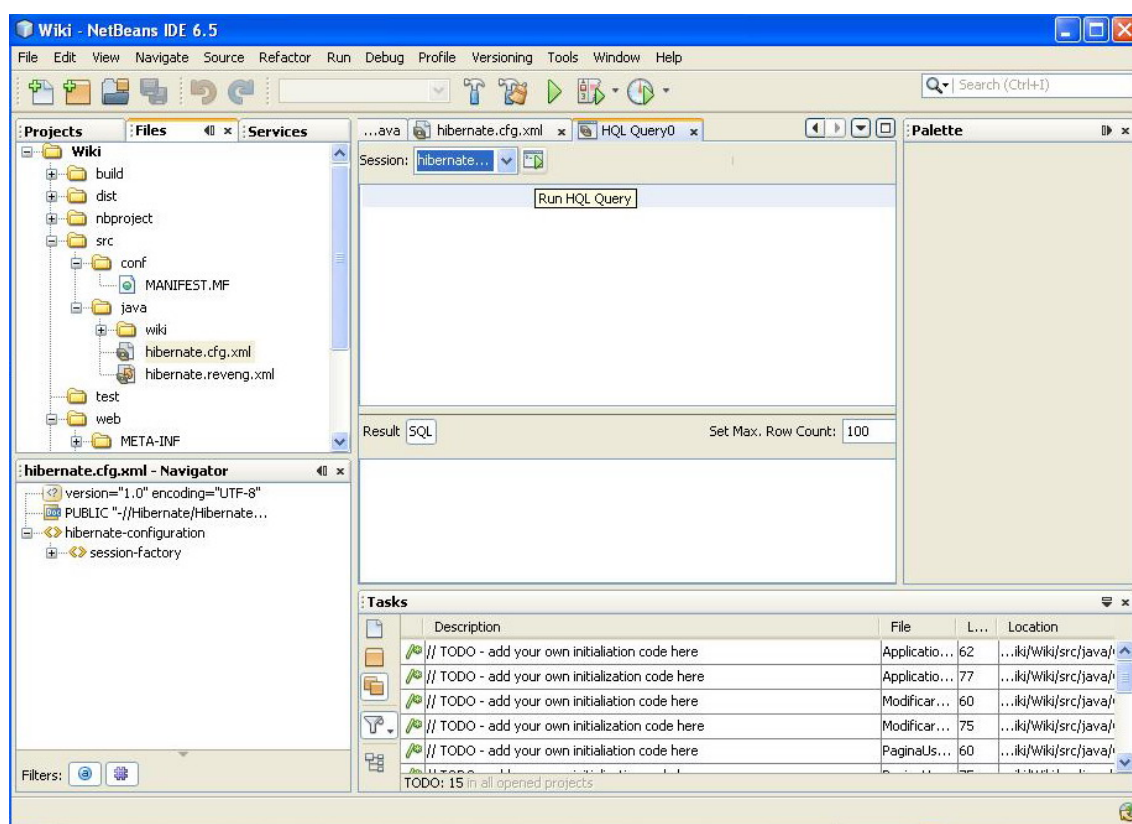


Figura 6.12: Creando la clase UsuarioAyuda.java, paso 1

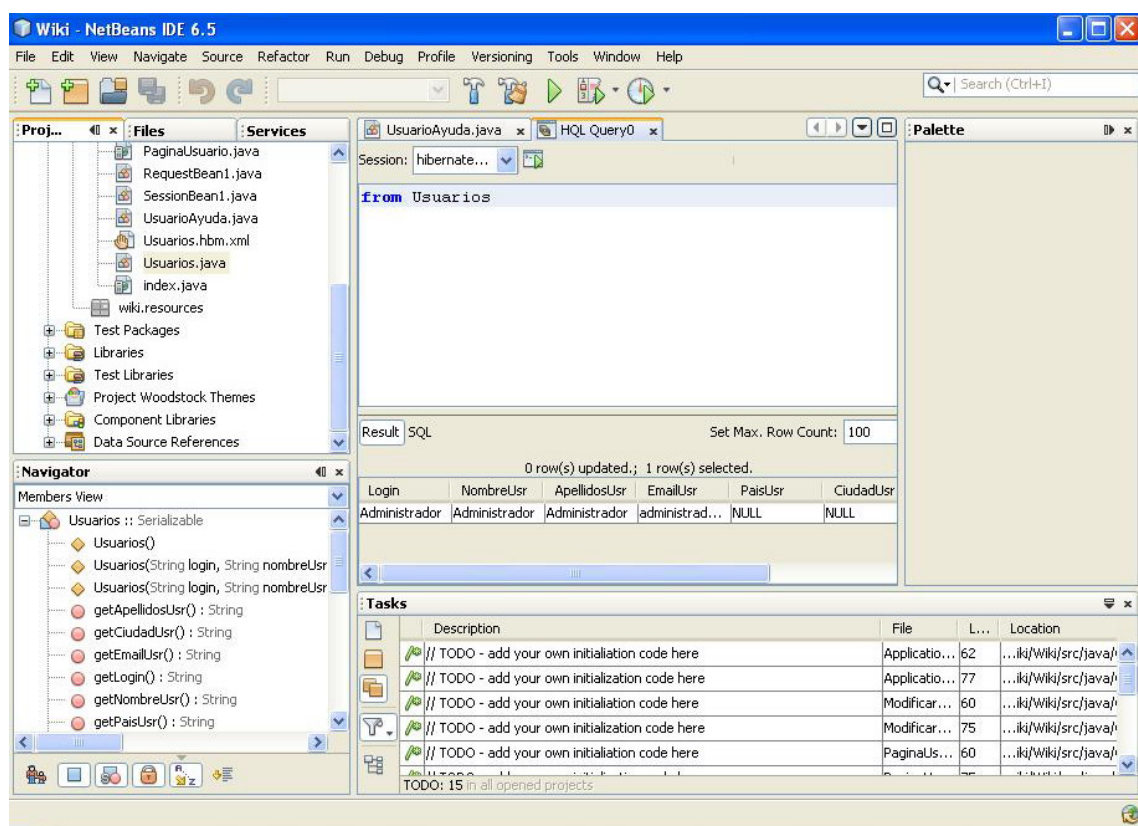


Figura 6.13: Creando la clase UsuarioAyuda.java, paso 2

Ahora vamos a implementar un método para obtener los datos de un usuario que tiene un determinado login.

```
1 public Usuarios getUsuarioLog(String log){
2
3     Usuarios user = null;
4
5     try {
6
7         org.hibernate.Transaction tx = session.beginTransaction();
8
9         List l = session.createQuery("from Usuarios as usuarios where usuarios.login=" + log).list
10            ();
11
12         user = (Usuarios) l.get(0);
13
14     } catch (Exception e) {
15         e.printStackTrace();
16     }
17
18     return user;
19 }
```

7 Implementación

Capítulo

Hibernate

hibernate.reveng.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-reverse-engineering PUBLIC "-//Hibernate/Hibernate Reverse Engineering DTD
3 3.0//EN" "http://hibernate.sourceforge.net/hibernate-reverse-engineering-3.0.dtd">
4 <hibernate-reverse-engineering>
5   <schema-selection match-catalog="wiki"/>
6   <table-filter match-name="administracion_paginas"/>
7   <table-filter match-name="etiquetas"/>
8   <table-filter match-name="etiquetas_pagina"/>
9   <table-filter match-name="paginas"/>
10  <table-filter match-name="usuarios"/>
11 </hibernate-reverse-engineering>
```

hibernate.cfg.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "
3 http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
4 <hibernate-configuration>
5   <session-factory>
6     <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
7     <property name="hibernate.connection.driver_class">org.gjt.mm.mysql.Driver</property>
8     <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/wiki</property>
9     <property name="hibernate.connection.username">root</property>
10    <property name="hibernate.connection.password">comandog</property>
11    <property name="hibernate.show_sql">true</property>
12    <property name="hibernate.current_session_context_class">thread</property>
13    <mapping resource="wiki/Usuarios.hbm.xml"/>
14    <mapping resource="wiki/AdministracionPaginas.hbm.xml"/>
15    <mapping resource="wiki/Paginas.hbm.xml"/>
16    <mapping resource="wiki/Usuarios.hbm.xml"/>
17    <mapping resource="wiki/AdministracionPaginas.hbm.xml"/>
18    <mapping resource="wiki/EtiquetasPagina.hbm.xml"/>
19    <mapping resource="wiki/Paginas.hbm.xml"/>
20    <mapping resource="wiki/Etiquetas.hbm.xml"/>
21  </session-factory>
22 </hibernate-configuration>
```

HibernateUtil.java

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
```

```

6 package wiki;
7
8
9 import javax.transaction.Transaction;
10 import org.hibernate.HibernateException;
11 import org.hibernate.Session;
12 import org.hibernate.cfg.AnnotationConfiguration;
13 import org.hibernate.SessionFactory;
14
15 /**
16  * Hibernate Utility class with a convenient method to get Session Factory object.
17  *
18  * @author Administrador
19  */
20 public class HibernateUtil {
21     private static final SessionFactory sessionFactory;
22
23     public static final ThreadLocal<Session> threadSession = new ThreadLocal<Session>();
24
25     public static final ThreadLocal<Transaction> threadTransaction = new ThreadLocal<Transaction>();
26
27     static {
28         try {
29             // Create the SessionFactory from standard (hibernate.cfg.xml)
30             // config file.
31             sessionFactory = new AnnotationConfiguration().configure().buildSessionFactory();
32         } catch (Throwable ex) {
33             // Log the exception.
34             System.err.println("Initial SessionFactory creation failed." + ex);
35             throw new ExceptionInInitializerError(ex);
36         }
37     }
38
39     public static Session currentSession() throws HibernateException {
40         Session s = threadSession.get();
41         if (s == null || s.isOpen()==false) {
42             s = sessionFactory.openSession();
43             threadSession.set(s);
44         }
45         return s;
46     }
47
48     public static SessionFactory getSessionFactory() {
49         return sessionFactory;
50     }
51 }

```

Usuarios.hbm.xml

```

1 <?xml version="1.0"?>
2 <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
3 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
4 <!-- Generated 16-may-2009 17:06:40 by Hibernate Tools 3.2.1.GA -->
5 <hibernate-mapping>
6     <class name="wiki.Usuarios" table="usuarios" catalog="wiki">
7         <id name="login" type="string">
8             <column name="login" length="20" />
9             <generator class="assigned" />
10        </id>
11        <property name="password" type="string">
12            <column name="password" length="20" not-null="true" />
13        </property>
14        <property name="rolUsr" type="string">
15            <column name="rol_usr" length="2" not-null="true" />

```



```

16     </property>
17     <property name="nombreUsr" type="string">
18         <column name="nombre_usr" length="30" not-null="true" />
19     </property>
20     <property name="apellidosUsr" type="string">
21         <column name="apellidos_usr" length="50" not-null="true" />
22     </property>
23     <property name="emailUsr" type="string">
24         <column name="email_usr" length="30" not-null="true" />
25     </property>
26     <property name="paisUsr" type="string">
27         <column name="pais_usr" length="20" />
28     </property>
29     <property name="ciudadUsr" type="string">
30         <column name="ciudad_usr" length="20" />
31     </property>
32     <property name="telefonoUsr" type="string">
33         <column name="telefono_usr" length="9" />
34     </property>
35 </class>
36 </hibernate-mapping>

```

Paginas.hbm.xml

```

1 <?xml version="1.0"?>
2 <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
3 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
4 <!-- Generated 16-may-2009 17:06:40 by Hibernate Tools 3.2.1.GA -->
5 <hibernate-mapping>
6     <class name="wiki.Paginas" table="paginas" catalog="wiki">
7         <id name="idPagina" type="long">
8             <column name="id_pagina" />
9             <generator class="assigned" />
10        </id>
11        <property name="nombrePagina" type="string">
12            <column name="nombre_pagina" length="20" not-null="true" />
13        </property>
14        <property name="descripcionPagina" type="string">
15            <column name="descripcion_pagina" length="50" not-null="true" />
16        </property>
17        <property name="contenido" type="string">
18            <column name="contenido" />
19        </property>
20        <property name="publicada" type="boolean">
21            <column name="publicada" not-null="true" />
22        </property>
23    </class>
24 </hibernate-mapping>

```

Configuración

context.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Context antiJARLocking="true" path="/Wiki">
3     <Resource auth="Container" driverClassName="com.mysql.jdbc.Driver" maxActive="20" maxIdle="10"
4         maxWait="-1" name="jdbc/wiki_MySQL" password="comandog" type="javax.sql.DataSource" url
5         ="jdbc:mysql://localhost:3306/wiki" username="root" />
6 </Context>

```

faces-config.xml

```

1 <?xml version='1.0' encoding='UTF-8'?>
2
3 <!-- ===== FULL CONFIGURATION FILE ===== -->
4
5 <faces-config version="1.2"
6   xmlns="http://java.sun.com/xml/ns/javaee"
7   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/
   web-facesconfig_1_2.xsd">
9
10
11   <managed-bean>
12     <managed-bean-name>SessionBean1</managed-bean-name>
13     <managed-bean-class>wiki.SessionBean1</managed-bean-class>
14     <managed-bean-scope>session</managed-bean-scope>
15   </managed-bean>
16   <managed-bean>
17     <managed-bean-name>Usuarios</managed-bean-name>
18     <managed-bean-class>wiki.Usuarios</managed-bean-class>
19     <managed-bean-scope>session</managed-bean-scope>
20   </managed-bean>
21   <managed-bean>
22     <managed-bean-name>index</managed-bean-name>
23     <managed-bean-class>wiki.index</managed-bean-class>
24     <managed-bean-scope>request</managed-bean-scope>
25   </managed-bean>
26   <managed-bean>
27     <managed-bean-name>ApplicationBean1</managed-bean-name>
28     <managed-bean-class>wiki.ApplicationBean1</managed-bean-class>
29     <managed-bean-scope>application</managed-bean-scope>
30   </managed-bean>
31   <managed-bean>
32     <managed-bean-name>RequestBean1</managed-bean-name>
33     <managed-bean-class>wiki.RequestBean1</managed-bean-class>
34     <managed-bean-scope>request</managed-bean-scope>
35   </managed-bean>
36   <managed-bean>
37     <managed-bean-name>ModificarPagWiki</managed-bean-name>
38     <managed-bean-class>wiki.ModificarPagWiki</managed-bean-class>
39     <managed-bean-scope>request</managed-bean-scope>
40   </managed-bean>
41   <managed-bean>
42     <managed-bean-name>PaginaUsuario</managed-bean-name>
43     <managed-bean-class>wiki.PaginaUsuario</managed-bean-class>
44     <managed-bean-scope>request</managed-bean-scope>
45   </managed-bean>
46   <managed-bean>
47     <managed-bean-name>RegistroUsuario</managed-bean-name>
48     <managed-bean-class>wiki.RegistroUsuario</managed-bean-class>
49     <managed-bean-scope>request</managed-bean-scope>
50   </managed-bean>
51   <managed-bean>
52     <managed-bean-name>PginaAdmin</managed-bean-name>
53     <managed-bean-class>wiki.PginaAdmin</managed-bean-class>
54     <managed-bean-scope>request</managed-bean-scope>
55   </managed-bean>
56   <navigation-rule>
57     <from-view-id>/index.jsp</from-view-id>
58     <navigation-case>
59       <from-outcome>nextPage</from-outcome>
60       <to-view-id>/PaginaUsuario.jsp</to-view-id>
61     </navigation-case>
62     <navigation-case>
63       <from-outcome>GoToPagVisita</from-outcome>

```

```

64         <to-view-id>/PginaVisitante.jsp</to-view-id>
65     </navigation-case>
66 </navigation-rule>
67 <managed-bean>
68     <managed-bean-name>PaginaAdmin</managed-bean-name>
69     <managed-bean-class>wiki.PaginaAdmin</managed-bean-class>
70     <managed-bean-scope>request</managed-bean-scope>
71 </managed-bean>
72 <navigation-rule>
73     <from-view-id>/index.jsp</from-view-id>
74     <navigation-case>
75         <from-outcome>nextPageAd</from-outcome>
76         <to-view-id>/PaginaAdmin.jsp</to-view-id>
77     </navigation-case>
78 </navigation-rule>
79 <managed-bean>
80     <managed-bean-name>ListadoUsuarios</managed-bean-name>
81     <managed-bean-class>wiki.ListadoUsuarios</managed-bean-class>
82     <managed-bean-scope>request</managed-bean-scope>
83 </managed-bean>
84 <managed-bean>
85     <managed-bean-name>PginaVisitante</managed-bean-name>
86     <managed-bean-class>wiki.PginaVisitante</managed-bean-class>
87     <managed-bean-scope>request</managed-bean-scope>
88 </managed-bean>
89 <navigation-rule>
90     <from-view-id>/ListadoUsuarios.jsp</from-view-id>
91     <navigation-case>
92         <from-outcome>GoToPagAdm</from-outcome>
93         <to-view-id>/PaginaAdmin.jsp</to-view-id>
94     </navigation-case>
95     <navigation-case>
96         <from-outcome>GoToCrearP</from-outcome>
97         <to-view-id>/CrearPgina.jsp</to-view-id>
98     </navigation-case>
99     <navigation-case>
100         <from-outcome>GoToListarP</from-outcome>
101         <to-view-id>/ListarPginas.jsp</to-view-id>
102     </navigation-case>
103     <navigation-case>
104         <from-outcome>GoToAdministrarP</from-outcome>
105         <to-view-id>/AdministrarPginas.jsp</to-view-id>
106     </navigation-case>
107 </navigation-rule>
108 <navigation-rule>
109     <from-view-id>/PginaVisitante.jsp</from-view-id>
110     <navigation-case>
111         <from-outcome>GoToIndex</from-outcome>
112         <to-view-id>/index.jsp</to-view-id>
113     </navigation-case>
114     <navigation-case>
115         <from-outcome>GoToListarP</from-outcome>
116         <to-view-id>/ListarPginas.jsp</to-view-id>
117     </navigation-case>
118 </navigation-rule>
119 <managed-bean>
120     <managed-bean-name>CrearPgina</managed-bean-name>
121     <managed-bean-class>wiki.CrearPgina</managed-bean-class>
122     <managed-bean-scope>request</managed-bean-scope>
123 </managed-bean>
124 <navigation-rule>
125     <from-view-id>/PaginaUsuario.jsp</from-view-id>
126     <navigation-case>
127         <from-outcome>GoToCrearP</from-outcome>
128         <to-view-id>/CrearPgina.jsp</to-view-id>
129     </navigation-case>
130     <navigation-case>

```

```

131         <from-outcome>GoToListarP</from-outcome>
132         <to-view-id>/ListarPginas.jsp</to-view-id>
133     </navigation-case>
134     <navigation-case>
135         <from-outcome>GoToAdministrarP</from-outcome>
136         <to-view-id>/AdministrarPginas.jsp</to-view-id>
137     </navigation-case>
138 </navigation-rule>
139 <navigation-rule>
140     <from-view-id>/PaginaAdmin.jsp</from-view-id>
141     <navigation-case>
142         <from-outcome>GoToCrearP</from-outcome>
143         <to-view-id>/CrearPgina.jsp</to-view-id>
144     </navigation-case>
145     <navigation-case>
146         <from-outcome>GoToPageAd</from-outcome>
147         <to-view-id>/PaginaAdmin.jsp</to-view-id>
148     </navigation-case>
149     <navigation-case>
150         <from-outcome>GoToPageUR</from-outcome>
151         <to-view-id>/PaginaUsuario.jsp</to-view-id>
152     </navigation-case>
153     <navigation-case>
154         <from-outcome>GoToListarP</from-outcome>
155         <to-view-id>/ListarPginas.jsp</to-view-id>
156     </navigation-case>
157     <navigation-case>
158         <from-outcome>GoToAdministrarP</from-outcome>
159         <to-view-id>/AdministrarPginas.jsp</to-view-id>
160     </navigation-case>
161 </navigation-rule>
162 <navigation-rule>
163     <from-view-id>/CrearPgina.jsp</from-view-id>
164     <navigation-case>
165         <from-outcome>GoToIndex</from-outcome>
166         <to-view-id>/index.jsp</to-view-id>
167     </navigation-case>
168     <navigation-case>
169         <from-outcome>GoToListarP</from-outcome>
170         <to-view-id>/ListarPginas.jsp</to-view-id>
171     </navigation-case>
172     <navigation-case>
173         <from-outcome>GoToAdministrarP</from-outcome>
174         <to-view-id>/AdministrarPginas.jsp</to-view-id>
175     </navigation-case>
176     <navigation-case>
177         <from-outcome>GoToPaginaAD</from-outcome>
178         <to-view-id>/PaginaAdmin.jsp</to-view-id>
179     </navigation-case>
180     <navigation-case>
181         <from-outcome>GoToPagina</from-outcome>
182         <to-view-id>/PaginaUsuario.jsp</to-view-id>
183     </navigation-case>
184 </navigation-rule>
185 <managed-bean>
186     <managed-bean-name>ListarPginas</managed-bean-name>
187     <managed-bean-class>wiki.ListarPginas</managed-bean-class>
188     <managed-bean-scope>request</managed-bean-scope>
189 </managed-bean>
190 <navigation-rule>
191     <from-view-id>/ListarPginas.jsp</from-view-id>
192     <navigation-case>
193         <from-outcome>GoToIndex</from-outcome>
194         <to-view-id>/index.jsp</to-view-id>
195     </navigation-case>
196     <navigation-case>
197         <from-outcome>GoToCrearP</from-outcome>

```

```

198         <to-view-id>/CrearPgina.jsp</to-view-id>
199     </navigation-case>
200 <navigation-case>
201     <from-outcome>GoToAdministrarP</from-outcome>
202     <to-view-id>/AdministrarPginas.jsp</to-view-id>
203 </navigation-case>
204 </navigation-rule>
205 <managed-bean>
206     <managed-bean-name>AdministrarPginas</managed-bean-name>
207     <managed-bean-class>wiki.AdministrarPginas</managed-bean-class>
208     <managed-bean-scope>request</managed-bean-scope>
209 </managed-bean>
210 <navigation-rule>
211     <from-view-id>/AdministrarPginas.jsp</from-view-id>
212 <navigation-case>
213     <from-outcome>GoToIndex</from-outcome>
214     <to-view-id>/index.jsp</to-view-id>
215 </navigation-case>
216 <navigation-case>
217     <from-outcome>DarPerm</from-outcome>
218     <to-view-id>/DarPermisos.jsp</to-view-id>
219 </navigation-case>
220 <navigation-case>
221     <from-outcome>GoToCrearP</from-outcome>
222     <to-view-id>/CrearPgina.jsp</to-view-id>
223 </navigation-case>
224 <navigation-case>
225     <from-outcome>GoToListarP</from-outcome>
226     <to-view-id>/ListarPginas.jsp</to-view-id>
227 </navigation-case>
228 <navigation-case>
229     <from-outcome>GoToModificarPW</from-outcome>
230     <to-view-id>/ModificarPagWiki.jsp</to-view-id>
231 </navigation-case>
232 </navigation-rule>
233 <application>
234     <view-handler>org.icefaces.netbeans.rave.web.ui.appbase.faces.ViewHandlerImpl</view-handler>
235 </application>
236 <navigation-rule>
237     <from-view-id>/ModificarPagWiki.jsp</from-view-id>
238 <navigation-case>
239     <from-outcome>GoToListarP</from-outcome>
240     <to-view-id>/ListarPginas.jsp</to-view-id>
241 </navigation-case>
242 <navigation-case>
243     <from-outcome>GoToPageAd</from-outcome>
244     <to-view-id>/PaginaAdmin.jsp</to-view-id>
245 </navigation-case>
246 <navigation-case>
247     <from-outcome>GoToPageUR</from-outcome>
248     <to-view-id>/PaginaUsuario.jsp</to-view-id>
249 </navigation-case>
250 </navigation-rule>
251 <navigation-rule>
252     <from-view-id>/ListarPginas.jsp</from-view-id>
253 <navigation-case>
254     <from-outcome>VisualizarPag</from-outcome>
255     <to-view-id>/PginaWiki.jsp</to-view-id>
256 </navigation-case>
257 <navigation-case>
258     <from-outcome>GoToPageAd</from-outcome>
259     <to-view-id>/PaginaAdmin.jsp</to-view-id>
260 </navigation-case>
261 <navigation-case>
262     <from-outcome>GoToPageUR</from-outcome>
263     <to-view-id>/PaginaUsuario.jsp</to-view-id>
264 </navigation-case>

```

```

265 </navigation-rule>
266
267 <navigation-rule>
268   <from-view-id>/DarPermisos.jsp</from-view-id>
269   <navigation-case>
270     <from-outcome>GoToIndex</from-outcome>
271     <to-view-id>/index.jsp</to-view-id>
272   </navigation-case>
273   <navigation-case>
274     <from-outcome>GoToListarP</from-outcome>
275     <to-view-id>/ListarPginas.jsp</to-view-id>
276   </navigation-case>
277   <navigation-case>
278     <from-outcome>GoToPageAd</from-outcome>
279     <to-view-id>/PaginaAdmin.jsp</to-view-id>
280   </navigation-case>
281   <navigation-case>
282     <from-outcome>GoToPageUR</from-outcome>
283     <to-view-id>/PaginaUsuario.jsp</to-view-id>
284   </navigation-case>
285 </navigation-rule>
286 <navigation-rule>
287   <from-view-id>/PginaWiki.jsp</from-view-id>
288   <navigation-case>
289     <from-outcome>GoToPageUR</from-outcome>
290     <to-view-id>/PaginaUsuario.jsp</to-view-id>
291   </navigation-case>
292   <navigation-case>
293     <from-outcome>GoToPageAd</from-outcome>
294     <to-view-id>/PaginaAdmin.jsp</to-view-id>
295   </navigation-case>
296   <navigation-case>
297     <from-outcome>GoToListarP</from-outcome>
298     <to-view-id>/ListarPginas.jsp</to-view-id>
299   </navigation-case>
300 </navigation-rule>
301 <navigation-rule>
302   <from-view-id>/AdministrarPginas.jsp</from-view-id>
303   <navigation-case>
304     <from-outcome>GoToPageUR</from-outcome>
305     <to-view-id>/PaginaUsuario.jsp</to-view-id>
306   </navigation-case>
307   <navigation-case>
308     <from-outcome>GoToPageAd</from-outcome>
309     <to-view-id>/PaginaAdmin.jsp</to-view-id>
310   </navigation-case>
311   <navigation-case>
312     <from-outcome>GoToListarP</from-outcome>
313     <to-view-id>/ListarPginas.jsp</to-view-id>
314   </navigation-case>
315 </navigation-rule>
316 <managed-bean>
317   <managed-bean-name>DarPermisos</managed-bean-name>
318   <managed-bean-class>wiki.DarPermisos</managed-bean-class>
319   <managed-bean-scope>request</managed-bean-scope>
320 </managed-bean>
321 <managed-bean>
322   <managed-bean-name>PginaWiki</managed-bean-name>
323   <managed-bean-class>wiki.PginaWiki</managed-bean-class>
324   <managed-bean-scope>request</managed-bean-scope>
325 </managed-bean>
326 <managed-bean>
327   <managed-bean-name>ModificarPaginaWiki</managed-bean-name>
328   <managed-bean-class>wiki.ModificarPaginaWiki</managed-bean-class>
329   <managed-bean-scope>request</managed-bean-scope>
330 </managed-bean>
331 <managed-bean>

```

```

332     <managed-bean-name>CrearPag</managed-bean-name>
333     <managed-bean-class>wiki.CrearPag</managed-bean-class>
334     <managed-bean-scope>request</managed-bean-scope>
335   </managed-bean>
336 </faces-config>

```

project.properties

```

1  auxiliary.org-netbeans-modules-editor-indent.CodeStyle.project.tab-size=8
2  auxiliary.org-netbeans-modules-editor-indent.CodeStyle.project.text-limit-width=80
3  auxiliary.org-netbeans-modules-editor-indent.CodeStyle.usedProfile=default
4  build.classes.dir=${build.web.dir}/WEB-INF/classes
5  build.classes.excludes=**/*.java,**/*.form
6  build.dir=build
7  build.generated.dir=${build.dir}/generated
8  build.test.classes.dir=${build.dir}/test/classes
9  build.test.results.dir=${build.dir}/test/results
10 build.web.dir=${build.dir}/web
11 build.web.excludes=${build.classes.excludes}
12 client.urlPart=
13 compile.jsps=false
14 conf.dir=${source.root}/conf
15 debug.classpath=${build.classes.dir}:${javac.classpath}
16 debug.test.classpath=\
17     ${run.test.classpath}
18 display.browser=true
19 dist.dir=dist
20 dist.ear.war=${dist.dir}/${war.ear.name}
21 dist.javadoc.dir=${dist.dir}/javadoc
22 dist.war=${dist.dir}/${war.name}
23 excludes=
24 file.reference.mysql-connector-java-5.0.0-bin.jar=mysql-connector-java-5.0.0-bin.jar
25 includes=**
26 j2ee.deploy.on.save=true
27 j2ee.platform=1.5
28 j2ee.server.type=Tomcat60
29 jar.compress=false
30 javac.classpath=\
31     ${libs.jsf12.classpath}:\
32     ${libs.jstl11.classpath}:\
33     ${libs.woodstock-theme-default.classpath}:\
34     ${libs.hibernate-support.classpath}:\
35     ${libs.ejb3-persistence.classpath}:\
36     ${file.reference.mysql-connector-java-5.0.0-bin.jar}:\
37     ${libs.icefaces-jsf12-design-time-180.classpath}:\
38     ${libs.jsf-1_2_12.classpath}:\
39     ${libs.jsf12-support-runtime-modified.classpath}:\
40     ${libs.icefaces-jsf12-bm.classpath}:\
41     ${libs.icefaces-jsf12-runtime-support.classpath}:\
42     ${libs.woodstock-components-modified.classpath}
43 # Space-separated list of extra javac options
44 javac.compilerargs=
45 javac.debug=true
46 javac.deprecation=false
47 javac.source=1.5
48 javac.target=1.5
49 javac.test.classpath=\
50     ${javac.classpath}:\
51     ${build.classes.dir}:\
52     ${libs.junit.classpath}:\
53     ${libs.junit_4.classpath}
54 javadoc.additionalparam=
55 javadoc.author=false
56 javadoc.encoding=${source.encoding}

```

```

57 javadoc.noindex=false
58 javadoc.nonavbar=false
59 javadoc.notree=false
60 javadoc.preview=true
61 javadoc.private=false
62 javadoc.splitindex=true
63 javadoc.use=true
64 javadoc.version=false
65 javadoc.windowtitle=
66 jspcompilation.classpath=${jspc.classpath}:${javac.classpath}
67 lib.dir=${web.docbase.dir}/WEB-INF/lib
68 persistence.xml.dir=${conf.dir}
69 platform.active=default_platform
70 resource.dir=setup
71 run.test.classpath=\
72     ${javac.test.classpath}:\
73     ${build.test.classes.dir}
74 # Space-separated list of JVM arguments used when running a class with a main method or a unit test
75 # (you may also define separate properties like run-sys-prop.name=value instead of -Dname=value):
76 runmain.jvmargs=
77 source.encoding=UTF-8
78 source.root=src
79 src.dir=${source.root}/java
80 test.src.dir=test
81 war.content.additional=\
82     ${libs.exceptionhandler-runtime.classpath}:\
83     ${libs.jstl.classpath}:\
84     ${libs.ws-icefaces-runtime-180.classpath}
85 war.ear.name=Wiki.war
86 war.name=Wiki.war
87 web.docbase.dir=web
88 webinf.dir=web/WEB-INF

```

ApplicationBean1.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8  import com.sun.rave.web.ui.appbase.AbstractApplicationBean;
9  import javax.faces.FacesException;
10
11  /**
12   * <p>Application scope data bean for your application. Create properties
13   * here to represent cached data that should be made available to all users
14   * and pages in the application.</p>
15   *
16   * <p>An instance of this class will be created for you automatically,
17   * the first time your application evaluates a value binding expression
18   * or method binding expression that references a managed bean using
19   * this class.</p>
20   *
21   * @version ApplicationBean1.java
22   * @version Created on 04-may-2009, 19:13:40
23   * @author WIND MOTEBOOK
24   */
25
26  public class ApplicationBean1 extends AbstractApplicationBean {
27      // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
28
29      /**

```



```

30  * <p>Automatically managed component initialization. <strong>WARNING:</strong>
31  * This method is automatically generated, so any user-specified code inserted
32  * here is subject to being replaced.</p>
33  */
34  private void _init() throws Exception {
35  }
36  // </editor-fold>
37
38  /**
39  * <p>Construct a new application data bean instance.</p>
40  */
41  public ApplicationBean1() {
42  }
43
44  /**
45  * <p>This method is called when this bean is initially added to
46  * application scope. Typically, this occurs as a result of evaluating
47  * a value binding or method binding expression, which utilizes the
48  * managed bean facility to instantiate this bean and store it into
49  * application scope.</p>
50  *
51  * <p>You may customize this method to initialize and cache application wide
52  * data values (such as the lists of valid options for dropdown list
53  * components), or to allocate resources that are required for the
54  * lifetime of the application.</p>
55  */
56  @Override
57  public void init() {
58      // Perform initializations inherited from our superclass
59      super.init();
60      // Perform application initialization that must complete
61      // *before* managed components are initialized
62      // TODO – add your own initialization code here
63
64      // <editor-fold defaultstate="collapsed" desc="Managed Component Initialization">
65      // Initialize automatically managed components
66      // *Note* – this logic should NOT be modified
67      try {
68          _init();
69      } catch (Exception e) {
70          log("ApplicationBean1 Initialization Failure", e);
71          throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
72      }
73
74      // </editor-fold>
75      // Perform application initialization that must complete
76      // *after* managed components are initialized
77      // TODO – add your own initialization code here
78  }
79
80  /**
81  * <p>This method is called when this bean is removed from
82  * application scope. Typically, this occurs as a result of
83  * the application being shut down by its owning container.</p>
84  *
85  * <p>You may customize this method to clean up resources allocated
86  * during the execution of the <code>init()</code> method, or
87  * at any later time during the lifetime of the application.</p>
88  */
89  @Override
90  public void destroy() {
91  }
92
93  /**
94  * <p>Return an appropriate character encoding based on the
95  * <code>Locale</code> defined for the current JavaServer Faces
96  * view. If no more suitable encoding can be found, return

```

```

97      * "UTF-8" as a general purpose default.</p>
98      *
99      * <p>The default implementation uses the implementation from
100      * our superclass, <code>AbstractApplicationBean</code>.</p>
101      */
102      @Override
103      public String getLocaleCharacterEncoding() {
104          return super.getLocaleCharacterEncoding();
105      }
106  }

```

SessionBean1.java

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package wiki;
7
8  import com.sun.rave.web.ui.appbase.AbstractSessionBean;
9  import com.sun.sql.rowset.CachedRowSetXImpl;
10 import java.util.List;
11 import javax.faces.FacesException;
12
13 /**
14 * <p>Session scope data bean for your application. Create properties
15 * here to represent cached data that should be made available across
16 * multiple HTTP requests for an individual user.</p>
17 *
18 * <p>An instance of this class will be created for you automatically,
19 * the first time your application evaluates a value binding expression
20 * or method binding expression that references a managed bean using
21 * this class.</p>
22 *
23 * @version SessionBean1.java
24 * @version Created on 04-may-2009, 19:13:40
25 * @author WIND MOTEBOOK
26 */
27
28 public class SessionBean1 extends AbstractSessionBean {
29     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
30     /**
31      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
32      * This method is automatically generated, so any user-specified code inserted
33      * here is subject to being replaced.</p>
34      */
35     private void _init() throws Exception {
36         usuariosRowSet.setDataSourceName("java:comp/env/jdbc/wiki_MySQL");
37         usuariosRowSet.setCommand("SELECT * FROM usuarios");
38         usuariosRowSet.setTableName("usuarios");
39     }
40     // </editor-fold>
41
42     private String password;
43     private String rolUsr;
44     private String nombreUsr;
45     private String apellidosUsr;
46     private String emailUsr;
47     private String paisUsr;
48     private String ciudadUsr;
49     private String telefonoUsr;
50     private String login;
51     private List<Usuarios> listaU;

```

```

52     private List<Paginas> listaP;
53     private List<Usuarios> listaUPermisos;
54     private Paginas pagina;
55     private CachedRowSetXImpl usuariosRowSet = new CachedRowSetXImpl();
56
57     public CachedRowSetXImpl getUsuariosRowSet() {
58         return usuariosRowSet;
59     }
60
61     public void setUsuariosRowSet(CachedRowSetXImpl crsxi) {
62         this.usuariosRowSet = crsxi;
63     }
64
65     /**
66      * <p>Construct a new session data bean instance.</p>
67      */
68     public SessionBean1() {
69     }
70
71     /**
72      * <p>This method is called when this bean is initially added to
73      * session scope. Typically, this occurs as a result of evaluating
74      * a value binding or method binding expression, which utilizes the
75      * managed bean facility to instantiate this bean and store it into
76      * session scope.</p>
77      *
78      * <p>You may customize this method to initialize and cache data values
79      * or resources that are required for the lifetime of a particular
80      * user session.</p>
81      */
82     @Override
83     public void init() {
84         // Perform initializations inherited from our superclass
85         super.init();
86         // Perform application initialization that must complete
87         // *before* managed components are initialized
88         // TODO – add your own initialization code here
89
90         // <editor–fold defaultstate="collapsed" desc="Managed Component Initialization">
91         // Initialize automatically managed components
92         // *Note* – this logic should NOT be modified
93         try {
94             _init();
95         } catch (Exception e) {
96             log("SessionBean1 Initialization Failure", e);
97             throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
98         }
99
100         // </editor–fold>
101         // Perform application initialization that must complete
102         // *after* managed components are initialized
103         // TODO – add your own initialization code here
104     }
105
106     /**
107      * <p>This method is called when the session containing it is about to be
108      * passivated. Typically, this occurs in a distributed servlet container
109      * when the session is about to be transferred to a different
110      * container instance, after which the <code>activate()</code> method
111      * will be called to indicate that the transfer is complete.</p>
112      *
113      * <p>You may customize this method to release references to session data
114      * or resources that can not be serialized with the session itself.</p>
115      */
116     @Override
117     public void passivate() {
118     }

```

```
119
120  /**
121   * <p>This method is called when the session containing it was
122   * reactivated.</p>
123   *
124   * <p>You may customize this method to reacquire references to session
125   * data or resources that could not be serialized with the
126   * session itself.</p>
127   */
128  @Override
129  public void activate() {
130  }
131
132  /**
133   * <p>This method is called when this bean is removed from
134   * session scope. Typically, this occurs as a result of
135   * the session timing out or being terminated by the application.</p>
136   *
137   * <p>You may customize this method to clean up resources allocated
138   * during the execution of the <code>init()</code> method, or
139   * at any later time during the lifetime of the application.</p>
140   */
141  @Override
142  public void destroy() {
143  }
144
145  /**
146   * <p>Return a reference to the scoped data bean.</p>
147   *
148   * @return reference to the scoped data bean
149   */
150  protected ApplicationBean1 getApplicationBean1() {
151      return (ApplicationBean1) getBean("ApplicationBean1");
152  }
153
154
155
156
157  /**
158   * @return the password
159   */
160  public String getPassword() {
161      return password;
162  }
163
164  /**
165   * @return the rolUsr
166   */
167  public String getRolUsr() {
168      return rolUsr;
169  }
170
171  /**
172   * @return the nombreUsr
173   */
174  public String getNombreUsr() {
175      return nombreUsr;
176  }
177
178  /**
179   * @return the apellidosUsr
180   */
181  public String getApellidosUsr() {
182      return apellidosUsr;
183  }
184
185  /**
```

```
186      * @return the emailUsr
187      */
188      public String getEmailUsr() {
189          return emailUsr;
190      }
191
192      /**
193       * @return the paisUsr
194       */
195      public String getPaisUsr() {
196          return paisUsr;
197      }
198
199      /**
200       * @return the ciudadUsr
201       */
202      public String getCiudadUsr() {
203          return ciudadUsr;
204      }
205
206      /**
207       * @return the telefonoUsr
208       */
209      public String getTelefonoUsr() {
210          return telefonoUsr;
211      }
212
213      /**
214       * @param password the password to set
215       */
216      public void setPassword(String password) {
217          this.password = password;
218      }
219
220      /**
221       * @param rolUsr the rolUsr to set
222       */
223      public void setRolUsr(String rolUsr) {
224          this.rolUsr = rolUsr;
225      }
226
227      /**
228       * @param nombreUsr the nombreUsr to set
229       */
230      public void setNombreUsr(String nombreUsr) {
231          this.nombreUsr = nombreUsr;
232      }
233
234      /**
235       * @param apellidosUsr the apellidosUsr to set
236       */
237      public void setApellidosUsr(String apellidosUsr) {
238          this.apellidosUsr = apellidosUsr;
239      }
240
241      /**
242       * @param emailUsr the emailUsr to set
243       */
244      public void setEmailUsr(String emailUsr) {
245          this.emailUsr = emailUsr;
246      }
247
248      /**
249       * @param paisUsr the paisUsr to set
250       */
251      public void setPaisUsr(String paisUsr) {
252          this.paisUsr = paisUsr;
```

```
253     }
254
255     /**
256      * @param ciudadUsr the ciudadUsr to set
257      */
258     public void setCiudadUsr(String ciudadUsr) {
259         this.ciudadUsr = ciudadUsr;
260     }
261
262     /**
263      * @param telefonoUsr the telefonoUsr to set
264      */
265     public void setTelefonoUsr(String telefonoUsr) {
266         this.telefonoUsr = telefonoUsr;
267     }
268
269     /**
270      * @return the login
271      */
272     public String getLogin() {
273         return login;
274     }
275
276     /**
277      * @param login the login to set
278      */
279     public void setLogin(String login) {
280         this.login = login;
281     }
282
283     /**
284      * @return the listaU
285      */
286     public List<Usuarios> getListaU() {
287         return listaU;
288     }
289
290     /**
291      * @param listaU the listaU to set
292      */
293     public void setListaU(List<Usuarios> listaU) {
294         this.listaU = listaU;
295     }
296
297     /**
298      * @return the listaP
299      */
300     public List<Paginas> getListaP() {
301         return listaP;
302     }
303
304     /**
305      * @param listaP the listaP to set
306      */
307     public void setListaP(List<Paginas> listaP) {
308         this.listaP = listaP;
309     }
310
311     /**
312      * @return the pagina
313      */
314     public Paginas getPagina() {
315         return pagina;
316     }
317
318     /**
319      * @param pagina the pagina to set
```

```

320     */
321     public void setPagina(Paginas pagina) {
322         this.pagina = pagina;
323     }
324
325     /**
326      * @return the listaUPermisos
327      */
328     public List<Usuarios> getListaUPermisos() {
329         return listaUPermisos;
330     }
331
332     /**
333      * @param listaUPermisos the listaUPermisos to set
334      */
335     public void setListaUPermisos(List<Usuarios> listaUPermisos) {
336         this.listaUPermisos = listaUPermisos;
337     }
338
339 }

```

RequestBean1.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8  import com.sun.rave.web.ui.appbase.AbstractRequestBean;
9  import javax.faces.FacesException;
10
11  /**
12   * <p>Request scope data bean for your application. Create properties
13   * here to represent data that should be made available across different
14   * pages in the same HTTP request, so that the page bean classes do not
15   * have to be directly linked to each other.</p>
16   *
17   * <p>An instance of this class will be created for you automatically,
18   * the first time your application evaluates a value binding expression
19   * or method binding expression that references a managed bean using
20   * this class.</p>
21   *
22   * @version RequestBean1.java
23   * @version Created on 04-may-2009, 19:13:41
24   * @author WIND MOTEBOOK
25   */
26
27  public class RequestBean1 extends AbstractRequestBean {
28      // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
29
30      /**
31       * <p>Automatically managed component initialization. <strong>WARNING:</strong>
32       * This method is automatically generated, so any user-specified code inserted
33       * here is subject to being replaced.</p>
34       */
35      private void _init() throws Exception {
36      }
37      // </editor-fold>
38
39      /**
40       * <p>Construct a new request data bean instance.</p>
41       */

```

```

42 public RequestBean1() {
43 }
44
45 /**
46  * <p>This method is called when this bean is initially added to
47  * request scope. Typically, this occurs as a result of evaluating
48  * a value binding or method binding expression, which utilizes the
49  * managed bean facility to instantiate this bean and store it into
50  * request scope.</p>
51  *
52  * <p>You may customize this method to allocate resources that are required
53  * for the lifetime of the current request.</p>
54  */
55 @Override
56 public void init() {
57     // Perform initializations inherited from our superclass
58     super.init();
59     // Perform application initialization that must complete
60     // *before* managed components are initialized
61     // TODO – add your own initialization code here
62
63     // <editor–fold defaultstate="collapsed" desc="Managed Component Initialization">
64     // Initialize automatically managed components
65     // *Note* – this logic should NOT be modified
66     try {
67         _init();
68     } catch (Exception e) {
69         log("RequestBean1 Initialization Failure", e);
70         throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
71     }
72
73     // </editor–fold>
74     // Perform application initialization that must complete
75     // *after* managed components are initialized
76     // TODO – add your own initialization code here
77 }
78
79 /**
80  * <p>This method is called when this bean is removed from
81  * request scope. This occurs automatically when the corresponding
82  * HTTP response has been completed and sent to the client.</p>
83  *
84  * <p>You may customize this method to clean up resources allocated
85  * during the execution of the <code>init()</code> method, or
86  * at any later time during the lifetime of the request.</p>
87  */
88 @Override
89 public void destroy() {
90 }
91
92 /**
93  * <p>Return a reference to the scoped data bean.</p>
94  *
95  * @return reference to the scoped data bean
96  */
97 protected SessionBean1 getSessionBean1() {
98     return (SessionBean1) getBean("SessionBean1");
99 }
100
101 /**
102  * <p>Return a reference to the scoped data bean.</p>
103  *
104  * @return reference to the scoped data bean
105  */
106 protected ApplicationBean1 getApplicationBean1() {
107     return (ApplicationBean1) getBean("ApplicationBean1");
108 }

```


109
110

}

Clases

Usuarios.java

```

1  package wiki;
2  // Generated 26--may--2009 20:37:26 by Hibernate Tools 3.2.1.GA
3
4
5
6  /**
7   * Usuarios generated by hbm2java
8   */
9  public class Usuarios implements java.io.Serializable {
10
11
12     private String login;
13     private String password;
14     private String rolUsr;
15     private String nombreUsr;
16     private String apellidosUsr;
17     private String emailUsr;
18     private String paisUsr;
19     private String ciudadUsr;
20     private String telefonoUsr;
21
22     public Usuarios() {
23     }
24
25
26     public Usuarios(String login, String password, String rolUsr, String nombreUsr, String apellidosUsr, String
27         emailUsr) {
28         this.login = login;
29         this.password = password;
30         this.rolUsr = rolUsr;
31         this.nombreUsr = nombreUsr;
32         this.apellidosUsr = apellidosUsr;
33         this.emailUsr = emailUsr;
34     }
35     public Usuarios(String login, String password, String rolUsr, String nombreUsr, String apellidosUsr, String
36         emailUsr, String paisUsr, String ciudadUsr, String telefonoUsr) {
37         this.login = login;
38         this.password = password;
39         this.rolUsr = rolUsr;
40         this.nombreUsr = nombreUsr;
41         this.apellidosUsr = apellidosUsr;
42         this.emailUsr = emailUsr;
43         this.paisUsr = paisUsr;
44         this.ciudadUsr = ciudadUsr;
45         this.telefonoUsr = telefonoUsr;
46     }
47
48     public String getLogin() {
49         return this.login;
50     }
51
52     public void setLogin(String login) {
53         this.login = login;
54     }
55
56     public String getPassword() {
57         return this.password;
58     }
59
60     public void setPassword(String password) {
61         this.password = password;
62     }
63
64     public String getRolUsr() {
65         return this.rolUsr;
66     }
67
68     public void setRolUsr(String rolUsr) {
69         this.rolUsr = rolUsr;
70     }
71
72     public String getNombreUsr() {
73         return this.nombreUsr;
74     }
75
76     public void setNombreUsr(String nombreUsr) {
77         this.nombreUsr = nombreUsr;
78     }
79
80     public String getApellidosUsr() {
81         return this.apellidosUsr;
82     }
83
84     public void setApellidosUsr(String apellidosUsr) {
85         this.apellidosUsr = apellidosUsr;
86     }
87
88     public String getEmailUsr() {
89         return this.emailUsr;
90     }
91
92     public void setEmailUsr(String emailUsr) {
93         this.emailUsr = emailUsr;
94     }
95
96     public String getPaisUsr() {
97         return this.paisUsr;
98     }
99
100    public void setPaisUsr(String paisUsr) {
101        this.paisUsr = paisUsr;
102    }
103
104    public String getCiudadUsr() {
105        return this.ciudadUsr;
106    }
107
108    public void setCiudadUsr(String ciudadUsr) {
109        this.ciudadUsr = ciudadUsr;
110    }
111
112    public String getTelefonoUsr() {
113        return this.telefonoUsr;
114    }
115
116    public void setTelefonoUsr(String telefonoUsr) {
117        this.telefonoUsr = telefonoUsr;
118    }
119
120    }

```

```

55     }
56
57     public void setPassword(String password) {
58         this.password = password;
59     }
60     public String getRolUsr() {
61         return this.rolUsr;
62     }
63
64     public void setRolUsr(String rolUsr) {
65         this.rolUsr = rolUsr;
66     }
67     public String getNombreUsr() {
68         return this.nombreUsr;
69     }
70
71     public void setNombreUsr(String nombreUsr) {
72         this.nombreUsr = nombreUsr;
73     }
74     public String getApellidosUsr() {
75         return this.apellidosUsr;
76     }
77
78     public void setApellidosUsr(String apellidosUsr) {
79         this.apellidosUsr = apellidosUsr;
80     }
81     public String getEmailUsr() {
82         return this.emailUsr;
83     }
84
85     public void setEmailUsr(String emailUsr) {
86         this.emailUsr = emailUsr;
87     }
88     public String getPaisUsr() {
89         return this.paisUsr;
90     }
91
92     public void setPaisUsr(String paisUsr) {
93         this.paisUsr = paisUsr;
94     }
95     public String getCiudadUsr() {
96         return this.ciudadUsr;
97     }
98
99     public void setCiudadUsr(String ciudadUsr) {
100         this.ciudadUsr = ciudadUsr;
101     }
102     public String getTelefonoUsr() {
103         return this.telefonoUsr;
104     }
105
106     public void setTelefonoUsr(String telefonoUsr) {
107         this.telefonoUsr = telefonoUsr;
108     }
109
110
111
112 }
113

```

UsuarioAyuda.java

```

1  /*
2  * To change this template, choose Tools | Templates

```

```

3  * and open the template in the editor.
4  */
5
6  package wiki;
7
8  import java.util.List;
9  import org.hibernate.Session;
10
11 /**
12  *
13  * @author Administrador
14  */
15 public class UsuarioAyuda {
16     Session session = null;
17
18     public UsuarioAyuda() {
19         this.session = HibernateUtil.getSessionFactory().getCurrentSession();
20     }
21
22     public Usuarios buscarUsuarioLogin(String log, String pass){
23         Usuarios enc=null;
24
25         try {
26             session= HibernateUtil.getSessionFactory().getCurrentSession();
27             org.hibernate.Transaction tx = session.beginTransaction();
28             List<Usuarios> lista = (List<Usuarios>) session.createQuery("from Usuarios as usuarios where
29                 usuarios.login=' " + log+" ' and usuarios.password=' "+pass+" ' ").list();
30             if(!lista.isEmpty())
31                 enc=lista.get(0);
32
33             } catch (Exception e) {
34                 e.printStackTrace();
35             }
36
37         return enc;
38     }
39
40     public Usuarios buscarUsuario(String log){
41         Usuarios enc=null;
42
43         try {
44             session= HibernateUtil.getSessionFactory().getCurrentSession();
45             org.hibernate.Transaction tx = session.beginTransaction();
46             List<Usuarios> lista = (List<Usuarios>) session.createQuery("from Usuarios as usuarios where
47                 usuarios.login=' " + log+" ' ").list();
48             if(!lista.isEmpty())
49                 enc=lista.get(0);
50
51             } catch (Exception e) {
52                 e.printStackTrace();
53             }
54
55         return enc;
56     }
57
58     public void registrarUsuario(Usuarios nuevo)
59     {
60         try {
61             session= HibernateUtil.getSessionFactory().getCurrentSession();
62             org.hibernate.Transaction tx = session.beginTransaction();
63             session.save(nuevo);
64             session.getTransaction().commit();
65         } catch (Exception e) {
66             e.printStackTrace();
67         }
68     }
69 }

```

```

68
69 public void modificarUsuario(Usuarios modificado)
70 {
71     try {
72         session= HibernateUtil.getSessionFactory().getCurrentSession();
73         org.hibernate.Transaction tx = session.beginTransaction();
74         session.merge(modificado);
75         session.getTransaction().commit();
76     } catch (Exception e) {
77         e.printStackTrace();
78     }
79
80 }
81
82 public void eliminarUsuario(Usuarios eliminado)
83 {
84     try {
85         session= HibernateUtil.getSessionFactory().getCurrentSession();
86         org.hibernate.Transaction tx = session.beginTransaction();
87         session.delete(eliminado);
88         session.getTransaction().commit();
89     } catch (Exception e) {
90         e.printStackTrace();
91     }
92
93 }
94
95 public List<Usuarios> recuperarUsuarios(){
96
97     List<Usuarios> lusuarios=null;
98
99     try {
100         session= HibernateUtil.getSessionFactory().getCurrentSession();
101         org.hibernate.Transaction tx = session.beginTransaction();
102         lusuarios = (List<Usuarios>) session.createQuery("from Usuarios as usuarios where usuarios.
103             rolUsr=' UR' ").list();
104         session.getTransaction().commit();
105     } catch (Exception e) {
106         e.printStackTrace();
107     }
108     return lusuarios;
109 }
110
111 public List<Usuarios> recuperarUsuariosPermisos(Usuarios user,Paginas page){
112
113     List<Usuarios> lusuarios=null;
114
115     try {
116         session= HibernateUtil.getSessionFactory().getCurrentSession();
117         org.hibernate.Transaction tx = session.beginTransaction();
118         lusuarios = (List<Usuarios>) session.createQuery("from Usuarios as usuarios where usuarios.
119             rolUsr=' UR' ").list();
120         lusuarios.remove(user);
121
122         List<AdministracionPaginas> idPag=(List<AdministracionPaginas>)session.createQuery("from
123             AdministracionPaginas as AdministracionPaginas where AdministracionPaginas.id
124             .idPagina="+page.getIdPagina()).list();
125         int nPagProp=idPag.size();
126         for(int i=0;i<nPagProp;i++){
127             String log=idPag.get(i).getId().getLogin();
128             Usuarios userelim=(Usuarios)session.createQuery("from Usuarios as usuarios where
129                 login=' "+log+" ' ").uniqueResult();
130             if(lusuarios.contains(userelim))
131                 lusuarios.remove(userelim);
132         }
133         session.getTransaction().commit();
134     }

```

```

130     catch (Exception e) {
131         e.printStackTrace();
132     }
133     return usuarios;
134 }
135
136
137
138
139
140
141 }

```

Paginas.java

```

1  package wiki;
2  // Generated 26-may-2009 20:37:26 by Hibernate Tools 3.2.1.GA
3
4
5
6  /**
7   * Paginas generated by hbm2java
8   */
9  public class Paginas implements java.io.Serializable {
10
11
12     private long idPagina;
13     private String nombrePagina;
14     private String descripcionPagina;
15     private String contenido;
16     private boolean publicada;
17
18     public Paginas() {
19     }
20
21
22     public Paginas(long idPagina, String nombrePagina, String descripcionPagina, boolean publicada) {
23         this.idPagina = idPagina;
24         this.nombrePagina = nombrePagina;
25         this.descripcionPagina = descripcionPagina;
26         this.publicada = publicada;
27     }
28     public Paginas(long idPagina, String nombrePagina, String descripcionPagina, String contenido, boolean
        publicada) {
29         this.idPagina = idPagina;
30         this.nombrePagina = nombrePagina;
31         this.descripcionPagina = descripcionPagina;
32         this.contenido = contenido;
33         this.publicada = publicada;
34     }
35
36     public long getIdPagina() {
37         return this.idPagina;
38     }
39
40     public void setIdPagina(long idPagina) {
41         this.idPagina = idPagina;
42     }
43     public String getNombrePagina() {
44         return this.nombrePagina;
45     }
46
47     public void setNombrePagina(String nombrePagina) {
48         this.nombrePagina = nombrePagina;

```

```

49     }
50     public String getDescripcionPagina() {
51         return this.descripcionPagina;
52     }
53
54     public void setDescripcionPagina(String descripcionPagina) {
55         this.descripcionPagina = descripcionPagina;
56     }
57     public String getContenido() {
58         return this.contenido;
59     }
60
61     public void setContenido(String contenido) {
62         this.contenido = contenido;
63     }
64     public boolean isPublicada() {
65         return this.publicada;
66     }
67
68     public void setPublicada(boolean publicada) {
69         this.publicada = publicada;
70     }
71
72
73
74
75 }

```

PaginaAyuda.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8
9  import java.util.List;
10 import org.hibernate.Session;
11
12 /**
13  *
14  * @author Administrador
15  */
16 public class PaginaAyuda {
17     Session session = null;
18
19     public PaginaAyuda() {
20         this.session = HibernateUtil.getSessionFactory().getCurrentSession();
21     }
22
23
24     public long buscarUltimoId(){
25         long enc=0;
26
27         try {
28             session= HibernateUtil.getSessionFactory().getCurrentSession();
29             org.hibernate.Transaction tx = session.beginTransaction();
30             String x= session.createQuery("SELECT MAX(paginas.id) from Paginas as paginas").uniqueResult
31                 ().toString();
32             enc=Long.parseLong(x);
33         } catch (Exception e) {
34             e.printStackTrace();
35         }
36     }
37 }

```

```

34     }
35
36     return enc;
37 }
38
39 public Paginas buscarPaginaId(long id){
40     Paginas enc=null;
41
42     try {
43         session= HibernateUtil.getSessionFactory().getCurrentSession();
44         org.hibernate.Transaction tx = session.beginTransaction();
45         enc=(Paginas) session.createQuery("from Paginas as paginas where paginas.id=" + id).
46             uniqueResult();
47     } catch (Exception e) {
48         e.printStackTrace();
49     }
50
51     return enc;
52 }
53
54 public Paginas buscarPaginaNom(String nom){
55     Paginas enc=null;
56
57     try {
58         session= HibernateUtil.getSessionFactory().getCurrentSession();
59         org.hibernate.Transaction tx = session.beginTransaction();
60         enc=(Paginas) session.createQuery("from Paginas as paginas where paginas.nombrePagina=' "
61             + nom+" ' ").uniqueResult();
62     } catch (Exception e) {
63         e.printStackTrace();
64     }
65
66     return enc;
67 }
68
69 public void registrarPagina(Paginas p)
70 {
71     try {
72         session= HibernateUtil.getSessionFactory().getCurrentSession();
73         org.hibernate.Transaction tx = session.beginTransaction();
74         session.save(p);
75         session.getTransaction().commit();
76     } catch (Exception e) {
77         e.printStackTrace();
78     }
79 }
80
81 public void modificarPagina(Paginas modificado)
82 {
83     try {
84         session= HibernateUtil.getSessionFactory().getCurrentSession();
85         org.hibernate.Transaction tx = session.beginTransaction();
86         session.merge(modificado);
87         session.getTransaction().commit();
88     } catch (Exception e) {
89         e.printStackTrace();
90     }
91 }
92
93 public void eliminarPaginas(Usuarios eliminado)
94 {
95     try {
96         session= HibernateUtil.getSessionFactory().getCurrentSession();
97         org.hibernate.Transaction tx = session.beginTransaction();
98         session.delete(eliminado);

```

```

99     session.getTransaction().commit();
100 } catch (Exception e) {
101     e.printStackTrace();
102 }
103
104 }
105
106 public List<Paginas> recuperarPaginas(String log,String rol){
107
108     //Segun el rol se recupera una lista de pginas distinta
109     //Si es administrador se recuperan todas las pginas
110     //Si es usuario registrado, se recuperan las suyas para administrarlas
111     //y las publicadas para visitarlas
112     //Si es invitado, recupera las publicadas para visualizarlas
113
114     List<Paginas> paginas=null;
115
116     try {
117         session= HibernateUtil.getSessionFactory().getCurrentSession();
118         org.hibernate.Transaction tx = session.beginTransaction();
119         if(rol.equals("AD"))
120             paginas = (List<Paginas>) session.createQuery("from Paginas as paginas").list();
121         else if(rol.equals("UR")) //Modificar esta consulta
122         {
123             paginas = (List<Paginas>) session.createQuery("from Paginas as paginas where paginas.
124                 publicada=true").list();
125
126             List<AdministracionPaginas> idPag=(List<AdministracionPaginas>)session.createQuery("from
127                 AdministracionPaginas as AdministracionPaginas where AdministracionPaginas
128                 .id.login=' "+log+"' ").list();
129             int nPagProp=idPag.size();
130             for(int i=0;i<nPagProp;i++){
131                 long id=idPag.get(i).getId().getIdPagina();
132                 Paginas p=(Paginas)session.createQuery("from Paginas as paginas where id="+id).
133                     uniqueResult();
134
135                 if(!paginas.contains(p))
136                     paginas.add(p);
137             }
138         }
139         else
140             paginas = (List<Paginas>) session.createQuery("from Paginas as paginas where paginas.
141                 publicada=true").list();
142         session.getTransaction().commit();
143     } catch (Exception e) {
144         e.printStackTrace();
145     }
146     return paginas;
147 }
148
149 public List<AdministracionPaginas> permisosModificacion(String log,String rol, long id){
150
151     List<AdministracionPaginas> idPag;
152
153     session= HibernateUtil.getSessionFactory().getCurrentSession();
154     org.hibernate.Transaction tx = session.beginTransaction();
155
156     idPag=(List<AdministracionPaginas>)session.createQuery("from AdministracionPaginas as
157         AdministracionPaginas where AdministracionPaginas.id.login=' "+log+"' and
158         id_pagina=' "+id+"' ").list();
159     return idPag;
160 }
161
162 public void registrarPaginaUsuario(AdministracionPaginas ap)
163 {

```



```

159     try {
160         session= HibernateUtil.getSessionFactory().getCurrentSession();
161         org.hibernate.Transaction tx = session.beginTransaction();
162         session.save(ap);
163         session.getTransaction().commit();
164     } catch (Exception e) {
165         e.printStackTrace();
166     }
167
168 }
169
170 }

```

Gestión de usuarios

index.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8  import com.sun.rave.web.ui.appbase.AbstractPageBean;
9  import com.sun.webui.jsf.component.Form;
10 import com.sun.webui.jsf.component.Label;
11 import com.sun.webui.jsf.component.PasswordField;
12 import com.sun.webui.jsf.component.TextArea;
13 import com.sun.webui.jsf.component.TextField;
14 import javax.faces.FacesException;
15 import javax.faces.event.ValueChangeEvent;
16
17 /**
18  * <p>Page bean that corresponds to a similarly named JSP page. This
19  * class contains component definitions (and initialization code) for
20  * all components that you have defined on this page, as well as
21  * lifecycle methods and event handlers where you may add behavior
22  * to respond to incoming events.</p>
23  *
24  * @version index2.java
25  * @version Created on 04-may-2009, 19:13:41
26  * @author WIND MOTEBOOK
27  */
28
29 public class index extends AbstractPageBean {
30     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
31
32     /**
33      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
34      * This method is automatically generated, so any user-specified code inserted
35      * here is subject to being replaced.</p>
36      */
37     private void _init() throws Exception {
38     }
39     private TextField textField1 = new TextField();
40
41     public TextField getTextField1() {
42         return textField1;
43     }
44
45     public void setTextField1(TextField tf) {
46         this.textField1 = tf;

```

```

47     }
48     private PasswordField passwordField1 = new PasswordField();
49
50     public PasswordField getPasswordField1() {
51         return passwordField1;
52     }
53
54     public void setPasswordField1(PasswordField pf) {
55         this.passwordField1 = pf;
56     }
57     private Form form2 = new Form();
58
59     public Form getForm2() {
60         return form2;
61     }
62
63     public void setForm2(Form f) {
64         this.form2 = f;
65     }
66     private Label label8 = new Label();
67
68     public Label getLabel8() {
69         return label8;
70     }
71
72     public void setLabel8(Label l) {
73         this.label8 = l;
74     }
75     private TextArea textArea1 = new TextArea();
76
77     public TextArea getTextArea1() {
78         return textArea1;
79     }
80
81     public void setTextArea1(TextArea ta) {
82         this.textArea1 = ta;
83     }
84
85     // </editor-fold>
86
87     /**
88      * <p>Construct a new Page bean instance.</p>
89      */
90     public index() {
91     }
92
93     /**
94      * <p>Callback method that is called whenever a page is navigated to,
95      * either directly via a URL, or indirectly via page navigation.
96      * Customize this method to acquire resources that will be needed
97      * for event handlers and lifecycle methods, whether or not this
98      * page is performing post back processing.</p>
99      *
100     * <p>Note that, if the current request is a postback, the property
101     * values of the components do <strong>not</strong> represent any
102     * values submitted with this request. Instead, they represent the
103     * property values that were saved for this view when it was rendered.</p>
104     */
105     @Override
106     public void init() {
107         // Perform initializations inherited from our superclass
108         super.init();
109         // Perform application initialization that must complete
110         // *before* managed components are initialized
111         // TODO – add your own initialization code here
112
113         // <editor-fold defaultstate="collapsed" desc="Managed Component Initialization">

```

```

114 // Initialize automatically managed components
115 // *Note* – this logic should NOT be modified
116 try {
117     _init();
118 } catch (Exception e) {
119     log("index2 Initialization Failure", e);
120     throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
121 }
122
123 // </editor-fold>
124 // Perform application initialization that must complete
125 // *after* managed components are initialized
126 // TODO – add your own initialization code here
127 }
128
129 /**
130  * <p>Callback method that is called after the component tree has been
131  * restored, but before any event processing takes place. This method
132  * will <strong>only</strong> be called on a postback request that
133  * is processing a form submit. Customize this method to allocate
134  * resources that will be required in your event handlers.</p>
135  */
136 @Override
137 public void preprocess() {
138 }
139
140 /**
141  * <p>Callback method that is called just before rendering takes place.
142  * This method will <strong>only</strong> be called for the page that
143  * will actually be rendered (and not, for example, on a page that
144  * handled a postback and then navigated to a different page). Customize
145  * this method to allocate resources that will be required for rendering
146  * this page.</p>
147  */
148 @Override
149 public void prerender() {
150 }
151
152 /**
153  * <p>Callback method that is called after rendering is completed for
154  * this request, if <code>init()</code> was called (regardless of whether
155  * or not this was the page that was actually rendered). Customize this
156  * method to release resources acquired in the <code>init()</code>,
157  * <code>preprocess()</code>, or <code>prerender()</code> methods (or
158  * acquired during execution of an event handler).</p>
159  */
160 @Override
161 public void destroy() {
162 }
163
164 /**
165  * <p>Return a reference to the scoped data bean.</p>
166  *
167  * @return reference to the scoped data bean
168  */
169 protected SessionBean1 getSessionBean1() {
170     return (SessionBean1) getBean("SessionBean1");
171 }
172
173 /**
174  * <p>Return a reference to the scoped data bean.</p>
175  *
176  * @return reference to the scoped data bean
177  */
178 protected RequestBean1 getRequestBean1() {
179     return (RequestBean1) getBean("RequestBean1");
180 }

```

```

181
182  /**
183   * <p>Return a reference to the scoped data bean.</p>
184   *
185   * @return reference to the scoped data bean
186   */
187  protected ApplicationBean1 getApplicationBean1() {
188      return (ApplicationBean1) getBean("ApplicationBean1");
189  }
190
191  public void textArea1_processValueChange(ValueChangeEvent event) {
192  }
193
194  public String hyperlink1_action() {
195      // TODO: Process the action. Return value is a navigation
196      // case name where null will return to the same page.
197      this.getSessionBean1().setLogin(" ");
198      this.getSessionBean1().setRolUsr(" ");
199      return "GoToPagVisita";
200  }
201
202  public String button1_action() {
203      // TODO: Process the action. Return value is a navigation
204      // case name where null will return to the same page.
205      String pagina=null;
206      String log=(String)textField1.getValue();
207      String pass=(String)passwordField1.getValue();
208
209      if(log == null || pass == null)
210      {
211          textField1.setText(" ");
212          passwordField1.setText(" ");
213          label8.setVisible(true);
214      }
215      else
216      {
217          UsuarioAyuda a=new UsuarioAyuda();
218          Usuarios buscado=a.buscarUsuarioLogin(log,pass);
219          if(buscado!=null)
220          {
221              this.getSessionBean1().setLogin(log);
222              this.getSessionBean1().setNombreUsr(buscado.getNombreUsr());
223              this.getSessionBean1().setPassword(pass);
224              this.getSessionBean1().setApellidosUsr(buscado.getApellidosUsr());
225              this.getSessionBean1().setCiudadUsr(buscado.getCiudadUsr());
226              this.getSessionBean1().setEmailUsr(buscado.getEmailUsr());
227              this.getSessionBean1().setPaisUsr(buscado.getPaisUsr());
228              this.getSessionBean1().setRolUsr(buscado.getRolUsr());
229              this.getSessionBean1().setTelefonoUsr(buscado.getTelefonoUsr());
230              if(buscado.getRolUsr().equals("AD"))
231                  return "nextPageAd";
232              if (buscado.getRolUsr().equals("UR"))
233                  return "nextPage";
234          }
235          else{
236              textField1.setText(" ");
237              passwordField1.setText(" ");
238              label8.setVisible(true);
239          }
240      }
241      return pagina;
242  }
243
244
245
246  public String hyperlink2_action() {
247      // TODO: Process the action. Return value is a navigation

```

```

248      // case name where null will return to the same page.
249      return null;
250  }
251
252 }

```

index.jsp

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!--
3      Document : index2
4      Created on : 04-may-2009, 19:13:41
5      Author : WIND MOTEBOOK
6  -->
7  <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
   html" xmlns:jsf="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
   webuijsf">
8      <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9      <f:view>
10         <webuijsf:page id="page1">
11             <webuijsf:html id="html1">
12                 <webuijsf:head id="head1">
13                     <webuijsf:link id="link1" url="/resources/styleSheet.css"/>
14                 </webuijsf:head>
15                 <webuijsf:body id="body1" style="-rave-layout: grid">
16                     <webuijsf:form id="form1">
17                         <webuijsf:label id="label1"
18                             style="background-color: rgb(102, 102, 102); height: 22px; left:
19                             696px; top: 0px; position: absolute; width: 166px" text="Jos
20                             Alberto Cordero Ros"/>
21                         <webuijsf:label id="label2"
22                             style="background-color: rgb(153, 153, 153); height: 22px; left:
23                             696px; top: 24px; position: absolute; width: 166px" text="
24                             Lorena Gutierrez Madroal"/>
25                         <webuijsf:label id="label3"
26                             style="background-color: rgb(255, 255, 255); height: 22px; left:
27                             696px; top: 72px; position: absolute; width: 166px" text="Flix
28                             Reyes Fernndez"/>
29                         <webuijsf:label id="label4"
30                             style="background-color: rgb(204, 204, 204); height: 22px; left:
31                             696px; top: 48px; position: absolute; width: 166px" text="
32                             Sergio Puerta Aragn"/>
33                         <webuijsf:label id="label5"
34                             style="font-family: 'Times New Roman','Times', serif; font-size: 36
35                             px; font-style: oblique; font-weight: lighter; left: 24px;
36                             top: 24px; position: absolute; text-decoration: underline;
37                             width: 408px" text="Wiki 4Proyect - Andaluca"/>
38                         <webuijsf:image height="96" id="image1" style="left: 600px; top: 96px;
39                             position: absolute"
40                             url="http://upload.wikimedia.org/wikipedia/commons/thumb/f/f7/
41                             P_wiki_letter_w.svg/400px-P_wiki_letter_w.svg.png" width="120
42                             "/>
43                     </webuijsf:form>
44                     <webuijsf:form binding="#{index.form2}" id="form2" style="border-top: solid rgb
45                     (128, 128, 128) 2px; height: 334px; left: 48px; top: 144px;
46                     position: absolute; width: 840px">
47                         <webuijsf:label id="label6"
48                             style="border-width: 2px; border-color: black; background-color:
49                             rgb(255, 255, 255); font-family: 'Times New Roman','Times',
50                             serif; font-size: 18px; font-style: italic; font-weight:
51                             lighter; height: 20px; left: 168px; top: 72px; position:
52                             absolute; width: 46px" text="login"/>
53                     <webuijsf:label id="label7"

```

```

34         style="border-width: 2px; border-color: black; background-color:
           rgb(255, 255, 255); font-family: 'Times New Roman','Times',
           serif; font-size: 18px; font-style: italic; font-weight:
           lighter; height: 20px; left: 168px; top: 120px; position:
           absolute; width: 46px" text="password"/>
35     <webuijsf:textField binding="#{index.textField1}" id="textField1" style="
           position: absolute; left: 288px; top: 72px" tabIndex="1"/>
36     <webuijsf:hyperlink actionExpression="#{index.hyperlink1_action}" id="
           hyperlink1"
37         style="color: rgb(102, 102, 102); font-family: Verdana,Arial,
           Helvetica,sans-serif; font-size: 14px; font-weight: bold;
           left: 360px; top: 264px; position: absolute"
38         tabIndex="5" text="Entrar como invitado"/>
39     <webuijsf:button actionExpression="#{index.button1_action}" id="button1"
40         style="border: 2px solid black; height: 22px; left: 239px; top:
           192px; position: absolute; width: 119px" tabIndex="3" text="
           Entrar"/>
41     <webuijsf:hyperlink actionExpression="#{index.hyperlink2_action}" id="
           hyperlink2"
42         style="color: rgb(102, 102, 102); font-family: Verdana,Arial,
           Helvetica,sans-serif; font-size: 14px; font-weight: bold;
           left: 120px; top: 264px; position: absolute"
43         tabIndex="4" text="Quiero registrarme!" url="/faces/RegistroUsuario.
           jsp"/>
44     <webuijsf:passwordField binding="#{index.passwordField1}" id="
           passwordField1"
45         style="height: 24px; left: 288px; top: 120px; position: absolute;
           width: 120px" tabIndex="2"/>
46     <webuijsf:textArea binding="#{index.textArea1}" columns="30" disabled="true"
47         id="textArea1" rows="10"
           style="left: 552px; top: 48px; position: absolute" text=" Bienvenido
           !&#xa;&#xa;Wiki 4Proyect-Andaluca es una &#xa;wiki creada
           por cuatro estudiantes &#xa;de la Ingeniera Informtica de &#
           xa;la Universidad de Cdiz.&#xa;&#xa;La elaboracin de la wiki
           surgi &#xa;como proyecto final para la &#xa;asignatura de
           IW (Ingeniera Web),&#xa;una asignatura optativa de la &#xa;
           titulacin."/>
48     <webuijsf:image height="286" id="image2" style="left: 648px; top: 216px;
           position: absolute" url="/resources/mundo.png" width="448"/>
49     <webuijsf:imageHyperlink height="166" id="imageHyperlink1"
50         imageURL="http://anahigonzalet.files.wordpress.com/2008/06/
           foto_diseno_web.jpg?w=300&h=206"
51         style="left: 216px; top: 432px; position: absolute" url="https://
           www.assembla.com/wiki/show/andalucia" width="240"/>
52     <webuijsf:label binding="#{index.label8}" id="label8"
53         style="border-width: 1px; border-style: solid; border-color: rgb
           (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
           background-color: rgb(255, 204, 102); color: rgb(102, 0, 0);
           font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: italic; font-weight: bold; height:
           46px; left: 144px; top: 312px; position: absolute; text-
           align: center; vertical-align: baseline; width: 406px"
54         text="El login o el password introducidos son incorrectos por
           favor intntalo de nuevo" visible="false"/>
55     <webuijsf:label id="label9"
56         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: oblique; font-weight: bold; height:
           34px; left: 240px; top: 384px; position: absolute; text-
           align: center; vertical-align: baseline; width: 192px" text="
           Haz click en la imagen y... Entra en nuestra web!"/>
57     </webuijsf:form>
58     </webuijsf:body>
59     </webuijsf:html>
60     </webuijsf:page>
61     </f:view>
62 </jsp:root>

```

PaginaUsuario.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8
9  import com.sun.rave.web.ui.appbase.AbstractPageBean;
10 import com.sun.webui.jsf.component.Form;
11 import com.sun.webui.jsf.component.Label;
12 import com.sun.webui.jsf.component.PasswordField;
13 import javax.faces.FacesException;
14 import javax.faces.component.UIComponent;
15 import javax.faces.context.FacesContext;
16 import javax.faces.validator.LengthValidator;
17
18 /**
19  * <p>Page bean that corresponds to a similarly named JSP page. This
20  * class contains component definitions (and initialization code) for
21  * all components that you have defined on this page, as well as
22  * lifecycle methods and event handlers where you may add behavior
23  * to respond to incoming events.</p>
24  *
25  * @version PaginaUsuario.java
26  * @version Created on 04-may-2009, 21:41:25
27  * @author WIND MOTEBOOK
28  */
29
30 public class PaginaUsuario extends AbstractPageBean {
31     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
32
33     /**
34      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
35      * This method is automatically generated, so any user-specified code inserted
36      * here is subject to being replaced.</p>
37      */
38     private void _init() throws Exception {
39         lengthValidator1.setMinimum(4);
40     }
41
42     // </editor-fold>
43     private UsuarioAyuda ayuda=new UsuarioAyuda();
44     private PasswordField passwordField1 = new PasswordField();
45
46     public PasswordField getPasswordField1() {
47         return passwordField1;
48     }
49
50     public void setPasswordField1(PasswordField pf) {
51         this.passwordField1 = pf;
52     }
53     private PasswordField passwordField2 = new PasswordField();
54
55     public PasswordField getPasswordField2() {
56         return passwordField2;
57     }
58
59     public void setPasswordField2(PasswordField pf) {
60         this.passwordField2 = pf;
61     }
62     private PasswordField passwordField3 = new PasswordField();
63
64     public PasswordField getPasswordField3() {

```

```

65         return passwordField3;
66     }
67
68     public void setPasswordField3(PasswordField pf) {
69         this.passwordField3 = pf;
70     }
71     private Form form2 = new Form();
72
73     public Form getForm2() {
74         return form2;
75     }
76
77     public void setForm2(Form f) {
78         this.form2 = f;
79     }
80     private LengthValidator lengthValidator1 = new LengthValidator();
81
82     public LengthValidator getLengthValidator1() {
83         return lengthValidator1;
84     }
85
86     public void setLengthValidator1(LengthValidator lv) {
87         this.lengthValidator1 = lv;
88     }
89     private Label labelGorda = new Label();
90
91     public Label getLabelGorda() {
92         return labelGorda;
93     }
94
95     public void setLabelGorda(Label l) {
96         this.labelGorda = l;
97     }
98     private Label label19 = new Label();
99
100    public Label getLabel19() {
101        return label19;
102    }
103
104    public void setLabel19(Label l) {
105        this.label19 = l;
106    }
107    private Label label18 = new Label();
108
109    public Label getLabel18() {
110        return label18;
111    }
112
113    public void setLabel18(Label l) {
114        this.label18 = l;
115    }
116
117    /**
118     * <p>Construct a new Page bean instance.</p>
119     */
120    public PaginaUsuario() {
121    }
122
123    /**
124     * <p>Callback method that is called whenever a page is navigated to,
125     * either directly via a URL, or indirectly via page navigation.
126     * Customize this method to acquire resources that will be needed
127     * for event handlers and lifecycle methods, whether or not this
128     * page is performing post back processing.</p>
129     *
130     * <p>Note that, if the current request is a postback, the property
131     * values of the components do <strong>not</strong> represent any

```



```

132  * values submitted with this request. Instead, they represent the
133  * property values that were saved for this view when it was rendered.</p>
134  */
135  @Override
136  public void init() {
137      // Perform initializations inherited from our superclass
138      super.init();
139      // Perform application initialization that must complete
140      // *before* managed components are initialized
141      // TODO – add your own initialization code here
142
143      // <editor–fold defaultstate="collapsed" desc="Managed Component Initialization">
144      // Initialize automatically managed components
145      // *Note* – this logic should NOT be modified
146      try {
147          _init();
148      } catch (Exception e) {
149          log("PaginaUsuario Initialization Failure", e);
150          throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
151      }
152
153      // </editor–fold>
154      // Perform application initialization that must complete
155      // *after* managed components are initialized
156      // TODO – add your own initialization code here
157      //String log=(String)label2.getText();
158      // System.out.println(log+" aaa");
159
160  }
161
162
163  /**
164   * <p>Callback method that is called after the component tree has been
165   * restored, but before any event processing takes place. This method
166   * will <strong>only</strong> be called on a postback request that
167   * is processing a form submit. Customize this method to allocate
168   * resources that will be required in your event handlers.</p>
169   */
170  @Override
171  public void preprocess() {
172  }
173
174  /**
175   * <p>Callback method that is called just before rendering takes place.
176   * This method will <strong>only</strong> be called for the page that
177   * will actually be rendered (and not, for example, on a page that
178   * handled a postback and then navigated to a different page). Customize
179   * this method to allocate resources that will be required for rendering
180   * this page.</p>
181   */
182  @Override
183  public void prerender() {
184  }
185
186  /**
187   * <p>Callback method that is called after rendering is completed for
188   * this request, if <code>init()</code> was called (regardless of whether
189   * or not this was the page that was actually rendered). Customize this
190   * method to release resources acquired in the <code>init()</code>,
191   * <code>preprocess()</code>, or <code>prerender()</code> methods (or
192   * acquired during execution of an event handler).</p>
193   */
194  @Override
195  public void destroy() {
196  }
197
198  /**

```

```

199      * <p>Return a reference to the scoped data bean.</p>
200      *
201      * @return reference to the scoped data bean
202      */
203      protected SessionBean1 getSessionBean1() {
204          return (SessionBean1) getBean("SessionBean1");
205      }
206
207      /**
208      * <p>Return a reference to the scoped data bean.</p>
209      *
210      * @return reference to the scoped data bean
211      */
212      protected ApplicationBean1 getApplicationBean1() {
213          return (ApplicationBean1) getBean("ApplicationBean1");
214      }
215
216      /**
217      * <p>Return a reference to the scoped data bean.</p>
218      *
219      * @return reference to the scoped data bean
220      */
221      protected RequestBean1 getRequestBean1() {
222          return (RequestBean1) getBean("RequestBean1");
223      }
224
225      public void textField3_validate(FacesContext context, UIComponent component, Object value) {
226      }
227
228      public String hyperlink1_action()
229      {
230          // TODO: Process the action. Return value is a navigation
231          // case name where null will return to the same page.
232
233
234
235          String nombre=this.getSessionBean1().getNombreUsr();
236          String log=this.getSessionBean1().getLogin();
237          String pass=this.getSessionBean1().getPassword();
238          String apell=this.getSessionBean1().getApellidosUsr();
239          String correo=this.getSessionBean1().getEmailUsr();
240          String pais=this.getSessionBean1().getPaisUsr();
241          String ciudad=this.getSessionBean1().getCiudadUsr();
242          String telefono=this.getSessionBean1().getTelefonoUsr();
243          String passaux1 = (String) passwordField2.getValue();
244          String passaux2 = (String) passwordField3.getValue();
245          String passActual = (String) passwordField1.getValue();
246          if(pass == null || passaux1 == null || passaux2 == null || nombre.isEmpty() || apell.isEmpty() || correo.
                isEmpty() || passaux1.length() < 4 )
247          {
248              labelGorda.setVisible(true);
249              label19.setVisible(false);
250              label18.setVisible(false);
251          }
252          else if(!passActual.equals(pass))
253          {
254
255              labelGorda.setVisible(false);
256              label19.setVisible(false);
257              label18.setVisible(true);
258
259          }
260          else
261          {
262              Usuarios modUsuario=new Usuarios(log,pass,"UR", nombre, apell, correo);
263              modUsuario.setPaisUsr(pais);
264              modUsuario.setCiudadUsr(ciudad);

```

```

265         modUsuario.setTelefonoUsr(telefono);
266
267
268         if(passaux1.equals(passaux2))
269         {
270             modUsuario.setPassword(passaux1);
271         }
272         ayuda.modificarUsuario(modUsuario);
273         label19.setVisible(true);
274         labelGorda.setVisible(false);
275         label18.setVisible(false);
276     }
277     return null;
278 }
279
280 public String hyperlink3_action() {
281     // TODO: Process the action. Return value is a navigation
282     // case name where null will return to the same page.
283     return "GoToAdministrarP";
284 }
285
286 public String hyperlink4_action() {
287     // TODO: Process the action. Return value is a navigation
288     // case name where null will return to the same page.
289     return "GoToListarP";
290 }
291
292 public String hyperlink5_action() {
293     // TODO: Process the action. Return value is a navigation
294     // case name where null will return to the same page.
295     return "GoToCrearP";
296 }
297
298 public String hyperlink2_action() {
299     // TODO: Process the action. Return value is a navigation
300     // case name where null will return to the same page.
301     return null;
302 }
303 }

```

PaginaUsuario.jsp

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : PaginaUsuario
4     Created on : 04-may-2009, 21:41:25
5     Author : WIND MOTEBOOK
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
8     html" xmlns:jsp="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
9     webuijsf">
10     <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
11     <f:view>
12         <webuijsf:page id="page1">
13             <webuijsf:html id="html1">
14                 <webuijsf:head id="head1">
15                     <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
16                 </webuijsf:head>
17                 <webuijsf:body id="body1" style="--rave-layout: grid">
18                     <webuijsf:form id="form1">
19                         <webuijsf:label id="label1"
20                             style="font-family: 'Times New Roman','Times', serif; font-size: 36
21                                 px; font-style: oblique; font-weight: lighter; left: 24px;
22                                 top: 24px; position: absolute; text-decoration: underline;

```

```

width: 408px" text="Wiki 4Proyect - Andaluca"/>
19 <webuijsf:label id="label2"
20 style="font-family: Verdana,Arial,Helvetica,sans-serif; font-size:
12px; font-style: oblique; font-weight: bold; left: 408px;
top: 96px; position: absolute; text-decoration: underline;
width: 190px" text="#{SessionBean1.login}"/>
21 <webuijsf:label id="label3"
22 style="background-color: rgb(255, 255, 255); height: 22px; left:
696px; top: 72px; position: absolute; width: 166px" text="Flix
Reyes Fernndez"/>
23 <webuijsf:label id="label4"
24 style="background-color: rgb(204, 204, 204); height: 22px; left:
696px; top: 48px; position: absolute; width: 166px" text="
Sergio Puerta Aragn"/>
25 <webuijsf:label id="label5"
26 style="background-color: rgb(153, 153, 153); height: 22px; left:
696px; top: 24px; position: absolute; width: 166px" text="
Lorena Gutierrez Madroal"/>
27 <webuijsf:label id="label6"
28 style="background-color: rgb(102, 102, 102); height: 22px; left:
696px; top: 0px; position: absolute; width: 166px" text="Jos
Alberto Cordero Ros"/>
29 <webuijsf:label id="label14"
30 style="font-family: Verdana,Arial,Helvetica,sans-serif; font-size:
12px; font-style: oblique; font-weight: lighter; left: 264
px; top: 96px; position: absolute" text="Bienvenido Usuario"/>
31 <webuijsf:image height="96" id="image2" style="left: 600px; top: 96px;
position: absolute"
32 url="http://upload.wikimedia.org/wikipedia/commons/thumb/f/f7/
P_wiki_letter_w.svg/400px-P_wiki_letter_w.svg.png" width="120
"/>
33 </webuijsf:form>
34 <webuijsf:form binding="#{PaginaUsuario.form2}" id="form2" style="border-top:
solid rgb(102, 102, 102) 2px; height: 550px; left: 48px; top: 144px
; position: absolute; width: 840px">
35 <webuijsf:hyperlink actionExpression="#{PaginaUsuario.hyperlink4_action}" id=
"hyperlink4"
36 style="background-color: rgb(204, 204, 204); color: black; font-
family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
font-style: oblique; font-weight: bold; height: 24px; left:
24px; top: 48px; position: absolute; text-decoration: none;
width: 168px"
37 tabIndex="13" text="Listar pginas"/>
38 <webuijsf:hyperlink actionExpression="#{PaginaUsuario.hyperlink5_action}" id=
"hyperlink5"
39 style="background-color: rgb(153, 153, 153); color: black; font-
family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
font-style: oblique; font-weight: bold; height: 24px; left:
24px; top: 72px; position: absolute; text-decoration: none;
width: 168px"
40 tabIndex="14" text="Crear pgina"/>
41 <webuijsf:hyperlink actionExpression="#{PaginaUsuario.hyperlink1_action}" id=
"hyperlink1"
42 style="color: gray; font-family: 'Verdana','Arial','Helvetica',
sans-serif; font-size: 12px; font-style: oblique; font-
weight: bold; left: 720px; top: 48px; position: absolute;
width: 118px"
43 tabIndex="10" text="Guardar cambios"/>
44 <webuijsf:label id="label7"
45 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
-size: 12px; font-style: italic; font-weight: lighter;
height: 24px; left: 288px; top: 120px; position: absolute;
width: 96px" text="Nombre (*)"/>
46 <webuijsf:textField columns="30" id="textField1" maxLength="30"
47 onChange="webui.suntheme4_2.common.timeoutSubmitForm(this.form, '
textField1');"
```

```

48         style="left: 456px; top: 120px; position: absolute; z-index: 500"
49         tabIndex="1" text="#{SessionBean1.nombreUsr}"/>
50     <webuijsf:textField columns="30" id="textField2" maxLength="50" style="left:
51         456px; top: 168px; position: absolute" tabIndex="2" text="#{
52         SessionBean1.apellidosUsr}"/>
53     <webuijsf:label id="label8"
54         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
55             -size: 12px; font-style: italic; font-weight: lighter;
56             height: 24px; left: 288px; top: 168px; position: absolute;
57             width: 96px" text="Apellidos (*)"/>
58     <webuijsf:textField columns="30" id="textField4" maxLength="20" style="left:
59         456px; top: 408px; position: absolute" tabIndex="7" text="#{
60         SessionBean1.paisUsr}"/>
61     <webuijsf:label id="label10"
62         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
63             -size: 12px; font-style: italic; font-weight: lighter;
64             height: 24px; left: 288px; top: 408px; position: absolute;
65             width: 96px" text="Pas"/>
66     <webuijsf:textField columns="30" id="textField5" maxLength="20" style="left:
67         456px; top: 456px; position: absolute" tabIndex="8" text="#{
68         SessionBean1.ciudadUsr}"/>
69     <webuijsf:label id="label11"
70         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
71             -size: 12px; font-style: italic; font-weight: lighter;
72             height: 24px; left: 288px; top: 456px; position: absolute;
73             width: 96px" text="Ciudad"/>
74     <webuijsf:label id="label12"
75         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
76             -size: 12px; font-style: italic; font-weight: lighter;
77             height: 24px; left: 288px; top: 504px; position: absolute;
78             width: 96px" text="Telefono"/>
79     <webuijsf:textField columns="30" id="textField6" maxLength="9" style="left:
80         456px; top: 504px; position: absolute" tabIndex="9" text="#{
81         SessionBean1.telefonoUsr}"/>
82     <webuijsf:hyperlink actionExpression="#{PaginaUsuario.hyperlink2_action}" id=
83         "hyperlink2"
84         style="background-color: rgb(102, 102, 102); color: white; font-
85             family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
86             font-style: oblique; font-weight: bold; height: 22px; left:
87             24px; top: 24px; position: absolute; text-decoration: none;
88             width: 166px"
89         tabIndex="11" text="Ir a Bienvenida" url="/faces/index.jsp"/>
90     <webuijsf:textField columns="30" id="textField7" maxLength="30" style="left:
91         456px; top: 216px; position: absolute" tabIndex="3" text="#{
92         SessionBean1.emailUsr}"/>
93     <webuijsf:label id="label13"
94         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
95             -size: 12px; font-style: italic; font-weight: lighter;
96             height: 24px; left: 288px; top: 216px; position: absolute;
97             width: 142px" text="Correo electrnico (*)"/>
98     <webuijsf:label id="label9"
99         style="color: black; font-family: 'Verdana','Arial','Helvetica',
100             sans-serif; font-size: 12px; font-style: oblique; font-
101             weight: lighter; height: 46px; left: 240px; top: 48px;
102             position: absolute; width: 430px" text="Si vas a modificar tus
103             datos, recuerda que los que tienen el asterisco son campos
104             obligatorios. No olvides darle a &quot;Guardar cambios&quot;
105             cuando hayas finalizado!"/>
106     <webuijsf:label id="label15"
107         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
108             -size: 12px; font-style: italic; font-weight: lighter;
109             height: 24px; left: 288px; top: 264px; position: absolute;
110             width: 142px" text="Password actual(*)"/>
111     <webuijsf:label id="label16"
112         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
113             -size: 12px; font-style: italic; font-weight: lighter;
114             height: 24px; left: 288px; top: 312px; position: absolute;

```

```

width: 142px" text="Password nueva(*)"/>
73 <webuijsf:label id="label17"
74 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
-size: 12px; font-style: italic; font-weight: lighter;
height: 24px; left: 288px; top: 360px; position: absolute;
width: 166px" text="Repite password nueva(*)"/>
75 <webuijsf:passwordField binding="#{PaginaUsuario.passwordField1}" columns="
30" id="passwordField1" maxLength="20"
76 style="position: absolute; left: 456px; top: 264px" tabIndex="4"/>
77 <webuijsf:passwordField binding="#{PaginaUsuario.passwordField2}" columns="
30" id="passwordField2" maxLength="20"
78 style="position: absolute; left: 456px; top: 312px" tabIndex="5"/>
79 <webuijsf:passwordField binding="#{PaginaUsuario.passwordField3}" columns="
30" id="passwordField3" maxLength="20"
80 style="position: absolute; left: 456px; top: 360px" tabIndex="6"/>
81 <webuijsf:label binding="#{PaginaUsuario.labelGorda}" id="labelGorda"
82 style="border-width: 1px; border-style: solid; border-color: rgb
(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
font-family: 'Verdana','Arial','Helvetica',sans-serif; font
-size: 12px; font-style: oblique; font-weight: bold; height:
68px; left: 672px; top: 216px; position: absolute; text-
align: center; vertical-align: baseline; width: 430px"
83 text="Recuerda que debes escribir en todos los campos que
contengan (*), a excepcin de la nueva password si quieres
conservar la misma. Si quieres cambiar la password recuerda
que la longitud mnima son 4 caracteres." visible="false"/>
84 <webuijsf:label binding="#{PaginaUsuario.label19}" id="label19"
85 style="border: 1px solid navy; background-color: rgb(204, 204,
255); color: navy; font-family: 'Verdana','Arial','Helvetica
',sans-serif; font-size: 12px; font-style: oblique; font-
weight: bold; height: 20px; left: 672px; top: 216px;
position: absolute; text-align: center; vertical-align:
baseline; width: 286px"
86 text="Los cambios se han realizado con xito" visible="false"/>
87 <webuijsf:label binding="#{PaginaUsuario.label18}" id="label18"
88 style="border-width: 1px; border-style: solid; border-color: rgb
(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
font-family: 'Verdana','Arial','Helvetica',sans-serif; font
-size: 12px; font-style: oblique; font-weight: bold; height:
46px; left: 672px; top: 264px; position: absolute; text-
align: center; vertical-align: baseline; width: 286px"
89 text="La clave introducida no coincide con la de esta cuenta"
visible="false"/>
90 <webuijsf:image height="209" id="image1" style="left: 672px; top: 360px;
position: absolute" url="/resources/mundo.png" width="335"/>
91 </webuijsf:form>
92 </webuijsf:body>
93 </webuijsf:html>
94 </webuijsf:page>
95 </f:view>
96 </jsp:root>

```

PaginaAdmin.java

```

1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 package wiki;
7
8 import com.sun.rave.web.ui.appbase.AbstractPageBean;

```

```

9  import com.sun.webui.jsf.component.Form;
10 import com.sun.webui.jsf.component.Hyperlink;
11 import com.sun.webui.jsf.component.Label;
12 import com.sun.webui.jsf.component.PasswordField;
13 import javax.faces.FacesException;
14
15 /**
16  * <p>Page bean that corresponds to a similarly named JSP page. This
17  * class contains component definitions (and initialization code) for
18  * all components that you have defined on this page, as well as
19  * lifecycle methods and event handlers where you may add behavior
20  * to respond to incoming events.</p>
21  *
22  * @version PaginaAdmin.java
23  * @version Created on 10-may-2009, 13:09:56
24  * @author ZoZe
25  */
26
27 public class PaginaAdmin extends AbstractPageBean {
28     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
29
30     /**
31      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
32      * This method is automatically generated, so any user-specified code inserted
33      * here is subject to being replaced.</p>
34      */
35     private void _init() throws Exception {
36     }
37
38     // </editor-fold>
39     private UsuarioAyuda ayuda=new UsuarioAyuda();
40
41
42     private PasswordField passwordField1 = new PasswordField();
43
44     public PasswordField getPasswordField1() {
45         return passwordField1;
46     }
47
48     public void setPasswordField1(PasswordField pf) {
49         this.passwordField1 = pf;
50     }
51     private PasswordField passwordField2 = new PasswordField();
52
53     public PasswordField getPasswordField2() {
54         return passwordField2;
55     }
56
57     public void setPasswordField2(PasswordField pf) {
58         this.passwordField2 = pf;
59     }
60     private PasswordField passwordField3 = new PasswordField();
61
62     public PasswordField getPasswordField3() {
63         return passwordField3;
64     }
65
66     public void setPasswordField3(PasswordField pf) {
67         this.passwordField3 = pf;
68     }
69     private Form form2 = new Form();
70
71     public Form getForm2() {
72         return form2;
73     }
74
75     public void setForm2(Form f) {

```

```

76         this.form2 = f;
77     }
78     private Label label17 = new Label();
79
80     public Label getLabel17() {
81         return label17;
82     }
83
84     public void setLabel17(Label l) {
85         this.label17 = l;
86     }
87     private Hyperlink hyperlink8 = new Hyperlink();
88
89     public Hyperlink getHyperlink8() {
90         return hyperlink8;
91     }
92
93     public void setHyperlink8(Hyperlink h) {
94         this.hyperlink8 = h;
95     }
96     private Hyperlink hyperlink4 = new Hyperlink();
97
98     public Hyperlink getHyperlink4() {
99         return hyperlink4;
100    }
101
102    public void setHyperlink4(Hyperlink h) {
103        this.hyperlink4 = h;
104    }
105    private Hyperlink hyperlink3 = new Hyperlink();
106
107    public Hyperlink getHyperlink3() {
108        return hyperlink3;
109    }
110
111    public void setHyperlink3(Hyperlink h) {
112        this.hyperlink3 = h;
113    }
114    private Hyperlink hyperlink1 = new Hyperlink();
115
116    public Hyperlink getHyperlink1() {
117        return hyperlink1;
118    }
119
120    public void setHyperlink1(Hyperlink h) {
121        this.hyperlink1 = h;
122    }
123    private Label label18 = new Label();
124
125    public Label getLabel18() {
126        return label18;
127    }
128
129    public void setLabel18(Label l) {
130        this.label18 = l;
131    }
132    private Label label19 = new Label();
133
134    public Label getLabel19() {
135        return label19;
136    }
137
138    public void setLabel19(Label l) {
139        this.label19 = l;
140    }
141    /**
142    * <p>Construct a new Page bean instance.</p>

```



```

143     */
144     public PaginaAdmin() {
145     }
146
147     /**
148      * <p>Callback method that is called whenever a page is navigated to,
149      * either directly via a URL, or indirectly via page navigation.
150      * Customize this method to acquire resources that will be needed
151      * for event handlers and lifecycle methods, whether or not this
152      * page is performing post back processing.</p>
153      *
154      * <p>Note that, if the current request is a postback, the property
155      * values of the components do <strong>not</strong> represent any
156      * values submitted with this request. Instead, they represent the
157      * property values that were saved for this view when it was rendered.</p>
158      */
159     @Override
160     public void init() {
161         // Perform initializations inherited from our superclass
162         super.init();
163         // Perform application initialization that must complete
164         // *before* managed components are initialized
165         // TODO – add your own initialization code here
166
167         // <editor–fold defaultstate="collapsed" desc="Managed Component Initialization">
168         // Initialize automatically managed components
169         // *Note* – this logic should NOT be modified
170         try {
171             _init();
172         } catch (Exception e) {
173             log("PaginaAdmin Initialization Failure", e);
174             throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
175         }
176
177         // </editor–fold>
178         // Perform application initialization that must complete
179         // *after* managed components are initialized
180         // TODO – add your own initialization code here
181     }
182
183     /**
184      * <p>Callback method that is called after the component tree has been
185      * restored, but before any event processing takes place. This method
186      * will <strong>only</strong> be called on a postback request that
187      * is processing a form submit. Customize this method to allocate
188      * resources that will be required in your event handlers.</p>
189      */
190     @Override
191     public void preprocess() {
192     }
193
194     /**
195      * <p>Callback method that is called just before rendering takes place.
196      * This method will <strong>only</strong> be called for the page that
197      * will actually be rendered (and not, for example, on a page that
198      * handled a postback and then navigated to a different page). Customize
199      * this method to allocate resources that will be required for rendering
200      * this page.</p>
201      */
202     @Override
203     public void prerender() {
204     }
205
206     /**
207      * <p>Callback method that is called after rendering is completed for
208      * this request, if <code>init()</code> was called (regardless of whether
209      * or not this was the page that was actually rendered). Customize this

```

```

210  * method to release resources acquired in the <code>init()</code>,
211  * <code>preprocess()</code>, or <code>prerender()</code> methods (or
212  * acquired during execution of an event handler).</p>
213  */
214  @Override
215  public void destroy() {
216  }
217
218  /**
219   * <p>Return a reference to the scoped data bean.</p>
220   *
221   * @return reference to the scoped data bean
222   */
223  protected ApplicationBean1 getApplicationBean1() {
224      return (ApplicationBean1) getBean("ApplicationBean1");
225  }
226
227  /**
228   * <p>Return a reference to the scoped data bean.</p>
229   *
230   * @return reference to the scoped data bean
231   */
232  protected SessionBean1 getSessionBean1() {
233      return (SessionBean1) getBean("SessionBean1");
234  }
235
236  /**
237   * <p>Return a reference to the scoped data bean.</p>
238   *
239   * @return reference to the scoped data bean
240   */
241  protected RequestBean1 getRequestBean1() {
242      return (RequestBean1) getBean("RequestBean1");
243  }
244
245  public String hyperlink8_action() {
246      // TODO: Process the action. Return value is a navigation
247      // case name where null will return to the same page.
248
249      return "listado";
250  }
251
252  public String hyperlink7_action() {
253      // TODO: Process the action. Return value is a navigation
254      // case name where null will return to the same page.
255      String nombre=this.getSessionBean1().getNombreUsr();
256      String log=this.getSessionBean1().getLogin();
257      String pass=this.getSessionBean1().getPassword();
258      String apell=this.getSessionBean1().getApellidosUsr();
259      String correo=this.getSessionBean1().getEmailUsr();
260      String pais=this.getSessionBean1().getPaisUsr();
261      String ciudad=this.getSessionBean1().getCiudadUsr();
262      String telefono=this.getSessionBean1().getTelefonoUsr();
263      String passaux1 = (String) passwordField2.getValue();
264      String passaux2 = (String) passwordField3.getValue();
265      String passActual = (String) passwordField1.getValue();
266
267      if(pass == null || passaux1 == null || passaux2 == null || nombre.equals("") || apell.equals("") ||
268         correo.equals("") || passaux1.length() < 4)
269      {
270          label17.setVisible(true);
271          label18.setVisible(false);
272          label19.setVisible(false);
273      }
274      else if(!passActual.equals(pass))
275      {

```

```

276         label17.setVisible(false);
277         label18.setVisible(true);
278         label19.setVisible(false);
279     }
280     else
281     {
282         Usuarios modUsuario=new Usuarios(log.pass,"AD", nombre, apell, correo);
283         modUsuario.setPaisUsr(pais);
284         modUsuario.setCiudadUsr(ciudad);
285         modUsuario.setTelefonoUsr(telefono);
286
287         if(passaux1.equals(passaux2))
288         {
289             modUsuario.setPassword(passaux1);
290         }
291         ayuda.modificarUsuario(modUsuario);
292         label19.setVisible(true);
293         label17.setVisible(false);
294         label18.setVisible(false);
295     }
296
297     return null;
298 }
299
300 public String hyperlink1_action() {
301     // TODO: Process the action. Return value is a navigation
302     // case name where null will return to the same page.
303     return null;
304 }
305
306 public String hyperlink2_action() {
307     // TODO: Process the action. Return value is a navigation
308     // case name where null will return to the same page.
309     return "GoToAdministrarP";
310 }
311
312 public String hyperlink3_action() {
313     // TODO: Process the action. Return value is a navigation
314     // case name where null will return to the same page.
315     return "GoToListarP";
316 }
317
318 public String hyperlink4_action() {
319     // TODO: Process the action. Return value is a navigation
320     // case name where null will return to the same page.
321     return "GoToCrearP";
322 }
323
324 }

```

PaginaAdmin.jsp

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : PaginaAdmin
4     Created on : 10-may-2009, 13:09:55
5     Author : ZoZe
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
   html" xmlns:jsp="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
   webuijsf">
8     <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9     <f:view>
10         <webuijsf:page id="page1">

```

```

11      <webuijsf:html id="html1">
12          <webuijsf:head id="head1">
13              <webuijsf:link id="link1" url="/resources/stylessheet.css"/>
14          </webuijsf:head>
15          <webuijsf:body id="body1" style="-rave-layout: grid">
16              <webuijsf:form id="form1">
17                  <webuijsf:label id="label1"
18                      style="font-family: 'Times New Roman','Times',serif; font-size: 36
19                          px; font-style: oblique; font-weight: lighter; left: 24px;
20                          top: 24px; position: absolute; text-decoration: underline;
21                          width: 408px" text="Wiki 4Proyect - Andaluca"/>
22                  <webuijsf:label id="label2"
23                      style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
24                          -size: 12px; font-style: oblique; font-weight: lighter;
25                          left: 288px; top: 96px; position: absolute; text-decoration:
26                          underline; width: 238px" text="Bienvenido administrador de la
27                          wiki"/>
28                  <webuijsf:label id="label3"
29                      style="background-color: rgb(255, 255, 255); height: 22px; left:
30                          696px; top: 72px; position: absolute; width: 166px" text="Flix
31                          Reyes Fernndez"/>
32                  <webuijsf:label id="label4"
33                      style="background-color: rgb(204, 204, 204); height: 22px; left:
34                          696px; top: 48px; position: absolute; width: 166px" text="
35                          Sergio Puerta Aragn"/>
36                  <webuijsf:label id="label5"
37                      style="background-color: rgb(153, 153, 153); height: 22px; left:
38                          696px; top: 24px; position: absolute; width: 166px" text="
39                          Lorena Gutirrez Madroal"/>
40                  <webuijsf:label id="label6"
41                      style="background-color: rgb(102, 102, 102); height: 22px; left:
42                          696px; top: 0px; position: absolute; width: 166px" text="Jos
43                          Alberto Cordero Ros"/>
44                  <webuijsf:label id="label14"
45                      style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
46                          -size: 12px; font-style: italic; font-weight: lighter;
47                          height: 24px; left: 336px; top: 360px; position: absolute;
48                          width: 142px" text="Password actual(*)"/>
49                  <webuijsf:label id="label15"
50                      style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
51                          -size: 12px; font-style: italic; font-weight: lighter;
52                          height: 24px; left: 336px; top: 408px; position: absolute;
53                          width: 142px" text="Password nueva(*)"/>
54                  <webuijsf:label id="label16"
55                      style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
56                          -size: 12px; font-style: italic; font-weight: lighter;
57                          height: 24px; left: 336px; top: 456px; position: absolute;
58                          width: 166px" text="Repite password nueva(*)"/>
59                  <webuijsf:image height="96" id="image1" style="left: 600px; top: 96px;
60                      position: absolute"
61                      url="http://upload.wikimedia.org/wikipedia/commons/thumb/f/f7/
62                          P_wiki_letter_w.svg/400px-P_wiki_letter_w.svg.png" width="120
63                          "/>
64                  <webuijsf:image height="207" id="image2" style="left: 720px; top: 504px;
65                      position: absolute" url="/resources/mundo.png" width="335"/>
66              </webuijsf:form>
67              <webuijsf:form binding="#{PaginaAdmin.form2}" id="form2" style="border-top:
68                  solid gray 2px; height: 502px; left: 48px; top: 144px; position:
69                  absolute; width: 840px">
70                  <webuijsf:hyperlink actionExpression="#{PaginaAdmin.hyperlink1_action}"
71                      binding="#{PaginaAdmin.hyperlink1}" id="hyperlink1"
72                      style="background-color: rgb(102, 102, 102); color: white; font-
73                          family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
74                          font-style: oblique; font-weight: bold; height: 22px; left:
75                          24px; top: 24px; position: absolute; text-decoration: none;
76                          width: 166px"
77                      tabIndex="11" text="Ir a Bienvenida" url="/faces/index.jsp"/>

```

```

43 <webuijsf:hyperlink actionExpression="#{PaginaAdmin.hyperlink3_action}"
    binding="#{PaginaAdmin.hyperlink3}" id="hyperlink3"
44 style="background-color: rgb(204, 204, 204); color: black; font-
    family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
    font-style: oblique; font-weight: bold; height: 24px; left:
    24px; top: 72px; position: absolute; text-decoration: none;
    width: 168px"
45 tabIndex="13" text="Listar pginas"/>
46 <webuijsf:hyperlink actionExpression="#{PaginaAdmin.hyperlink4_action}"
    binding="#{PaginaAdmin.hyperlink4}" id="hyperlink4"
47 style="background-color: rgb(153, 153, 153); color: black; font-
    family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
    font-style: oblique; font-weight: bold; height: 24px; left:
    24px; top: 96px; position: absolute; text-decoration: none;
    width: 168px"
48 tabIndex="14" text="Crear pgina"/>
49 <webuijsf:label id="label7"
50 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
    -size: 12px; font-style: italic; font-weight: lighter;
    height: 24px; left: 288px; top: 120px; position: absolute;
    width: 96px" text="Nombre (*)"/>
51 <webuijsf:textField columns="30" id="textField1" maxLength="30" style="left:
    456px; top: 120px; position: absolute; z-index: 500"
52 tabIndex="1" text="#{SessionBean1.nombreUsr}"/>
53 <webuijsf:textField columns="30" id="textField2" maxLength="50" style="left:
    456px; top: 168px; position: absolute" tabIndex="2" text="#{
    SessionBean1.apellidosUsr}"/>
54 <webuijsf:label id="label8"
55 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
    -size: 12px; font-style: italic; font-weight: lighter;
    height: 24px; left: 288px; top: 168px; position: absolute;
    width: 96px" text="Apellidos (*)"/>
56 <webuijsf:label id="label9"
57 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
    -size: 12px; font-style: italic; font-weight: lighter;
    height: 24px; left: 288px; top: 360px; position: absolute;
    width: 142px" text="Correo electrónico (*)"/>
58 <webuijsf:textField columns="30" id="textField3" maxLength="30" style="left:
    456px; top: 360px; position: absolute" tabIndex="6" text="#{
    SessionBean1.emailUsr}"/>
59 <webuijsf:hyperlink actionExpression="#{PaginaAdmin.hyperlink7_action}" id="
    hyperlink7"
60 style="color: gray; font-family: 'Verdana','Arial','Helvetica',
    sans-serif; font-size: 12px; font-style: oblique; font-
    weight: bold; left: 720px; top: 120px; position: absolute;
    width: 118px"
61 tabIndex="10" text="Guardar cambios"/>
62 <webuijsf:label id="label10"
63 style="color: black; font-family: 'Verdana','Arial','Helvetica',
    sans-serif; font-size: 12px; font-style: oblique; font-
    weight: lighter; height: 46px; left: 240px; top: 48px;
    position: absolute; width: 502px" text="Si vas a modificar tus
    datos, recuerda que los que tienen el asterisco son campos
    obligatorios. No olvides darle a "Guardar cambios"
    cuando hayas finalizado!"/>
64 <webuijsf:textField columns="30" id="textField4" maxLength="20" style="left:
    456px; top: 408px; position: absolute" tabIndex="7" text="#{
    SessionBean1.paisUsr}"/>
65 <webuijsf:label id="label11"
66 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
    -size: 12px; font-style: italic; font-weight: lighter;
    height: 24px; left: 288px; top: 408px; position: absolute;
    width: 96px" text="Pas"/>
67 <webuijsf:label id="label12"
68 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
    -size: 12px; font-style: italic; font-weight: lighter;
    height: 24px; left: 288px; top: 456px; position: absolute;

```

```

        width: 96px" text="Ciudad"/>
69 <webuijsf:textField columns="30" id="textField5" maxLength="20" style="left:
    456px; top: 456px; position: absolute" tabIndex="8" text="#{
    SessionBean1.ciudadUsr}"/>
70 <webuijsf:textField columns="30" id="textField6" maxLength="9" style="left:
    456px; top: 504px; position: absolute" tabIndex="9" text="#{
    SessionBean1.telefonoUsr}"/>
71 <webuijsf:label id="label13"
72     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: italic; font-weight: lighter;
        height: 24px; left: 288px; top: 504px; position: absolute;
        width: 96px" text="Telefono"/>
73 <webuijsf:hyperlink actionExpression="#{PaginaAdmin.hyperlink8_action}"
    binding="#{PaginaAdmin.hyperlink8}" id="hyperlink8"
74     style="background-color: rgb(255, 255, 255); color: black; font-
        family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
        font-style: oblique; font-weight: bold; height: 24px; left:
        24px; top: 48px; position: absolute; text-decoration: none;
        width: 168px"
75     tabIndex="17" text="Listar usuarios" url="/faces/ListadoUsuarios.jsp"
    />
76 <webuijsf:passwordField binding="#{PaginaAdmin.passwordField1}" columns="30
    " id="passwordField1" maxLength="20"
77     style="left: 456px; top: 216px; position: absolute" tabIndex="3"/>
78 <webuijsf:passwordField binding="#{PaginaAdmin.passwordField2}" columns="30
    " id="passwordField2" maxLength="20"
79     style="left: 456px; top: 264px; position: absolute" tabIndex="4"/>
80 <webuijsf:passwordField binding="#{PaginaAdmin.passwordField3}" columns="30
    " id="passwordField3" maxLength="20"
81     style="left: 456px; top: 312px; position: absolute" tabIndex="5"/>
82 <webuijsf:label binding="#{PaginaAdmin.label17}" id="label17"
83     style="border-width: 1px; border-style: solid; border-color: rgb
        (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
        background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
        font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: oblique; font-weight: bold; height:
        68px; left: 624px; top: 240px; position: absolute; text-
        align: center; vertical-align: baseline; width: 430px"
84     text="Recuerda que tienes que completar todos los campos, si los
        campos de clave aparecen en amarillo es que no cumplen los
        requisitos de longitud o son diferentes." visible="false"/>
85 <webuijsf:label binding="#{PaginaAdmin.label18}" id="label18"
86     style="border-width: 1px; border-style: solid; border-color: rgb
        (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
        background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
        font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: oblique; font-weight: bold; height:
        44px; left: 600px; top: 240px; position: absolute; text-
        align: center; vertical-align: baseline"
87     text="La clave introducida no coincide con la de esta cuenta"
        visible="false"/>
88 <webuijsf:label binding="#{PaginaAdmin.label19}" id="label19"
89     style="border: 1px solid navy; background-color: rgb(204, 204,
        255); color: navy; font-family: 'Verdana','Arial','Helvetica
        ',sans-serif; font-size: 12px; font-style: oblique; font-
        weight: bold; height: 44px; left: 600px; top: 240px;
        position: absolute; text-align: center; vertical-align:
        baseline; width: 214px"
90     text="Los cambios se han realizado con xito" visible="false"/>
91 </webuijsf:form>
92 </webuijsf:body>
93 </webuijsf:html>
94 </webuijsf:page>
95 </f:view>
96 </jsp:root>

```

RegistroUsuario.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8  import com.sun.rave.web.ui.appbase.AbstractPageBean;
9  import com.sun.webui.jsf.component.Label;
10 import com.sun.webui.jsf.component.PasswordField;
11 import com.sun.webui.jsf.component.TextField;
12 import javax.faces.FacesException;
13
14 /**
15  * <p>Page bean that corresponds to a similarly named JSP page. This
16  * class contains component definitions (and initialization code) for
17  * all components that you have defined on this page, as well as
18  * lifecycle methods and event handlers where you may add behavior
19  * to respond to incoming events.</p>
20  *
21  * @version RegistroUsuario.java
22  * @version Created on 10-may-2009, 13:15:50
23  * @author ZoZe
24  */
25
26 public class RegistroUsuario extends AbstractPageBean {
27     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
28
29     /**
30      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
31      * This method is automatically generated, so any user-specified code inserted
32      * here is subject to being replaced.</p>
33      */
34     private void _init() throws Exception {
35     }
36     private TextField textField8 = new TextField();
37
38     public TextField getTextField8() {
39         return textField8;
40     }
41
42     public void setTextField8(TextField tf) {
43         this.textField8 = tf;
44     }
45     private TextField textField1 = new TextField();
46
47     public TextField getTextField1() {
48         return textField1;
49     }
50
51     public void setTextField1(TextField tf) {
52         this.textField1 = tf;
53     }
54     private TextField textField2 = new TextField();
55
56     public TextField getTextField2() {
57         return textField2;
58     }
59
60     public void setTextField2(TextField tf) {
61         this.textField2 = tf;
62     }
63     private TextField textField7 = new TextField();
64

```

```
65     public TextField getTextField7() {
66         return textField7;
67     }
68
69     public void setTextField7(TextField tf) {
70         this.textField7 = tf;
71     }
72     private TextField textField4 = new TextField();
73
74     public TextField getTextField4() {
75         return textField4;
76     }
77
78     public void setTextField4(TextField tf) {
79         this.textField4 = tf;
80     }
81     private TextField textField5 = new TextField();
82
83     public TextField getTextField5() {
84         return textField5;
85     }
86
87     public void setTextField5(TextField tf) {
88         this.textField5 = tf;
89     }
90     private TextField textField6 = new TextField();
91
92     public TextField getTextField6() {
93         return textField6;
94     }
95
96     public void setTextField6(TextField tf) {
97         this.textField6 = tf;
98     }
99     private PasswordField passwordField1 = new PasswordField();
100
101     public PasswordField getPasswordField1() {
102         return passwordField1;
103     }
104
105     public void setPasswordField1(PasswordField pf) {
106         this.passwordField1 = pf;
107     }
108     private PasswordField passwordField2 = new PasswordField();
109
110     public PasswordField getPasswordField2() {
111         return passwordField2;
112     }
113
114     public void setPasswordField2(PasswordField pf) {
115         this.passwordField2 = pf;
116     }
117
118     private Label label20 = new Label();
119
120     public Label getLabel20() {
121         return label20;
122     }
123
124     public void setLabel20(Label l) {
125         this.label20 = l;
126     }
127     private Label label21 = new Label();
128
129     public Label getLabel21() {
130         return label21;
131     }
```



```

132
133     public void setLabel21(Label l) {
134         this.label21 = l;
135     }
136     private Label label22 = new Label();
137
138     public Label getLabel22() {
139         return label22;
140     }
141
142     public void setLabel22(Label l) {
143         this.label22 = l;
144     }
145
146     // </editor-fold>
147     private UsuarioAyuda ayuda=new UsuarioAyuda();
148     private Label label23 = new Label();
149
150     public Label getLabel23() {
151         return label23;
152     }
153
154     public void setLabel23(Label l) {
155         this.label23 = l;
156     }
157     private Label label24 = new Label();
158
159     public Label getLabel24() {
160         return label24;
161     }
162
163     public void setLabel24(Label l) {
164         this.label24 = l;
165     }
166     /**
167      * <p>Construct a new Page bean instance.</p>
168      */
169     public RegistroUsuario() {
170
171     }
172
173     /**
174      * <p>Callback method that is called whenever a page is navigated to,
175      * either directly via a URL, or indirectly via page navigation.
176      * Customize this method to acquire resources that will be needed
177      * for event handlers and lifecycle methods, whether or not this
178      * page is performing post back processing.</p>
179      *
180      * <p>Note that, if the current request is a postback, the property
181      * values of the components do <strong>not</strong> represent any
182      * values submitted with this request. Instead, they represent the
183      * property values that were saved for this view when it was rendered.</p>
184      */
185     @Override
186     public void init() {
187         // Perform initializations inherited from our superclass
188         super.init();
189
190         // Perform application initialization that must complete
191         // *before* managed components are initialized
192         // TODO – add your own initialization code here
193
194         // <editor-fold defaultstate="collapsed" desc="Managed Component Initialization">
195         // Initialize automatically managed components
196         // *Note* – this logic should NOT be modified
197         try {
198             _init();

```

```

199         } catch (Exception e) {
200             log("RegistroUsuario Initialization Failure", e);
201             throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
202         }
203     }
204
205     // </editor-fold>
206     // Perform application initialization that must complete
207     // *after* managed components are initialized
208     // TODO – add your own initialization code here
209 }
210
211 /**
212  * <p>Callback method that is called after the component tree has been
213  * restored, but before any event processing takes place. This method
214  * will <strong>only</strong> be called on a postback request that
215  * is processing a form submit. Customize this method to allocate
216  * resources that will be required in your event handlers.</p>
217  */
218 @Override
219 public void preprocess() {
220 }
221
222 /**
223  * <p>Callback method that is called just before rendering takes place.
224  * This method will <strong>only</strong> be called for the page that
225  * will actually be rendered (and not, for example, on a page that
226  * handled a postback and then navigated to a different page). Customize
227  * this method to allocate resources that will be required for rendering
228  * this page.</p>
229  */
230 @Override
231 public void prerender() {
232 }
233
234 /**
235  * <p>Callback method that is called after rendering is completed for
236  * this request, if <code>init()</code> was called (regardless of whether
237  * or not this was the page that was actually rendered). Customize this
238  * method to release resources acquired in the <code>init()</code>,
239  * <code>preprocess()</code>, or <code>prerender()</code> methods (or
240  * acquired during execution of an event handler).</p>
241  */
242 @Override
243 public void destroy() {
244 }
245
246 /**
247  * <p>Return a reference to the scoped data bean.</p>
248  *
249  * @return reference to the scoped data bean
250  */
251 protected ApplicationBean1 getApplicationBean1() {
252     return (ApplicationBean1) getBean("ApplicationBean1");
253 }
254
255 /**
256  * <p>Return a reference to the scoped data bean.</p>
257  *
258  * @return reference to the scoped data bean
259  */
260 protected SessionBean1 getSessionBean1() {
261     return (SessionBean1) getBean("SessionBean1");
262 }
263
264 /**
265  * <p>Return a reference to the scoped data bean.</p>

```

```

266      *
267      * @return reference to the scoped data bean
268      */
269      protected RequestBean1 getRequestBean1() {
270          return (RequestBean1) getBean("RequestBean1");
271      }
272
273      public String hyperlink7_action() {
274          // TODO: Process the action. Return value is a navigation
275          // case name where null will return to the same page.
276          textField1.setText("");
277          textField2.setText("");
278          textField4.setText("");
279          textField5.setText("");
280          textField6.setText("");
281          textField7.setText("");
282          textField8.setText("");
283          passwordField1.setText("");
284          passwordField2.setText("");
285
286          return null;
287      }
288
289      public String hyperlink2_action() {
290          // TODO: Process the action. Return value is a navigation
291          // case name where null will return to the same page.
292          return null;
293      }
294
295      public String hyperlink6_action()
296      {
297          // Tendramos que realizar comprobaciones de que el login no existe,
298          // las claves son las mismas, los datos son introducidos correctamente
299
300
301          String pass=(String)passwordField1.getText();
302          String pass2=(String)passwordField2.getText();
303          String log=(String)textField8.getText();
304          String nom=(String)textField1.getText();
305          String apell=(String)textField2.getText();
306          String correo=(String)textField7.getText();
307          String pais=(String)textField4.getText();
308          String ciudad=(String)textField5.getText();
309          String telefono=(String)textField6.getText();
310
311
312
313
314
315
316
317          if(pass == null || pass2 == null || log == null || nom == null || apell == null || correo == null)
318          {
319              textField8.setText("");
320              passwordField1.setText("");
321              passwordField2.setText("");
322              label20.setVisible(false);
323              label21.setVisible(false);
324              label22.setVisible(false);
325              label23.setVisible(false);
326              label24.setVisible(true);
327          }
328          else if(pass.isEmpty() || pass2.isEmpty() || log.isEmpty() || nom.isEmpty() || apell.isEmpty() || correo.
329              isEmpty())
330          {
331              textField8.setText("");
332              passwordField1.setText("");

```

```

332     passwordField2.setText("");
333     label20.setVisible(false);
334     label21.setVisible(false);
335     label22.setVisible(false);
336     label23.setVisible(false);
337     label24.setVisible(true);
338 }
339 else
340 {
341     UsuarioAyuda a=new UsuarioAyuda();
342     Usuarios u = a.buscarUsuario(log);
343
344     if(u != null)
345     {
346
347         textField8.setText("");
348         passwordField1.setText("");
349         passwordField2.setText("");
350         label20.setVisible(false);
351         label21.setVisible(true);
352         label22.setVisible(false);
353         label23.setVisible(false);
354         label24.setVisible(false);
355
356     }
357     else if(!pass.equals(pass2))
358     {
359         passwordField1.setText("");
360         passwordField2.setText("");
361         label20.setVisible(true);
362         label21.setVisible(false);
363         label22.setVisible(false);
364         label23.setVisible(false);
365         label24.setVisible(false);
366     }
367
368     else if(pass.length()< 4 || pass2.length()< 4)
369     {
370         passwordField1.setText("");
371         passwordField2.setText("");
372         label20.setVisible(false);
373         label21.setVisible(false);
374         label22.setVisible(true);
375         label23.setVisible(false);
376         label24.setVisible(false);
377     }
378     else
379     {
380         Usuarios nuevoUsuario=new Usuarios(log,pass,"UR", nom, apell, correo);
381         nuevoUsuario.setPaisUsr(pais);
382         nuevoUsuario.setCiudadUsr(ciudad);
383         nuevoUsuario.setTelefonoUsr(telefono);
384
385         ayuda.registrarUsuario(nuevoUsuario);
386         label20.setVisible(false);
387         label21.setVisible(false);
388         label22.setVisible(false);
389         label23.setVisible(true);
390         hyperlink7_action();
391         //Mostrar un mensaje de que se ha registrado sin problemas
392         // TODO: Replace with your code
393     }
394 }
395 return null;
396
397 }
398

```

399

}

RegistroUsuario.jsp

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : RegistroUsuario
4     Created on : 10-may-2009, 13:15:50
5     Author : ZoZe
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
  html" xmlns:jspx="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
  webuijsf">
8 <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9 <f:view>
10     <webuijsf:page id="page1">
11         <webuijsf:html id="html1">
12             <webuijsf:head id="head1">
13                 <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
14             </webuijsf:head>
15             <webuijsf:body id="body1" style="-rave-layout: grid">
16                 <webuijsf:form id="form1">
17                     <webuijsf:label id="label1"
18                         style="font-family: 'Times New Roman','Times', serif; font-size: 36
19                             px; font-style: oblique; font-weight: lighter; left: 24px;
20                             top: 24px; position: absolute; text-decoration: underline;
21                             width: 408px" text="Wiki 4Proyect - Andaluca"/>
22                     <webuijsf:label id="label2"
23                         style="background-color: rgb(255, 255, 255); height: 22px; left:
24                             696px; top: 72px; position: absolute; width: 166px" text="Flix
25                             Reyes Fernndez"/>
26                     <webuijsf:label id="label3"
27                         style="background-color: rgb(204, 204, 204); height: 22px; left:
28                             696px; top: 48px; position: absolute; width: 166px" text="
29                             Sergio Puerta Aragn"/>
30                     <webuijsf:label id="label4"
31                         style="background-color: rgb(153, 153, 153); height: 22px; left:
32                             696px; top: 24px; position: absolute; width: 166px" text="
33                             Lorena Gutirrez Madroal"/>
34                     <webuijsf:label id="label5"
35                         style="background-color: rgb(102, 102, 102); height: 22px; left:
36                             696px; top: 0px; position: absolute; width: 166px" text="Jos
37                             Alberto Cordero Ros"/>
38                 </webuijsf:form>
39                 <webuijsf:form id="form2" style="border-top: solid gray 2px; height: 526px;
40                     left: 48px; top: 144px; position: absolute; width: 840px">
41                     <webuijsf:image height="207" id="image1" style="left: 672px; top: 336px;
42                         position: absolute" url="/resources/mundo.png" width="332"/>
43                     <webuijsf:label id="label6"
44                         style="font-family: 'Verdana','Arial','Helvetica', sans-serif; font
45                             -size: 12px; font-style: italic; font-weight: lighter;
46                             height: 24px; left: 288px; top: 240px; position: absolute;
47                             width: 96px" text="Nombre (*)"/>
48                     <webuijsf:hyperlink actionExpression="#{RegistroUsuario.hyperlink6_action}"
49                         id="hyperlink6"
50                         style="color: gray; font-family: 'Verdana','Arial','Helvetica',
51                             sans-serif; font-size: 12px; font-style: oblique; font-
52                             weight: bold; height: 20px; left: 744px; top: 48px;
53                             position: absolute; width: 118px"
54                         tabIndex="10" text="Guardar datos"/>
55                     <webuijsf:textField binding="#{RegistroUsuario.textField1}" columns="30" id="
56                         textField1" maxLength="30"
57                         style="left: 456px; top: 240px; position: absolute; z-index: 500"
58                         tabIndex="4"/>

```

```

37 <webuijsf:label id="label13"
38     style="color: black; font-family: Verdana,Arial,Helvetica,sans-
        serif; font-size: 12px; font-style: oblique; font-weight:
        lighter; height: 22px; left: 240px; top: 24px; position:
        absolute; width: 432px" text="Rellene los campos para realizar
        su registro en la wiki 4Proyect - Andaluca. Los campos que
        tienen el asterisco (*) son obligatorios."/>
39 <webuijsf:textField binding="#{RegistroUsuario.textField2}" columns="30" id="
        textField2" maxLength="50"
40     style="left: 456px; top: 288px; position: absolute" tabIndex="5"/>
41 <webuijsf:label id="label7"
42     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: italic; font-weight: lighter;
        height: 24px; left: 288px; top: 288px; position: absolute;
        width: 96px" text="Apellidos (*)"/>
43 <webuijsf:label id="label9"
44     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: italic; font-weight: lighter;
        height: 24px; left: 288px; top: 384px; position: absolute;
        width: 96px" text="Pas"/>
45 <webuijsf:textField binding="#{RegistroUsuario.textField4}" columns="30" id="
        textField4" maxLength="20"
46     style="left: 456px; top: 384px; position: absolute" tabIndex="7"/>
47 <webuijsf:label id="label10"
48     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: italic; font-weight: lighter;
        height: 24px; left: 288px; top: 432px; position: absolute;
        width: 96px" text="Ciudad"/>
49 <webuijsf:textField binding="#{RegistroUsuario.textField5}" columns="30" id="
        textField5" maxLength="20"
50     style="left: 456px; top: 432px; position: absolute" tabIndex="8"/>
51 <webuijsf:label id="label11"
52     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: italic; font-weight: lighter;
        height: 24px; left: 288px; top: 480px; position: absolute;
        width: 96px" text="Telefono"/>
53 <webuijsf:textField binding="#{RegistroUsuario.textField6}" columns="30" id="
        textField6" maxLength="9"
54     style="left: 456px; top: 480px; position: absolute" tabIndex="9"/>
55 <webuijsf:label id="label12"
56     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: italic; font-weight: lighter;
        height: 24px; left: 288px; top: 336px; position: absolute;
        width: 142px" text="Correo electrnico (*)"/>
57 <webuijsf:textField binding="#{RegistroUsuario.textField7}" columns="30"
        disabled="true" id="textField7" maxLength="30"
58     style="left: 456px; top: 336px; position: absolute" tabIndex="6"/>
59 <webuijsf:label id="label14"
60     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: italic; font-weight: lighter;
        height: 24px; left: 288px; top: 96px; position: absolute;
        width: 96px" text="Login (*)"/>
61 <webuijsf:textField binding="#{RegistroUsuario.textField8}" columns="30" id="
        textField8" maxLength="20"
62     style="left: 456px; top: 96px; position: absolute; z-index: 500"
        tabIndex="1"/>
63 <webuijsf:hyperlink actionExpression="#{RegistroUsuario.hyperlink7_action}"
        id="hyperlink7"
64     style="color: gray; font-family: Verdana,Arial,Helvetica,sans-
        serif; font-size: 12px; font-style: oblique; font-weight:
        bold; height: 20px; left: 744px; top: 72px; position:
        absolute; width: 94px"
        tabIndex="11" text="Borrar datos"/>
65 <webuijsf:label id="label15"
66     style="background-color: rgb(204, 204, 204); height: 24px; left:
        24px; top: 72px; position: absolute; width: 166px" text=" "/>
67 <webuijsf:label id="label16"
68

```

```

69         style="background-color: rgb(153, 153, 153); color: rgb(153, 153,
           153); height: 24px; left: 24px; top: 96px; position:
           absolute; width: 166px" text="Label"/>
70 <webuijsf:hyperlink actionExpression="#{RegistroUsuario.hyperlink2_action}"
           id="hyperlink2"
71     style="background-color: rgb(102, 102, 102); color: rgb(255, 255,
           255); font-family: Verdana,Arial,Helvetica,sans-serif; font-
           size: 12px; font-style: oblique; font-weight: bold; height:
           22px; left: 24px; top: 24px; position: absolute; text-
           decoration: none; width: 166px"
72     tabIndex="12" text="Ir a Bienvenida" url="/faces/index.jsp"/>
73 <webuijsf:label id="label17"
74     style="background-color: rgb(102, 102, 102); color: rgb(102, 102,
           102); height: 20px; left: 24px; top: 120px; position:
           absolute; width: 166px" text="Label"/>
75 <webuijsf:label id="label18" style="background-color: #ffffff; height: 24px
           ; left: 24px; top: 48px; position: absolute; width: 166px" text="
           "/>
76 <webuijsf:label id="label18"
77     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: italic; font-weight: lighter;
           height: 24px; left: 288px; top: 144px; position: absolute;
           width: 96px" text="Password (*)"/>
78 <webuijsf:label id="label19"
79     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: italic; font-weight: lighter;
           height: 20px; left: 288px; top: 192px; position: absolute;
           width: 142px" text="Repita password (*)"/>
80 <webuijsf:passwordField binding="#{RegistroUsuario.passwordField1}" columns
           ="30" id="passwordField1" maxLength="20"
81     style="position: absolute; left: 456px; top: 144px" tabIndex="2"/>
82 <webuijsf:passwordField binding="#{RegistroUsuario.passwordField2}" columns
           ="30" id="passwordField2" maxLength="20"
83     style="left: 456px; top: 192px; position: absolute" tabIndex="3"/>
84 <webuijsf:label binding="#{RegistroUsuario.label20}" id="label20"
85     style="border-width: 1px; border-style: solid; border-color: rgb
           (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
           background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
           font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: oblique; font-weight: bold; height:
           44px; left: 672px; top: 144px; position: absolute; text-
           align: center; vertical-align: baseline; width: 286px"
86     text="Las contraseñas no coinciden, por favor introdzcaldas de
           nuevo" visible="false"/>
87 <webuijsf:label binding="#{RegistroUsuario.label21}" id="label21"
88     style="border: 1px solid rgb(153, 0, 0); background-color: rgb
           (255, 204, 102); color: rgb(153, 0, 0); font-family: Verdana
           ,Arial,Helvetica,sans-serif; font-size: 12px; font-style:
           oblique; font-weight: bold; height: 22px; left: 672px; top:
           144px; position: absolute; text-align: center; vertical-
           align: baseline; width: 432px"
89     text="El login introducido ya existe, por favor intente
           introducir otro." visible="false"/>
90 <webuijsf:label binding="#{RegistroUsuario.label22}" id="label22"
91     style="color: rgb(102, 102, 102); font-family: 'Verdana','Arial','
           Helvetica',sans-serif; font-size: 12px; font-style: oblique;
           font-weight: lighter; height: 22px; left: 672px; top: 144px
           ; position: absolute; width: 432px"
92     text="La longitud mnima para la clave son 4 dgitos, por favor
           cambie su clave." visible="false"/>
93 <webuijsf:label binding="#{RegistroUsuario.label23}" id="label23"
94     style="border: 1px solid navy; background-color: rgb(204, 204,
           255); color: navy; font-family: 'Verdana','Arial','Helvetica
           ',sans-serif; font-size: 12px; font-style: oblique; font-
           weight: bold; height: 44px; left: 288px; top: 528px;
           position: absolute; text-align: center; vertical-align:
           baseline; width: 432px"

```

```

95         text="Su registro ha sido llevado a cabo con xito, si desea
           entrar en el sistema vaya de nuevo a la pgina de bienvenida,
           gracias." visible="false"/>
96         <webuijsf:label binding="#{RegistroUsuario.label24}" id="label24"
97         style="border-width: 1px; border-style: solid; border-color: rgb
           (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
           background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
           font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: oblique; font-weight: bold; height:
           44px; left: 672px; top: 168px; position: absolute; text-
           align: center; vertical-align: baseline; width: 360px"
98         text="Debe rellenar todos los campos que contiene (*) para que el
           registro sea correcto." visible="false"/>
99         </webuijsf:form>
100        </webuijsf:body>
101        </webuijsf:html>
102        </webuijsf:page>
103    </f:view>
104 </jsp:root>

```

PaginaVisitante.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8  import com.sun.rave.web.ui.appbase.AbstractPageBean;
9  import com.sun.webui.jsf.component.Hyperlink;
10 import javax.faces.FacesException;
11
12 /**
13  * <p>Page bean that corresponds to a similarly named JSP page. This
14  * class contains component definitions (and initialization code) for
15  * all components that you have defined on this page, as well as
16  * lifecycle methods and event handlers where you may add behavior
17  * to respond to incoming events.</p>
18  *
19  * @version PginaVisitante.java
20  * @version Created on 14-may-2009, 23:56:59
21  * @author WIND MOTEBOOK
22  */
23
24 public class PginaVisitante extends AbstractPageBean {
25     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
26
27     /**
28      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
29      * This method is automatically generated, so any user-specified code inserted
30      * here is subject to being replaced.</p>
31      */
32     private void _init() throws Exception {
33     }
34     private Hyperlink hyperlink2 = new Hyperlink();
35
36     public Hyperlink getHyperlink2() {
37         return hyperlink2;
38     }
39
40     public void setHyperlink2(Hyperlink h) {
41         this.hyperlink2 = h;
42     }

```



```

43     private Hyperlink hyperlink1 = new Hyperlink();
44
45     public Hyperlink getHyperlink1() {
46         return hyperlink1;
47     }
48
49     public void setHyperlink1(Hyperlink h) {
50         this.hyperlink1 = h;
51     }
52     private Hyperlink hyperlink3 = new Hyperlink();
53
54     public Hyperlink getHyperlink3() {
55         return hyperlink3;
56     }
57
58     public void setHyperlink3(Hyperlink h) {
59         this.hyperlink3 = h;
60     }
61
62     // </editor-fold>
63
64     /**
65      * <p>Construct a new Page bean instance.</p>
66      */
67     public PginaVisitante() {
68     }
69
70     /**
71      * <p>Callback method that is called whenever a page is navigated to,
72      * either directly via a URL, or indirectly via page navigation.
73      * Customize this method to acquire resources that will be needed
74      * for event handlers and lifecycle methods, whether or not this
75      * page is performing post back processing.</p>
76      *
77      * <p>Note that, if the current request is a postback, the property
78      * values of the components do <strong>not</strong> represent any
79      * values submitted with this request. Instead, they represent the
80      * property values that were saved for this view when it was rendered.</p>
81      */
82     @Override
83     public void init() {
84         // Perform initializations inherited from our superclass
85         super.init();
86         // Perform application initialization that must complete
87         // *before* managed components are initialized
88         // TODO – add your own initialization code here
89
90         // <editor-fold defaultstate="collapsed" desc="Managed Component Initialization">
91         // Initialize automatically managed components
92         // *Note* – this logic should NOT be modified
93         try {
94             _init();
95         } catch (Exception e) {
96             log("PginaVisitante Initialization Failure", e);
97             throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
98         }
99
100        // </editor-fold>
101        // Perform application initialization that must complete
102        // *after* managed components are initialized
103        // TODO – add your own initialization code here
104    }
105
106    /**
107     * <p>Callback method that is called after the component tree has been
108     * restored, but before any event processing takes place. This method
109     * will <strong>only</strong> be called on a postback request that

```

```

110      * is processing a form submit. Customize this method to allocate
111      * resources that will be required in your event handlers.</p>
112      */
113      @Override
114      public void preprocess() {
115      }
116
117      /**
118      * <p>Callback method that is called just before rendering takes place.
119      * This method will <strong>only</strong> be called for the page that
120      * will actually be rendered (and not, for example, on a page that
121      * handled a postback and then navigated to a different page). Customize
122      * this method to allocate resources that will be required for rendering
123      * this page.</p>
124      */
125      @Override
126      public void prerender() {
127      }
128
129      /**
130      * <p>Callback method that is called after rendering is completed for
131      * this request, if <code>init()</code> was called (regardless of whether
132      * or not this was the page that was actually rendered). Customize this
133      * method to release resources acquired in the <code>init()</code>,
134      * <code>preprocess()</code>, or <code>prerender()</code> methods (or
135      * acquired during execution of an event handler).</p>
136      */
137      @Override
138      public void destroy() {
139      }
140
141      /**
142      * <p>Return a reference to the scoped data bean.</p>
143      *
144      * @return reference to the scoped data bean
145      */
146      protected SessionBean1 getSessionBean1() {
147          return (SessionBean1) getBean("SessionBean1");
148      }
149
150      /**
151      * <p>Return a reference to the scoped data bean.</p>
152      *
153      * @return reference to the scoped data bean
154      */
155      protected RequestBean1 getRequestBean1() {
156          return (RequestBean1) getBean("RequestBean1");
157      }
158
159      /**
160      * <p>Return a reference to the scoped data bean.</p>
161      *
162      * @return reference to the scoped data bean
163      */
164      protected ApplicationBean1 getApplicationBean1() {
165          return (ApplicationBean1) getBean("ApplicationBean1");
166      }
167
168      public String hyperlink2_action() {
169          // TODO: Process the action. Return value is a navigation
170          // case name where null will return to the same page.
171          return "GoToListarP";
172      }
173
174      public String hyperlink1_action() {
175          // TODO: Process the action. Return value is a navigation
176          // case name where null will return to the same page.

```

```

177     return "GoToIndex";
178 }
179
180 }

```

PaginaVisitante.jsp

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : PginaVisitante
4     Created on : 14-may-2009, 23:56:58
5     Author : WIND MOTEBOOK
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
  html" xmlns:jsf="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
  webuijsf">
8 <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9 <f:view>
10     <webuijsf:page id="page1">
11         <webuijsf:html id="html1">
12             <webuijsf:head id="head1">
13                 <webuijsf:link id="link1" url="/resources/stylessheet.css"/>
14             </webuijsf:head>
15             <webuijsf:body id="body1" style="-rave-layout: grid">
16                 <webuijsf:form id="form1">
17                     <webuijsf:label id="label1"
18                         style="font-family: 'Times New Roman','Times', serif; font-size: 36
19                             px; font-style: oblique; font-weight: lighter; left: 24px;
20                             top: 24px; position: absolute; text-decoration: underline;
21                             width: 408px" text="Wiki 4Proyect - Andaluca"/>
22                     <webuijsf:label id="label2"
23                         style="background-color: rgb(255, 255, 255); height: 22px; left:
24                             696px; top: 72px; position: absolute; width: 166px" text="Flix
25                             Reyes Fernndez"/>
26                     <webuijsf:label id="label3"
27                         style="background-color: rgb(204, 204, 204); height: 22px; left:
28                             696px; top: 48px; position: absolute; width: 166px" text="
29                             Sergio Puerta Aragn"/>
30                     <webuijsf:label id="label4"
31                         style="background-color: rgb(153, 153, 153); height: 22px; left:
32                             696px; top: 24px; position: absolute; width: 166px" text="
33                             Lorena Gutirrez Madroal"/>
34                     <webuijsf:label id="label5"
35                         style="background-color: rgb(102, 102, 102); height: 22px; left:
36                             696px; top: 0px; position: absolute; width: 166px" text="Jos
37                             Alberto Cordero Ros"/>
38                     <webuijsf:hyperlink actionExpression="#{PginaVisitante.hyperlink1_action}"
39                         binding="#{PginaVisitante.hyperlink1}" id="hyperlink1"
40                         style="background-color: rgb(102, 102, 102); color: #ffffff; font-
41                             family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
42                             font-style: oblique; font-weight: bold; height: 22px; left:
43                             72px; top: 168px; position: absolute; text-decoration: none
44                             ; width: 166px"
45                         tabIndex="11" text="Ir a Bienvenida" url="/faces/index.jsp"/>
46                     <webuijsf:hyperlink actionExpression="#{PginaVisitante.hyperlink2_action}"
47                         binding="#{PginaVisitante.hyperlink2}" id="hyperlink2"
48                         style="color: black; font-family: Verdana,Arial,Helvetica,sans-
49                             serif; font-size: 12px; font-style: oblique; font-weight:
50                             bold; height: 24px; left: 72px; top: 192px; position:
51                             absolute; text-decoration: none; width: 168px"
52                         tabIndex="12" text="Listar pginas"/>
53                     <webuijsf:label id="label6"
54                         style="color: rgb(102, 102, 102); font-family: 'Verdana','Arial','
55                             Helvetica',sans-serif; font-size: 18px; font-style: oblique;

```

```

font-weight: lighter; height: 139px; left: 456px; top: 240
px; position: absolute; width: 478px" text=" Como visitante,
le permitiremos visitar nuestra wiki navegando por alguna de
sus pginas. Si deseas contribuir en nuestra wiki Regstrate
!. Haz click en el enlace Quiero registrarme! y completa los
datos. Gracias por tu visita."/>
35 <webuijsf:label id="label7"
36 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
-size: 24px; font-style: oblique; font-weight: lighter;
left: 312px; top: 144px; position: absolute; width: 646px"
text="Bienvenido a nuestra wiki &quot;Wiki 4 Proyect-Andaluca
!"/>
37 <webuijsf:hyperlink actionExpression="#{PginaVisitante.hyperlink2_action}"
binding="#{PginaVisitante.hyperlink3}" id="hyperlink3"
38 style="color: rgb(102, 102, 102); font-family: 'Verdana','Arial','
Helvetica',sans-serif; font-size: 14px; font-weight: bold;
left: 600px; top: 408px; position: absolute"
39 tabIndex="4" text="Quiero registrarme!" url="/faces/RegistroUsuario.
jsp"/>
40 <webuijsf:image height="213" id="image1" style="left: 24px; top: 264px;
position: absolute"
41 url="http://www.seocharlie.com/blog/wp-content/uploads/2007/08/
wiki.gif" width="384"/>
42 </webuijsf:form>
43 </webuijsf:body>
44 </webuijsf:html>
45 </webuijsf:page>
46 </f:view>
47 </jsp:root>

```

Gestión de páginas

AdministracionPaginas.java

```

1 package wiki;
2 // Generated 26-may-2009 20:37:26 by Hibernate Tools 3.2.1.GA
3
4
5
6 /**
7  * AdministracionPaginas generated by hbm2java
8  */
9 public class AdministracionPaginas implements java.io.Serializable {
10
11
12     private AdministracionPaginasId id;
13
14     public AdministracionPaginas() {
15     }
16
17     public AdministracionPaginas(AdministracionPaginasId id) {
18         this.id = id;
19     }
20
21     public AdministracionPaginasId getId() {
22         return this.id;
23     }
24
25     public void setId(AdministracionPaginasId id) {
26         this.id = id;
27     }
28
29

```

30
31
32

```
}
```

AdministracionPaginasId.java

```
1 package wiki;
2 // Generated 26--may--2009 20:37:26 by Hibernate Tools 3.2.1.GA
3
4
5
6 /**
7  * AdministracionPaginasId generated by hbm2java
8  */
9 public class AdministracionPaginasId implements java.io.Serializable {
10
11
12     private String login;
13     private long idPagina;
14
15     public AdministracionPaginasId() {
16     }
17
18     public AdministracionPaginasId(String login, long idPagina) {
19         this.login = login;
20         this.idPagina = idPagina;
21     }
22
23     public String getLogin() {
24         return this.login;
25     }
26
27     public void setLogin(String login) {
28         this.login = login;
29     }
30     public long getIdPagina() {
31         return this.idPagina;
32     }
33
34     public void setIdPagina(long idPagina) {
35         this.idPagina = idPagina;
36     }
37
38
39     public boolean equals(Object other) {
40         if ( (this == other) ) return true;
41         if ( (other == null) ) return false;
42         if ( !(other instanceof AdministracionPaginasId) ) return false;
43         AdministracionPaginasId castOther = ( AdministracionPaginasId ) other;
44
45         return ( (this.getLogin()==castOther.getLogin()) || ( this.getLogin()!=null && castOther.getLogin()
46         && (this.getIdPagina()==castOther.getIdPagina()));
47     }
48
49     public int hashCode() {
50         int result = 17;
51
52         result = 37 * result + ( getLogin() == null ? 0 : this.getLogin().hashCode() );
53         result = 37 * result + (int) this.getIdPagina();
54         return result;
55     }
56
57
```

```
58 }
```

AdministrarPaginas.java

```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8  import com.sun.rave.web.ui.appbase.AbstractPageBean;
9  import com.sun.webui.jsf.component.Button;
10 import com.sun.webui.jsf.component.Form;
11 import com.sun.webui.jsf.component.Hyperlink;
12 import com.sun.webui.jsf.component.StaticText;
13 import javax.faces.FacesException;
14
15 /**
16  * <p>Page bean that corresponds to a similarly named JSP page. This
17  * class contains component definitions (and initialization code) for
18  * all components that you have defined on this page, as well as
19  * lifecycle methods and event handlers where you may add behavior
20  * to respond to incoming events.</p>
21  *
22  * @version AdministrarPginas.java
23  * @version Created on 15-may-2009, 1:20:50
24  * @author WIND MOTEBOOK
25  */
26
27 public class AdministrarPginas extends AbstractPageBean {
28     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
29
30     /**
31      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
32      * This method is automatically generated, so any user-specified code inserted
33      * here is subject to being replaced.</p>
34      */
35     private void _init() throws Exception {
36     }
37     private Form form2 = new Form();
38
39     public Form getForm2() {
40         return form2;
41     }
42
43     public void setForm2(Form f) {
44         this.form2 = f;
45     }
46     private Hyperlink hyperlink2 = new Hyperlink();
47
48     public Hyperlink getHyperlink2() {
49         return hyperlink2;
50     }
51
52     public void setHyperlink2(Hyperlink h) {
53         this.hyperlink2 = h;
54     }
55     private Hyperlink hyperlink4 = new Hyperlink();
56
57     public Hyperlink getHyperlink4() {
58         return hyperlink4;
59     }
60 }
```

```

61     public void setHyperlink4(Hyperlink h) {
62         this.hyperlink4 = h;
63     }
64     private Button button3 = new Button();
65
66     public Button getButton3() {
67         return button3;
68     }
69
70     public void setButton3(Button b) {
71         this.button3 = b;
72     }
73     private Button button2 = new Button();
74
75     public Button getButton2() {
76         return button2;
77     }
78
79     public void setButton2(Button b) {
80         this.button2 = b;
81     }
82     private Button button1 = new Button();
83
84     public Button getButton1() {
85         return button1;
86     }
87
88     public void setButton1(Button b) {
89         this.button1 = b;
90     }
91
92     // </editor-fold>
93     private PaginaAyuda ayudaP=new PaginaAyuda();
94     private StaticText staticText1 = new StaticText();
95
96     public StaticText getStaticText1() {
97         return staticText1;
98     }
99
100    public void setStaticText1(StaticText st) {
101        this.staticText1 = st;
102    }
103    private StaticText staticText2 = new StaticText();
104
105    public StaticText getStaticText2() {
106        return staticText2;
107    }
108
109    public void setStaticText2(StaticText st) {
110        this.staticText2 = st;
111    }
112
113
114
115    /**
116     * <p>Construct a new Page bean instance.</p>
117     */
118    public AdministrarPginas() {
119        boolean published=getSessionBean1().getPagina().isPublicada();
120        if(published)
121            button1.setText("Despublicar");
122        else
123            button1.setText("Publicar");
124
125
126    }
127

```

```

128  /**
129   * <p>Callback method that is called whenever a page is navigated to,
130   * either directly via a URL, or indirectly via page navigation.
131   * Customize this method to acquire resources that will be needed
132   * for event handlers and lifecycle methods, whether or not this
133   * page is performing post back processing.</p>
134   *
135   * <p>Note that, if the current request is a postback, the property
136   * values of the components do <strong>not</strong> represent any
137   * values submitted with this request. Instead, they represent the
138   * property values that were saved for this view when it was rendered.</p>
139   */
140  @Override
141  public void init() {
142      // Perform initializations inherited from our superclass
143      super.init();
144
145      // Perform application initialization that must complete
146      // *before* managed components are initialized
147      // TODO – add your own initialization code here
148
149      // <editor–fold defaultstate="collapsed" desc="Managed Component Initialization">
150      // Initialize automatically managed components
151      // *Note* – this logic should NOT be modified
152      try {
153          _init();
154      } catch (Exception e) {
155          log("AdministrarPginas Initialization Failure", e);
156          throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
157      }
158
159      // </editor–fold>
160      // Perform application initialization that must complete
161      // *after* managed components are initialized
162      // TODO – add your own initialization code here
163  }
164
165  /**
166   * <p>Callback method that is called after the component tree has been
167   * restored, but before any event processing takes place. This method
168   * will <strong>only</strong> be called on a postback request that
169   * is processing a form submit. Customize this method to allocate
170   * resources that will be required in your event handlers.</p>
171   */
172  @Override
173  public void preprocess() {
174  }
175
176  /**
177   * <p>Callback method that is called just before rendering takes place.
178   * This method will <strong>only</strong> be called for the page that
179   * will actually be rendered (and not, for example, on a page that
180   * handled a postback and then navigated to a different page). Customize
181   * this method to allocate resources that will be required for rendering
182   * this page.</p>
183   */
184  @Override
185  public void prerender() {
186
187  }
188
189  /**
190   * <p>Callback method that is called after rendering is completed for
191   * this request, if <code>init()</code> was called (regardless of whether
192   * or not this was the page that was actually rendered). Customize this
193   * method to release resources acquired in the <code>init()</code>,

```



```

195     * <code>preprocess()</code>, or <code>prerender()</code> methods (or
196     * acquired during execution of an event handler).</p>
197     */
198     @Override
199     public void destroy() {
200     }
201
202     /**
203     * <p>Return a reference to the scoped data bean.</p>
204     *
205     * @return reference to the scoped data bean
206     */
207     protected SessionBean1 getSessionBean1() {
208         return (SessionBean1) getBean("SessionBean1");
209     }
210
211     /**
212     * <p>Return a reference to the scoped data bean.</p>
213     *
214     * @return reference to the scoped data bean
215     */
216     protected RequestBean1 getRequestBean1() {
217         return (RequestBean1) getBean("RequestBean1");
218     }
219
220     /**
221     * <p>Return a reference to the scoped data bean.</p>
222     *
223     * @return reference to the scoped data bean
224     */
225     protected ApplicationBean1 getApplicationBean1() {
226         return (ApplicationBean1) getBean("ApplicationBean1");
227     }
228
229     public String hyperlink1_action() {
230         // TODO: Process the action. Return value is a navigation
231         // case name where null will return to the same page.
232         return "GoToIndex";
233     }
234
235     public String hyperlink2_action() {
236         // TODO: Process the action. Return value is a navigation
237         // case name where null will return to the same page.
238         if(getSessionBean1().getRolUusr().equals("AD"))
239             return "GoToPageAd";
240         else
241             return "GoToPageUR";
242     }
243
244     public String hyperlink3_action() {
245         // TODO: Process the action. Return value is a navigation
246         // case name where null will return to the same page.
247         return "GoToCrearP";
248     }
249
250     public String hyperlink4_action() {
251         // TODO: Process the action. Return value is a navigation
252         // case name where null will return to the same page.
253         return "GoToListarP";
254     }
255
256     public String button1_action() {
257         // TODO: Process the action. Return value is a navigation
258         // case name where null will return to the same page.
259         Paginas p=getSessionBean1().getPagina();
260         if(p.isPublicada()){
261             p.setPublicada(false);

```

```

262         button1.setText("Publicar");
263     }
264     else{
265         p.setPublicada(true);
266         button1.setText("Despublicar");
267     }
268     ayudaP.modificarPagina(p);
269     return null;
270 }
271
272 public String button2_action() {
273     // TODO: Process the action. Return value is a navigation
274     // case name where null will return to the same page.
275     return "GoToModificarPW";
276 }
277
278 public String button3_action() {
279     // TODO: Process the action. Return value is a navigation
280     // case name where null will return to the same page.
281     UsuarioAyuda ayuda=new UsuarioAyuda();
282     Usuarios useract=ayuda.buscarUsuario(getSessionBean1().getLogin());
283     getSessionBean1().setListaUPermisos(ayuda.recuperarUsuariosPermisos(useract.getSessionBean1().
284         getPagina()));
285     return "DarPerm";
286 }
287
288 public String button4_action() {
289     // TODO: Process the action. Return value is a navigation
290     // case name where null will return to the same page.
291     return null;
292 }
293
294 public String button5_action() {
295     // TODO: Process the action. Return value is a navigation
296     // case name where null will return to the same page.
297     return null;
298 }
299 }

```

AdministrarPaginas.jsp

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : AdministrarPginas
4     Created on : 15-may-2009, 1:20:50
5     Author : WIND MOTEBOOK
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
8     html" xmlns:jsf="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
9     webuijsf">
10     <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
11     <f:view>
12         <webuijsf:page id="page1">
13             <webuijsf:html id="html1">
14                 <webuijsf:head id="head1">
15                     <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
16                 </webuijsf:head>
17                 <webuijsf:body id="body1" style="-rave-layout: grid">
18                     <webuijsf:form id="form1">
19                         <webuijsf:label id="label1"
20                             style="font-family: 'Times New Roman','Times', serif; font-size: 36
21                             px; font-style: oblique; font-weight: lighter; left: 24px;
22                             top: 24px; position: absolute; text-decoration: underline;

```

```

width: 408px" text="Wiki 4Proyect - Andaluca"/>
19 <webuijsf:label id="label2"
20 style="background-color: rgb(102, 102, 102); height: 22px; left:
696px; top: 0px; position: absolute; width: 166px" text="Jos
Alberto Cordero Ros"/>
21 <webuijsf:label id="label3"
22 style="background-color: rgb(153, 153, 153); height: 22px; left:
696px; top: 24px; position: absolute; width: 166px" text="
Lorena Gutierrez Madroal"/>
23 <webuijsf:label id="label4"
24 style="background-color: rgb(204, 204, 204); height: 22px; left:
696px; top: 48px; position: absolute; width: 166px" text="
Sergio Puerta Aragn"/>
25 <webuijsf:label id="label5"
26 style="background-color: rgb(255, 255, 255); height: 22px; left:
696px; top: 72px; position: absolute; width: 166px" text="Flix
Reyes Fernndez"/>
27 </webuijsf:form>
28 <webuijsf:form binding="#{AdministrarPginas.form2}" id="form2" style="border-
top: solid gray 2px; height: 526px; left: 48px; top: 144px;
position: absolute; width: 960px">
29 <webuijsf:hyperlink actionExpression="#{AdministrarPginas.hyperlink1_action}"
" id="hyperlink1"
30 style="background-color: rgb(102, 102, 102); color: #ffffff; font-
family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
font-style: oblique; font-weight: bold; height: 22px; left:
24px; top: 24px; position: absolute; text-decoration: none;
width: 166px"
31 tabIndex="11" text="Ir a Bienvenida" url="/faces/index.jsp"/>
32 <webuijsf:hyperlink actionExpression="#{AdministrarPginas.hyperlink2_action}"
" binding="#{AdministrarPginas.hyperlink2}"
33 id="hyperlink2"
34 style="color: black; font-family: Verdana,Arial,Helvetica,sans-
serif; font-size: 12px; font-style: oblique; font-weight:
bold; height: 24px; left: 24px; top: 48px; position:
absolute; text-decoration: none; width: 168px"
35 tabIndex="12" text="Ir a mi pgina"/>
36 <webuijsf:hyperlink actionExpression="#{AdministrarPginas.hyperlink4_action}"
" binding="#{AdministrarPginas.hyperlink4}"
37 id="hyperlink4"
38 style="background-color: rgb(153, 153, 153); color: black; font-
family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
font-style: oblique; font-weight: bold; height: 24px; left:
24px; top: 72px; position: absolute; text-decoration: none;
width: 168px"
39 tabIndex="14" text="Listar pginas"/>
40 <webuijsf:button actionExpression="#{AdministrarPginas.button1_action}"
binding="#{AdministrarPginas.button1}" id="button1" style="color:
rgb(0, 0, 0); font-family: 'Verdana','Arial','Helvetica',sans-
serif; font-size: 10px; font-style: normal; height: 22px; left:
167px; top: 168px; position: absolute; width: 144px"/>
41 <webuijsf:label id="label6"
42 style="border-width: 2px; border-color: navy; font-family: Verdana
,Arial,Helvetica,sans-serif; font-size: 12px; font-style:
oblique; font-weight: lighter; height: 66px; left: 264px;
top: 48px; position: absolute; width: 598px" text="Estas son
las diferentes opciones que tenemos para administrar una
pgina: modificar el contenido de dicha pgina, dar permisos a
otros usuarios con respecto la pgina y publicar o
despublicar la pgina."/>
43 <webuijsf:button actionExpression="#{AdministrarPginas.button2_action}"
binding="#{AdministrarPginas.button2}" id="button2"
44 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
-size: 10px; height: 22px; left: 407px; top: 168px;
position: absolute; width: 72px" text="Modificar"/>
45 <webuijsf:button actionExpression="#{AdministrarPginas.button3_action}"
binding="#{AdministrarPginas.button3}" id="button3"

```

```

46         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 10px; height: 22px; left: 575px; top: 168px;
           position: absolute; width: 96px" text="Dar permisos"/>
47     <webuijsf:label id="label7"
48         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: oblique; font-weight: bold; left:
           168px; top: 216px; position: absolute" text="Nombre de la
           pagina"/>
49     <webuijsf:label id="label8"
50         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: oblique; font-weight: bold; left:
           168px; top: 288px; position: absolute" text="Descripcin de la
           pagina"/>
51     <webuijsf:staticText binding="#{AdministrarPginas.staticText1}" id="
           staticText1"
52         style="font-family: Verdana,Arial,Helvetica,sans-serif; font-size:
           12px; font-style: oblique; font-weight: bold; height: 22px;
           left: 408px; top: 216px; position: absolute; width: 144px"
           text="#{SessionBean1.pagina.nombrePagina}"/>
53     <webuijsf:staticText binding="#{AdministrarPginas.staticText2}" id="
           staticText2"
54         style="font-family: Verdana,Arial,Helvetica,sans-serif; font-size:
           12px; font-style: oblique; font-weight: bold; height: 22px;
           left: 408px; top: 288px; position: absolute; width: 144px"
           text="#{SessionBean1.pagina.descripcionPagina}"/>
55     <webuijsf:image height="204" id="image1" style="left: 672px; top: 312px;
           position: absolute"
56         url="http://www.mrboo.fr/wp-content/uploads/dotclear/
           illustrations/wiki.jpg" width="271"/>
57     </webuijsf:form>
58     </webuijsf:body>
59     </webuijsf:html>
60     </webuijsf:page>
61     </f:view>
62 </jsp:root>

```

CrearPagina.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8
9  import com.sun.rave.web.ui.appbase.AbstractPageBean;
10 import com.sun.webui.jsf.component.Button;
11 import com.sun.webui.jsf.component.Form;
12 import com.sun.webui.jsf.component.Hyperlink;
13 import com.sun.webui.jsf.component.Label;
14 import com.sun.webui.jsf.component.TextArea;
15 import com.sun.webui.jsf.component.TextField;
16 import javax.faces.FacesException;
17 import javax.faces.component.UIComponent;
18 import javax.faces.component.html.HtmlInputTextarea;
19 import javax.faces.event.ValueChangeEvent;
20
21
22
23 /**
24  * <p>Page bean that corresponds to a similarly named JSP page. This
25  * class contains component definitions (and initialization code) for
26  * all components that you have defined on this page, as well as

```

```

27 * lifecycle methods and event handlers where you may add behavior
28 * to respond to incoming events.</p>
29 *
30 * @version CrearPgina.java
31 * @version Created on 15-may-2009, 0:30:40
32 * @author WIND MOTEBOOK
33 */
34
35 public class CrearPgina extends AbstractPageBean {
36     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
37     PaginaAyuda ayudap=new PaginaAyuda();
38     /**
39      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
40      * This method is automatically generated, so any user-specified code inserted
41      * here is subject to being replaced.</p>
42      */
43     private void _init() throws Exception {
44     }
45     private HtmlInputTextarea contArea1 = new HtmlInputTextarea();
46
47     public HtmlInputTextarea getContArea1() {
48         return contArea1;
49     }
50
51     public void setContArea1(HtmlInputTextarea hit) {
52         this.contArea1 = hit;
53     }
54     private TextField textField1 = new TextField();
55
56     public TextField getTextField1() {
57         return textField1;
58     }
59
60     public void setTextField1(TextField tf) {
61         this.textField1 = tf;
62     }
63     private Label label9 = new Label();
64
65     public Label getLabel9() {
66         return label9;
67     }
68
69     public void setLabel9(Label l) {
70         this.label9 = l;
71     }
72     private Form form2 = new Form();
73
74     public Form getForm2() {
75         return form2;
76     }
77
78     public void setForm2(Form f) {
79         this.form2 = f;
80     }
81     private Label label10 = new Label();
82
83     public Label getLabel10() {
84         return label10;
85     }
86
87     public void setLabel10(Label l) {
88         this.label10 = l;
89     }
90     private Label label11 = new Label();
91
92     public Label getLabel11() {
93         return label11;

```

```

94     }
95
96     public void setLabel11(Label l) {
97         this.label11 = l;
98     }
99     private Button button1 = new Button();
100
101     public Button getButton1() {
102         return button1;
103     }
104
105     public void setButton1(Button b) {
106         this.button1 = b;
107     }
108     private TextField textField2 = new TextField();
109
110     public TextField getTextField2() {
111         return textField2;
112     }
113
114     public void setTextField2(TextField tf) {
115         this.textField2 = tf;
116     }
117
118
119     // </editor-fold>
120
121     /**
122     * <p>Construct a new Page bean instance.</p>
123     */
124     public CrearPgina() {
125     }
126
127     /**
128     * <p>Callback method that is called whenever a page is navigated to,
129     * either directly via a URL, or indirectly via page navigation.
130     * Customize this method to acquire resources that will be needed
131     * for event handlers and lifecycle methods, whether or not this
132     * page is performing post back processing.</p>
133     *
134     * <p>Note that, if the current request is a postback, the property
135     * values of the components do <strong>not</strong> represent any
136     * values submitted with this request. Instead, they represent the
137     * property values that were saved for this view when it was rendered.</p>
138     */
139     @Override
140     public void init() {
141         // Perform initializations inherited from our superclass
142         super.init();
143         // Perform application initialization that must complete
144         // *before* managed components are initialized
145         // TODO – add your own initialization code here
146
147         // <editor-fold defaultstate="collapsed" desc="Managed Component Initialization">
148         // Initialize automatically managed components
149         // *Note* – this logic should NOT be modified
150         try {
151             _init();
152         } catch (Exception e) {
153             log("CrearPgina Initialization Failure", e);
154             throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
155         }
156
157         // </editor-fold>
158         // Perform application initialization that must complete
159         // *after* managed components are initialized
160         // TODO – add your own initialization code here

```

```

161     }
162
163     /**
164      * <p>Callback method that is called after the component tree has been
165      * restored, but before any event processing takes place. This method
166      * will <strong>only</strong> be called on a postback request that
167      * is processing a form submit. Customize this method to allocate
168      * resources that will be required in your event handlers.</p>
169      */
170     @Override
171     public void preprocess() {
172     }
173
174     /**
175      * <p>Callback method that is called just before rendering takes place.
176      * This method will <strong>only</strong> be called for the page that
177      * will actually be rendered (and not, for example, on a page that
178      * handled a postback and then navigated to a different page). Customize
179      * this method to allocate resources that will be required for rendering
180      * this page.</p>
181      */
182     @Override
183     public void prerender() {
184     }
185
186     /**
187      * <p>Callback method that is called after rendering is completed for
188      * this request, if <code>init()</code> was called (regardless of whether
189      * or not this was the page that was actually rendered). Customize this
190      * method to release resources acquired in the <code>init()</code>,
191      * <code>preprocess()</code>, or <code>prerender()</code> methods (or
192      * acquired during execution of an event handler).</p>
193      */
194     @Override
195     public void destroy() {
196     }
197
198     /**
199      * <p>Return a reference to the scoped data bean.</p>
200      *
201      * @return reference to the scoped data bean
202      */
203     protected SessionBean1 getSessionBean1() {
204         return (SessionBean1) getBean("SessionBean1");
205     }
206
207     /**
208      * <p>Return a reference to the scoped data bean.</p>
209      *
210      * @return reference to the scoped data bean
211      */
212     protected RequestBean1 getRequestBean1() {
213         return (RequestBean1) getBean("RequestBean1");
214     }
215
216     /**
217      * <p>Return a reference to the scoped data bean.</p>
218      *
219      * @return reference to the scoped data bean
220      */
221     protected ApplicationBean1 getApplicationBean1() {
222         return (ApplicationBean1) getBean("ApplicationBean1");
223     }
224
225     public String hyperlink1_action() {
226         // TODO: Process the action. Return value is a navigation
227         // case name where null will return to the same page.

```

```

228     return "GoToIndex";
229 }
230
231 public String hyperlink2_action() {
232     // TODO: Process the action. Return value is a navigation
233     // case name where null will return to the same page.
234     String rol=getSessionBean1().getRolUlsr();
235     if(rol.equals("AD"))
236         return "GoToPaginaAD";
237     else
238         return "GoToPagina";
239 }
240
241 public String hyperlink3_action() {
242     // TODO: Process the action. Return value is a navigation
243     // case name where null will return to the same page.
244     return "GoToListarP";
245 }
246
247 public String hyperlink4_action() {
248     // TODO: Process the action. Return value is a navigation
249     // case name where null will return to the same page.
250     return "GoToAdministrarP";
251 }
252
253 public String button1_action() {
254     // TODO: Process the action. Return value is a navigation
255     // case name where null will return to the same page.
256     PaginaAyuda ayudap = new PaginaAyuda();
257
258     String contenido = contArea1.getValue().toString();
259
260     String nombre=(String)textField1.getText();
261
262     String descrip=(String)textField2.getText();
263
264
265     long ident=ayudap.buscarUltimoId()+1;
266     Paginas page=ayudap.buscarPaginaNom(nombre);
267     if( nombre == null || nombre.isEmpty() || contenido == null || contenido.isEmpty() || descrip == null ||
        descrip.isEmpty())
268     {
269         label9.setVisible(true);
270         label10.setVisible(false);
271         label11.setVisible(false);
272     }
273     else if(page!=null){
274         textField1.setText("");
275         label9.setVisible(false);
276
277         label10.setVisible(true);
278         label11.setVisible(false);
279     }
280     else{
281         Paginas p=new Paginas(ident,nombre,descrip,contenido,false);
282         ayudap.registrarPagina(p);
283         AdministracionPaginasId admId=new AdministracionPaginasId();
284         AdministracionPaginas admP=new AdministracionPaginas();
285         admId.setLogin(getSessionBean1().getLogin());
286         admId.setIdPagina(ident);
287         admP.setId(admId);
288         ayudap.registrarPaginaUsuario(admP);
289         label9.setVisible(false);
290
291         label10.setVisible(false);
292         label11.setVisible(true);
293     }

```



```

294         button1.setDisabled(true);
295     }
296
297     return null;
298 }
299
300
301
302
303
304
305 }

```

CrearPagina.jsp

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!--
3      Document : CrearPgina
4      Created on : 15-may-2009, 0:30:40
5      Author : WIND MOTEBOOK
6  -->
7  <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
8      html" xmlns:ice="http://www.icesoft.com/icefaces/component"
9      xmlns:jsf="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/webuijsf
10     ">
11     <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
12     <f:view>
13         <webuijsf:page id="page1">
14             <webuijsf:html id="html1">
15                 <webuijsf:head id="head1">
16                     <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
17                     <webuijsf:script type="text/javascript" url="/nicEdit.js"/>
18                     <webuijsf:script type="text/javascript">
19                         bkLib.onDomLoaded(function() {
20                             new nicEditor({fullPanel : true}).panelInstance('form2:contArea1');
21                         });
22                     </webuijsf:script>
23                 </webuijsf:head>
24                 <webuijsf:body id="body1" style="-rave-layout: grid">
25                     <webuijsf:form id="form1">
26                         <webuijsf:label id="label1"
27                             style="font-family: 'Times New Roman','Times', serif; font-size: 36
28                             px; font-style: oblique; font-weight: lighter; left: 24px;
29                             top: 24px; position: absolute; text-decoration: underline;
30                             width: 408px" text="Wiki 4Proyect - Andaluca"/>
31                         <webuijsf:label id="label2"
32                             style="background-color: rgb(102, 102, 102); height: 22px; left:
33                             696px; top: 0px; position: absolute; width: 166px" text="Jos
34                             Alberto Cordero Ros"/>
35                         <webuijsf:label id="label3"
36                             style="background-color: rgb(153, 153, 153); height: 22px; left:
37                             696px; top: 24px; position: absolute; width: 166px" text="
38                             Lorena Gutierrez Madroal"/>
39                         <webuijsf:label id="label4"
40                             style="background-color: rgb(204, 204, 204); height: 22px; left:
41                             696px; top: 48px; position: absolute; width: 166px" text="
42                             Sergio Puerta Aragon"/>
43                         <webuijsf:label id="label5"
44                             style="background-color: rgb(255, 255, 255); height: 22px; left:
45                             696px; top: 72px; position: absolute; width: 166px" text="Flix
46                             Reyes Fernandez"/>
47                     </webuijsf:form>
48                     <webuijsf:form binding="#{CrearPgina.form2}" id="form2" style="height: 552px;
49                         left: 24px; top: 120px; position: absolute; width: 840px">

```

```

36 <div style="background-color: white; left: 150px; top: 408px;
    position: absolute">
37 <h:inputTextarea binding="#{CrearPgina.contAreal}" cols="150" id="
    contAreal" rows="70" style="left: 72px; top: 360px; position:
    absolute"/>
38 </div>
39 <webuijsf:button actionExpression="#{CrearPgina.button1_action}" binding="#{
    CrearPgina.button1}" id="button1"
40 style="height: 24px; left: 647px; top: 360px; position: absolute"
    text="Guardar Cambios"/>
41 <webuijsf:textField binding="#{CrearPgina.textField1}" id="textField1"
    maxLength="20" style="left: 576px; top: 48px; position: absolute"/>
42 <webuijsf:image height="172" id="image1" style="left: 24px; top: 96px;
    position: absolute" url="/resources/noshacemosunwiki.jpg" width="
    170"/>
43 <webuijsf:hyperlink actionExpression="#{CrearPgina.hyperlink2_action}" id="
    hyperlink1"
44 style="color: black; font-family: 'Verdana','Arial','Helvetica',
    sans-serif; font-size: 12px; font-style: oblique; font-
    weight: bold; height: 24px; left: 24px; top: 24px; position:
    absolute; text-decoration: none; width: 168px"
45 tabIndex="12" text="Ir a mi pgina"/>
46 <webuijsf:hyperlink actionExpression="#{CrearPgina.hyperlink1_action}" id="
    hyperlink2"
47 style="background-color: rgb(102, 102, 102); color: rgb(255, 255,
    255); font-family: 'Verdana','Arial','Helvetica',sans-serif;
    font-size: 12px; font-style: oblique; font-weight: bold;
    height: 22px; left: 24px; top: 0px; position: absolute; text-
    decoration: none; width: 166px"
48 tabIndex="11" text="Ir a Bienvenida" url="/faces/index.jsp"/>
49 <webuijsf:label id="label6"
50 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font-
    -size: 12px; font-style: oblique; font-weight: bold; left:
    288px; top: 48px; position: absolute" text="Escriba el nombre
    de la pgina"/>
51 <webuijsf:label id="label7"
52 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font-
    -size: 12px; font-style: oblique; font-weight: bold; left:
    264px; top: 120px; position: absolute" text="Escriba la
    descripcin de la pgina"/>
53 <webuijsf:label id="label8"
54 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font-
    -size: 12px; font-style: oblique; font-weight: lighter;
    height: 46px; left: 240px; top: 216px; position: absolute;
    width: 502px" text="Antes de comenzar tiene que escribir un
    nombre para la pgina y comprobar que no existe dicho nombre.
    Recuerde utilizar un lenguaje correcto y contrastar la
    informacin que va a aadir antes de realizar los cambios."/>
55 <webuijsf:label binding="#{CrearPgina.label9}" id="label9"
56 style="border-width: 1px; border-style: solid; border-color: rgb
    (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
    background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
    font-family: 'Verdana','Arial','Helvetica',sans-serif; font-
    -size: 12px; font-style: oblique; font-weight: bold; height:
    46px; left: 624px; top: 288px; position: absolute; text-
    align: center; vertical-align: baseline; width: 192px"
57 text="Tiene que completar todos los campos" visible="false"/>
58 <webuijsf:label binding="#{CrearPgina.label10}" id="label10"
59 style="border-width: 1px; border-style: solid; border-color: rgb
    (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
    background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
    font-family: 'Verdana','Arial','Helvetica',sans-serif; font-
    -size: 12px; font-style: oblique; font-weight: bold; height:
    46px; left: 576px; top: 288px; position: absolute; text-
    align: center; vertical-align: baseline; width: 216px"
60 text="El nombre de la pgina ya existe. Elija otro" visible="false"/>

```

```

61         <webuijsf:label binding="#{CrearPgina.label11}" id="label11"
62             style="border: 1px solid navy; background-color: rgb(204, 204,
                255); color: navy; font-family: 'Verdana','Arial','Helvetica
                ',sans-serif; font-size: 12px; font-style: oblique; font-
                weight: bold; height: 22px; left: 576px; top: 288px;
                position: absolute; text-align: center; vertical-align:
                baseline; width: 142px"
63             text="Pgina guardada" visible="false"/>
64         <webuijsf:textField binding="#{CrearPgina.textField2}" id="textField2"
                maxLength="50" style="position: absolute; left: 576px; top: 120px;
                width: 120px; height: 24px"/>
65     </webuijsf:form>
66 </webuijsf:body>
67 </webuijsf:html>
68 </webuijsf:page>
69 </f:view>
70 </jsp:root>

```

DarPermisos.java

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : DarPermisos
4     Created on : 18-may-2009, 18:43:36
5     Author : WIND MOTEBOOK
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
    html" xmlns:jsf="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
    webuijsf">
8     <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9     <f:view>
10         <webuijsf:page id="page1">
11             <webuijsf:html id="html1">
12                 <webuijsf:head id="head1">
13                     <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
14                 </webuijsf:head>
15                 <webuijsf:body id="body1" style="-rave-layout: grid">
16                     <webuijsf:form id="form1">
17                         <webuijsf:label id="label1"
18                             style="font-family: 'Times New Roman','Times', serif; font-size: 36
                                px; font-style: oblique; font-weight: lighter; left: 24px;
                                top: 24px; position: absolute; text-decoration: underline;
                                width: 408px" text="Wiki 4Proyect - Andaluca"/>
19                         <webuijsf:label id="label2"
20                             style="background-color: rgb(102, 102, 102); height: 22px; left:
                                696px; top: 0px; position: absolute; width: 166px" text="Jos
                                Alberto Cordero Ros"/>
21                         <webuijsf:label id="label3"
22                             style="background-color: rgb(153, 153, 153); height: 22px; left:
                                696px; top: 24px; position: absolute; width: 166px" text="
                                Lorena Gutierrez Madroal"/>
23                         <webuijsf:label id="label4"
24                             style="background-color: rgb(204, 204, 204); height: 22px; left:
                                696px; top: 48px; position: absolute; width: 166px" text="
                                Sergio Puerta Aragn"/>
25                         <webuijsf:label id="label5"
26                             style="background-color: rgb(255, 255, 255); height: 22px; left:
                                696px; top: 72px; position: absolute; width: 166px" text="Flix
                                Reyes Fernandez"/>
27                         <webuijsf:image height="96" id="image1" style="left: 600px; top: 96px;
                                position: absolute"
28                             url="http://upload.wikimedia.org/wikipedia/commons/thumb/f/f7/
                                P_wiki_letter_w.svg/400px-P_wiki_letter_w.svg.png" width="120
                                "/>

```

```

29     </webuijsf:form>
30     <webuijsf:form binding="#{DarPermisos.form2}" id="form2" style="border-top:
        solid gray 2px; height: 382px; left: 48px; top: 144px; position:
        absolute; width: 840px">
31         <webuijsf:hyperlink actionExpression="#{DarPermisos.hyperlink1_action}" id="
            hyperlink1"
32             style="background-color: rgb(102, 102, 102); color: #ffffff; font-
                family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
                font-style: oblique; font-weight: bold; height: 22px; left:
                24px; top: 24px; position: absolute; text-decoration: none;
                width: 168px"
33             text="Ir a Bienvenida" url="/faces/index.jsp"/>
34         <webuijsf:hyperlink actionExpression="#{DarPermisos.hyperlink2_action}" id="
            hyperlink2"
35             style="color: black; font-family: Verdana,Arial,Helvetica,sans-
                serif; font-size: 12px; font-style: oblique; font-weight:
                bold; height: 24px; left: 24px; top: 48px; position:
                absolute; text-decoration: none; width: 168px" text="Ir a mi
                pgina"/>
36         <webuijsf:hyperlink actionExpression="#{DarPermisos.hyperlink3_action}" id="
            hyperlink3"
37             style="background-color: rgb(204, 204, 204); color: black; font-
                family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
                font-style: oblique; font-weight: bold; height: 24px; left:
                24px; top: 72px; position: absolute; text-decoration: none;
                width: 168px" text="Listar pginas"/>
38         <webuijsf:table augmentTitle="false" binding="#{DarPermisos.table1}" id="
            table1"
39             style="left: 240px; top: 216px; position: absolute; width: 0px"
            title="Usuarios a dar permisos" width="0">7777
40         <script>
41             function initAllRows() {
42                 var table = document.getElementById("form2:table1");
43                 table.initAllRows();
44             }
45         </script>
46         <webuijsf:tableRowGroup binding="#{DarPermisos.tableRowGroup1}" id="
            tableRowGroup1" rows="10"
47             selected="#{DarPermisos.selectedState}" sourceData="#{
                SessionBean1.listaUPermisos}" sourceVar="currentRow">
48             <webuijsf:tableColumn headerText="login" id="tableColumn1" sort="login"
                >
49                 <webuijsf:staticText id="staticText1" text="#{currentRow.value['
                    login']}" />
50             </webuijsf:tableColumn>
51             <webuijsf:tableColumn headerText="nombreUsr" id="tableColumn2" sort="
                nombreUsr">
52                 <webuijsf:staticText id="staticText2" text="#{currentRow.value['
                    nombreUsr']}" />
53             </webuijsf:tableColumn>
54             <webuijsf:tableColumn headerText="apellidosUsr" id="tableColumn3" sort
                ="apellidosUsr">
55                 <webuijsf:staticText id="staticText3" text="#{currentRow.value['
                    apellidosUsr']}" />
56             </webuijsf:tableColumn>
57             <webuijsf:tableColumn headerText="Permitir" id="tableColumn4" onClick=
                "setTimeout('initAllRows()',0)" selectId="#{DarPermisos.
                checkbox1.id}">
58                 <webuijsf:checkbox binding="#{DarPermisos.checkbox1}" id="
                    checkbox1" name=" #{DarPermisos.checkbox1.id}"
59                 selected="#{DarPermisos.selected}" selectedValue="#{
                    DarPermisos.selectedValue}" />
60             </webuijsf:tableColumn>
61         </webuijsf:tableRowGroup>
62     </webuijsf:table>
63     <webuijsf:label id="label16"

```

```

64         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: oblique; font-weight: lighter;
           left: 264px; top: 48px; position: absolute" text="En este
           listado aparecen todos los usuarios de nuestra wiki,
           seleccione todos aquellos a los que quiere dar permisos en
           esta pgina. Una vez hecho esto, pulse en el botn &quot;
           Aceptar&quot; para confirmarlo.."/>
65     <webuijsf:button actionExpression="#{DarPermisos.button1_action}" binding="#{
           DarPermisos.button1}" id="button1"
66         onClick="if(!confirm(&quot;Realmente desea dar plenos poderes
           sobre su pgina a estos usuarios?&quot;)) return false;"
67         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 10px; height: 22px; left: 263px; top: 144px;
           position: absolute; width: 72px" text="Aceptar"/>
68     <webuijsf:label id="label7"
69         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: oblique; font-weight: bolder; left:
           432px; top: 144px; position: absolute" text="Nombre pgina:"/>
70     <webuijsf:staticText binding="#{DarPermisos.staticText4}" id="staticText4"
71         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
           -size: 12px; font-style: oblique; font-weight: bold; height:
           24px; left: 552px; top: 144px; position: absolute; width:
           120px" text="#{SessionBean1.pagina.nombrePagina}"/>
72     <webuijsf:label binding="#{DarPermisos.label8}" id="label8"
73         style="border: 1px solid navy; background-color: rgb(204, 204,
           255); color: navy; font-family: Verdana,Arial,Helvetica,sans
           -serif; font-size: 12px; font-style: oblique; font-weight:
           bold; height: 44px; left: 456px; top: 144px; position:
           absolute; text-align: center; vertical-align: baseline;
           width: 238px"
74         text="Los permisos se concedieron satisfactoriamente." visible="
           false"/>
75     </webuijsf:form>
76     </webuijsf:body>
77     </webuijsf:html>
78     </webuijsf:page>
79     </f:view>
80 </jsp:root>

```

DarPermisos.jsp

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : DarPermisos
4     Created on : 18-may-2009, 18:43:36
5     Author : WIND MOTEBOOK
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsp/core" xmlns:h="http://java.sun.com/jsp/
   html" xmlns:jsp="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
   webuijsf">
8     <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9     <f:view>
10         <webuijsf:page id="page1">
11             <webuijsf:html id="html1">
12                 <webuijsf:head id="head1">
13                     <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
14                 </webuijsf:head>
15                 <webuijsf:body id="body1" style="-rave-layout: grid">
16                     <webuijsf:form id="form1">
17                         <webuijsf:label id="label1"
18                             style="font-family: 'Times New Roman','Times',serif; font-size: 36
                               px; font-style: oblique; font-weight: lighter; left: 24px;
                               top: 24px; position: absolute; text-decoration: underline;
                               width: 408px" text="Wiki 4Proyect - Andaluca"/>

```

```

19         <webuijsf:label id="label2"
20             style="background-color: rgb(102, 102, 102); height: 22px; left:
                696px; top: 0px; position: absolute; width: 166px" text="Jos
                Alberto Cordero Ros"/>
21         <webuijsf:label id="label3"
22             style="background-color: rgb(153, 153, 153); height: 22px; left:
                696px; top: 24px; position: absolute; width: 166px" text="
                Lorena Gutierrez Madroal"/>
23         <webuijsf:label id="label4"
24             style="background-color: rgb(204, 204, 204); height: 22px; left:
                696px; top: 48px; position: absolute; width: 166px" text="
                Sergio Puerta Aragn"/>
25         <webuijsf:label id="label5"
26             style="background-color: rgb(255, 255, 255); height: 22px; left:
                696px; top: 72px; position: absolute; width: 166px" text="Flix
                Reyes Fernndez"/>
27         <webuijsf:image height="96" id="image1" style="left: 600px; top: 96px;
                position: absolute"
28             url="http://upload.wikimedia.org/wikipedia/commons/thumb/f/f7/
                P_wiki_letter_w.svg/400px-P_wiki_letter_w.svg.png" width="120
                "/>
29     </webuijsf:form>
30     <webuijsf:form binding="#{DarPermisos.form2}" id="form2" style="border-top:
                solid gray 2px; height: 382px; left: 48px; top: 144px; position:
                absolute; width: 840px">
31         <webuijsf:hyperlink actionExpression="#{DarPermisos.hyperlink1_action}" id="
                hyperlink1"
32             style="background-color: rgb(102, 102, 102); color: #ffffff; font-
                family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
                font-style: oblique; font-weight: bold; height: 22px; left:
                24px; top: 24px; position: absolute; text-decoration: none;
                width: 168px"
33             text="Ir a Bienvenida" url="/faces/index.jsp"/>
34         <webuijsf:hyperlink actionExpression="#{DarPermisos.hyperlink2_action}" id="
                hyperlink2"
35             style="color: black; font-family: Verdana,Arial,Helvetica,sans-
                serif; font-size: 12px; font-style: oblique; font-weight:
                bold; height: 24px; left: 24px; top: 48px; position:
                absolute; text-decoration: none; width: 168px" text="Ir a mi
                pgina"/>
36         <webuijsf:hyperlink actionExpression="#{DarPermisos.hyperlink3_action}" id="
                hyperlink3"
37             style="background-color: rgb(204, 204, 204); color: black; font-
                family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
                font-style: oblique; font-weight: bold; height: 24px; left:
                24px; top: 72px; position: absolute; text-decoration: none;
                width: 168px" text="Listar pginas"/>
38         <webuijsf:table augmentTitle="false" binding="#{DarPermisos.table1}" id="
                table1"
39             style="left: 240px; top: 216px; position: absolute; width: 0px"
                title="Usuarios a dar permisos" width="0">7777
40         <script>
41             function initAllRows() {
42                 var table = document.getElementById("form2:table1");
43                 table.initAllRows();
44             }
45         </script>
46         <webuijsf:tableRowGroup binding="#{DarPermisos.tableRowGroup1}" id="
                tableRowGroup1" rows="10"
47             selected="#{DarPermisos.selectedState}" sourceData="#{
                SessionBean1.listaUPermisos}" sourceVar="currentRow">
48             <webuijsf:tableColumn headerText="login" id="tableColumn1" sort="login
                ">
49                 <webuijsf:staticText id="staticText1" text="#{currentRow.value['
                login']}" />
50             </webuijsf:tableColumn>

```

```

51         <webuijsf:tableColumn headerText="nombreUsr" id="tableColumn2" sort="
52             nombreUsr">
53             <webuijsf:staticText id="staticText2" text="#{currentRow.value['
54                 nombreUsr']}" />
55         </webuijsf:tableColumn>
56         <webuijsf:tableColumn headerText="apellidosUsr" id="tableColumn3" sort
57             ="apellidosUsr">
58             <webuijsf:staticText id="staticText3" text="#{currentRow.value['
59                 apellidosUsr']}" />
60         </webuijsf:tableColumn>
61         <webuijsf:tableColumn headerText="Permitir" id="tableColumn4" onClick=
62             "setTimeout('initAllRows()',0)" selectId="#{DarPermisos.
63                 checkbox1.id}">
64         <webuijsf:checkbox binding="#{DarPermisos.checkbox1}" id="
65             checkbox1" name="#{DarPermisos.checkbox1.id}"
66             selected="#{DarPermisos.selected}" selectedValue="#{
67                 DarPermisos.selectedValue}" />
68         </webuijsf:tableColumn>
69     </webuijsf:tableRowGroup>
70 </webuijsf:table>
71 <webuijsf:label id="label6"
72     style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
73         -size: 12px; font-style: oblique; font-weight: lighter;
74         left: 264px; top: 48px; position: absolute" text="En este
75         listado aparecen todos los usuarios de nuestra wiki,
76         seleccione todos aquellos a los que quiere dar permisos en
77         esta pgina. Una vez hecho esto, pulse en el botn &quot;
78         Aceptar&quot; para confirmarlo.." />
79     <webuijsf:button actionExpression="#{DarPermisos.button1_action}" binding="#{
80         DarPermisos.button1}" id="button1"
81         onClick="if(!confirm(&quot;Realmente desea dar plenos poderes
82             sobre su pgina a estos usuarios?&quot;)) return false;"
83         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
84             -size: 10px; height: 22px; left: 263px; top: 144px;
85             position: absolute; width: 72px" text="Aceptar" />
86     <webuijsf:label id="label7"
87         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
88             -size: 12px; font-style: oblique; font-weight: bolder; left:
89             432px; top: 144px; position: absolute" text="Nombre pgina:" />
90     <webuijsf:staticText binding="#{DarPermisos.staticText4}" id="staticText4"
91         style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
92             -size: 12px; font-style: oblique; font-weight: bold; height:
93             24px; left: 552px; top: 144px; position: absolute; width:
94             120px" text="#{SessionBean1.pagina.nombrePagina}" />
95     <webuijsf:label binding="#{DarPermisos.label8}" id="label8"
96         style="border: 1px solid navy; background-color: rgb(204, 204,
97             255); color: navy; font-family: Verdana,Arial,Helvetica,sans
98             -serif; font-size: 12px; font-style: oblique; font-weight:
99             bold; height: 44px; left: 456px; top: 144px; position:
100            absolute; text-align: center; vertical-align: baseline;
101            width: 238px"
102         text="Los permisos se concedieron satisfactoriamente." visible="
103             false" />
104 </webuijsf:form>
105 </webuijsf:body>
106 </webuijsf:html>
107 </webuijsf:page>
108 </f:view>
109 </jsp:root>

```

ListarPaginas.java

```

1  /*
2  * To change this template, choose Tools | Templates

```

```

3      * and open the template in the editor.
4      */
5
6      package wiki;
7
8      import com.sun.data.provider.RowKey;
9      import com.sun.rave.web.ui.appbase.AbstractPageBean;
10     import com.sun.webui.jsf.component.Button;
11     import com.sun.webui.jsf.component.Form;
12     import com.sun.webui.jsf.component.Hyperlink;
13     import com.sun.webui.jsf.component.Label;
14     import com.sun.webui.jsf.component.RadioButton;
15     import com.sun.webui.jsf.component.Table;
16     import com.sun.webui.jsf.component.TableRowGroup;
17     import com.sun.webui.jsf.event.TableSelectPhaseListener;
18     import java.util.List;
19     import javax.faces.FacesException;
20
21     /**
22      * <p>Page bean that corresponds to a similarly named JSP page. This
23      * class contains component definitions (and initialization code) for
24      * all components that you have defined on this page, as well as
25      * lifecycle methods and event handlers where you may add behavior
26      * to respond to incoming events.</p>
27      *
28      * @version ListarPginas.java
29      * @version Created on 15-may-2009, 0:57:54
30      * @author WIND MOTEBOOK
31      */
32
33     public class ListarPginas extends AbstractPageBean {
34         // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
35
36         /**
37          * <p>Automatically managed component initialization. <strong>WARNING:</strong>
38          * This method is automatically generated, so any user-specified code inserted
39          * here is subject to being replaced.</p>
40          */
41         private void _init() throws Exception {
42         }
43         private Table table1 = new Table();
44
45         public Table getTable1() {
46             return table1;
47         }
48
49         public void setTable1(Table t) {
50             this.table1 = t;
51         }
52         private Hyperlink hyperlink1 = new Hyperlink();
53
54         public Hyperlink getHyperlink1() {
55             return hyperlink1;
56         }
57
58         public void setHyperlink1(Hyperlink h) {
59             this.hyperlink1 = h;
60         }
61         private Hyperlink hyperlink2 = new Hyperlink();
62
63         public Hyperlink getHyperlink2() {
64             return hyperlink2;
65         }
66
67         public void setHyperlink2(Hyperlink h) {
68             this.hyperlink2 = h;
69         }

```



```

70     private Form form2 = new Form();
71
72     public Form getForm2() {
73         return form2;
74     }
75
76     public void setForm2(Form f) {
77         this.form2 = f;
78     }
79     private Button button2 = new Button();
80
81     public Button getButton2() {
82         return button2;
83     }
84
85     public void setButton2(Button b) {
86         this.button2 = b;
87     }
88
89     // </editor-fold>
90     private PaginaAyuda ayuda=new PaginaAyuda();
91     private Paginas pagSel;
92
93
94     private TableSelectPhaseListener tablePhaseListener =
95         new TableSelectPhaseListener();
96
97
98     public void setSelected(Object object) {
99         RowKey rowKey = (RowKey)getValue("#{currentRow.tableRow}");
100         if (rowKey != null) {
101             tablePhaseListener.setSelected(rowKey, object);
102         }
103     }
104
105     public Object getSelected(){
106         RowKey rowKey = (RowKey)getValue("#{currentRow.tableRow}");
107         return tablePhaseListener.getSelected(rowKey);
108     }
109
110     public Object getSelectedValue() {
111         RowKey rowKey = (RowKey)getValue("#{currentRow.tableRow}");
112         return (rowKey != null) ? rowKey.getRowId() : null;
113     }
114
115     public boolean getSelectedState() {
116         RowKey rowKey = (RowKey)getValue("#{currentRow.tableRow}");
117         return tablePhaseListener.isSelected(rowKey);
118     }
119     private TableRowGroup tableRowGroup1 = new TableRowGroup();
120
121     public TableRowGroup getTableRowGroup1() {
122         return tableRowGroup1;
123     }
124
125     public void setTableRowGroup1(TableRowGroup trg) {
126         this.tableRowGroup1 = trg;
127     }
128     private Button button1 = new Button();
129
130     public Button getButton1() {
131         return button1;
132     }
133
134
135
136

```

```

137     public void setButton1(Button b) {
138         this.button1 = b;
139     }
140     private RadioButton radioButton2 = new RadioButton();
141
142     public RadioButton getRadioButton2() {
143         return radioButton2;
144     }
145
146     public void setRadioButton2(RadioButton rb) {
147         this.radioButton2 = rb;
148     }
149     private Label label7 = new Label();
150
151     public Label getLabel7() {
152         return label7;
153     }
154
155     public void setLabel7(Label l) {
156         this.label7 = l;
157     }
158     private Label label8 = new Label();
159
160     public Label getLabel8() {
161         return label8;
162     }
163
164     public void setLabel8(Label l) {
165         this.label8 = l;
166     }
167
168     /**
169     * <p>Construct a new Page bean instance.</p>
170     */
171     public ListarPginas() {
172         String rol=getSessionBean1().getRolUsr();
173         if(rol.equals(" ")){
174             button2.setVisible(false);
175             //System.out.println(rol);
176         }
177     }
178
179
180     /**
181     * <p>Callback method that is called whenever a page is navigated to,
182     * either directly via a URL, or indirectly via page navigation.
183     * Customize this method to acquire resources that will be needed
184     * for event handlers and lifecycle methods, whether or not this
185     * page is performing post back processing.</p>
186     *
187     * <p>Note that, if the current request is a postback, the property
188     * values of the components do <strong>not</strong> represent any
189     * values submitted with this request. Instead, they represent the
190     * property values that were saved for this view when it was rendered.</p>
191     */
192     @Override
193     public void init() {
194         // Perform initializations inherited from our superclass
195         super.init();
196         // Perform application initialization that must complete
197         // *before* managed components are initialized
198         // TODO – add your own initialization code here
199
200         // <editor–fold defaultstate="collapsed" desc="Managed Component Initialization">
201         // Initialize automatically managed components
202         // *Note* – this logic should NOT be modified
203         try {

```

```

204     _init();
205     } catch (Exception e) {
206         log("ListarPaginas Initialization Failure", e);
207         throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
208     }
209
210     // </editor-fold>
211     // Perform application initialization that must complete
212     // *after* managed components are initialized
213     // TODO – add your own initialization code here
214 }
215
216 /**
217  * <p>Callback method that is called after the component tree has been
218  * restored, but before any event processing takes place. This method
219  * will <strong>only</strong> be called on a postback request that
220  * is processing a form submit. Customize this method to allocate
221  * resources that will be required in your event handlers.</p>
222  */
223 @Override
224 public void preprocess() {
225 }
226
227 /**
228  * <p>Callback method that is called just before rendering takes place.
229  * This method will <strong>only</strong> be called for the page that
230  * will actually be rendered (and not, for example, on a page that
231  * handled a postback and then navigated to a different page). Customize
232  * this method to allocate resources that will be required for rendering
233  * this page.</p>
234  */
235 @Override
236 public void prerender() {
237     if(getTableRowGroup1().getSelectedRowCount()>0)
238     {
239         button1.setRendered(true);
240         button2.setDisabled(false);
241         //Habilitar botn Administrar y Visualizar
242         RowKey[] selectedRowKeys=getTableRowGroup1().getSelectedRowKeys();
243         List<Paginas> pages=(List) getSessionBean1().getListasP();
244         int rowId = Integer.parseInt(selectedRowKeys[0].getRowId());
245         pagSel = pages.get(rowId);
246     }
247     String login=getSessionBean1().getLogin();
248     String rol=getSessionBean1().getRolUsr();
249     getSessionBean1().setListaP(ayuda.recuperarPaginas(login,rol));
250 }
251
252 /**
253  * <p>Callback method that is called after rendering is completed for
254  * this request, if <code>init()</code> was called (regardless of whether
255  * or not this was the page that was actually rendered). Customize this
256  * method to release resources acquired in the <code>init()</code>,
257  * <code>preprocess()</code>, or <code>prerender()</code> methods (or
258  * acquired during execution of an event handler).</p>
259  */
260 @Override
261 public void destroy() {
262 }
263
264 /**
265  * <p>Return a reference to the scoped data bean.</p>
266  *
267  * @return reference to the scoped data bean
268  */
269 protected SessionBean1 getSessionBean1() {
270

```

```

271     return (SessionBean1) getBean("SessionBean1");
272 }
273
274 /**
275  * <p>Return a reference to the scoped data bean.</p>
276  *
277  * @return reference to the scoped data bean
278  */
279 protected RequestBean1 getRequestBean1() {
280     return (RequestBean1) getBean("RequestBean1");
281 }
282
283 /**
284  * <p>Return a reference to the scoped data bean.</p>
285  *
286  * @return reference to the scoped data bean
287  */
288 protected ApplicationBean1 getApplicationBean1() {
289     return (ApplicationBean1) getBean("ApplicationBean1");
290 }
291
292 public String hyperlink1_action() {
293     // TODO: Process the action. Return value is a navigation
294     // case name where null will return to the same page.
295     return "GoToIndex";
296 }
297
298 public String hyperlink2_action() {
299     // TODO: Process the action. Return value is a navigation
300     // case name where null will return to the same page.
301     if(getSessionBean1().getRolUsr().equals("AD"))
302         return "GoToPageAd";
303     else
304         return "GoToPageUR";
305 }
306
307 public String hyperlink3_action() {
308     // TODO: Process the action. Return value is a navigation
309     // case name where null will return to the same page.
310     return "GoToCrearP";
311 }
312
313 public String hyperlink4_action() {
314     // TODO: Process the action. Return value is a navigation
315     // case name where null will return to the same page.
316     return null;
317 }
318
319 public String button1_action() {
320     // TODO: Process the action. Return value is a navigation
321     // case name where null will return to the same page.
322     if(getTableRowGroup1().getSelectedRowCount()>0)
323     {
324         RowKey[] selectedRowKeys = getTableRowGroup1().getSelectedRowKeys();
325         List<Paginas> pages =getSessionBean1().getListP();
326         int rowId = Integer.parseInt(selectedRowKeys[0].getRowId());
327         pagSel=pages.get(rowId);
328         getSessionBean1().setPagina(pagSel);
329         return "VisualizarPag";
330     }
331     else{
332         label8.setText("Selecciona una pagina");
333         label8.setVisible(true);
334         return null;
335     }
336 }
337 }

```

```

338
339 public String button2_action() {
340     // TODO: Process the action. Return value is a navigation
341     // case name where null will return to the same page.
342     if(getTableRowGroup1().getSelectedRowCount()>0)
343     {
344         String login=getSessionBean1().getLogin();
345         String rol=getSessionBean1().getRolUsr();
346
347         RowKey[] selectedRowKeys = getTableRowGroup1().getSelectedRowKeys();
348         List<Paginas> pages =getSessionBean1().getListP();
349         int rowId = Integer.parseInt(selectedRowKeys[0].getRowId());
350         pagSel=pages.get(rowId);
351
352         long id = pagSel.getIdPagina();
353
354
355
356         if(!getSessionBean1().getRolUsr().equals("AD "))
357         {
358             List<AdministracionPaginas> modify = ayuda.permisosModificacion(login,rol,id);
359             if(modify == null)
360             {
361                 return null;
362             }
363             else if(!modify.isEmpty())
364             {
365
366                 getSessionBean1().setPagina(pagSel);
367                 return "GoToAdministrarP";
368             }
369             else
370             {
371                 label7.setVisible(true);
372                 label8.setVisible(false);
373                 return null;
374             }
375         }
376         else
377         {
378             getSessionBean1().setPagina(pagSel);
379             return "GoToAdministrarP";
380         }
381     }
382     else{
383         label8.setText("Selecciona una pgina");
384         label8.setVisible(true);
385         return null;
386     }
387
388 }
389
390
391
392
393
394 }

```

ListarPaginas.jsp

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : ListarPginas
4     Created on : 15-may-2009, 0:57:54

```

```

5      Author : WIND MOTEBOOK
6      -->
7      <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
      html" xmlns:jsp="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
      webuijsf">
8      <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9      <f:view>
10         <webuijsf:page id="page1">
11             <webuijsf:html id="html1">
12                 <webuijsf:head id="head1">
13                     <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
14                 </webuijsf:head>
15                 <webuijsf:body id="body1" style="-rave-layout: grid">
16                     <webuijsf:form id="form1">
17                         <webuijsf:label id="label1"
18                             style="background-color: rgb(255, 255, 255); height: 22px; left:
19                             696px; top: 72px; position: absolute; width: 166px" text="Flix
20                             Reyes Fernndez"/>
21                         <webuijsf:label id="label2"
22                             style="background-color: rgb(204, 204, 204); height: 22px; left:
23                             696px; top: 48px; position: absolute; width: 166px" text="
24                             Sergio Puerta Aragn"/>
25                         <webuijsf:label id="label3"
26                             style="background-color: rgb(153, 153, 153); height: 22px; left:
27                             696px; top: 24px; position: absolute; width: 166px" text="
28                             Lorena Gutierrez Madroal"/>
29                         <webuijsf:label id="label4"
30                             style="background-color: rgb(102, 102, 102); height: 22px; left:
31                             696px; top: 0px; position: absolute; width: 166px" text="Jos
32                             Alberto Cordero Ros"/>
33                         <webuijsf:label id="label5"
34                             style="font-family: 'Times New Roman','Times',serif; font-size: 36
35                             px; font-style: oblique; font-weight: lighter; left: 24px;
36                             top: 24px; position: absolute; text-decoration: underline;
37                             width: 408px" text="Wiki 4Proyect - Andaluca"/>
38                     </webuijsf:form>
39                     <webuijsf:form binding="#{ListarPginas.form2}" id="form2" style="border-top:
40                     solid gray 2px; height: 382px; left: 48px; top: 168px; position:
41                     absolute; width: 840px">
42                         <webuijsf:hyperlink actionExpression="#{ListarPginas.hyperlink1_action}"
43                             binding="#{ListarPginas.hyperlink1}" id="hyperlink1"
44                             style="background-color: rgb(102, 102, 102); color: white; font-
45                             family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
46                             font-style: oblique; font-weight: bold; height: 22px; left:
47                             24px; top: 24px; position: absolute; text-decoration: none;
48                             width: 166px"
49                             tabIndex="11" text="Ir a Bienvenida" url="/faces/index.jsp"/>
50                         <webuijsf:hyperlink actionExpression="#{ListarPginas.hyperlink2_action}"
51                             binding="#{ListarPginas.hyperlink2}" id="hyperlink2"
52                             style="color: black; font-family: Verdana,Arial,Helvetica,sans-
53                             serif; font-size: 12px; font-style: oblique; font-weight:
54                             bold; height: 24px; left: 24px; top: 48px; position:
55                             absolute; text-decoration: none; width: 168px"
56                             tabIndex="12" text="Ir a mi pgina"/>
57                         <webuijsf:label id="label6"
58                             style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
59                             -size: 12px; font-style: oblique; font-weight: lighter;
60                             left: 192px; top: 96px; position: absolute; width: 454px" text
61                             ="Este es un listado de las pginas donde usted tiene
62                             permisos, bien porque usted sea el creador de la misma o
63                             bien porque otros usuarios de la wiki le hayan dado permisos
64                             . Si desea modificar algunos datos de la pgina pulse el botn
65                             de "Administrar";, si lo que desea es ver la pgina
66                             pulse el botn "Visualizar"."/>
67                         <webuijsf:table augmentTitle="false" id="table1" style="left: 72px; top: 288
68                             px; position: absolute; width: 434px"
69                             title="Listado de pginas" width="434">

```

```

39     <script>
40         function initAllRows()
41         {
42             var table = document.getElementById("form2:table1");
43             table.initAllRows();
44         }
45     </script>
46     <webuijsf:tableRowGroup binding="#{ListarPginas.tableRowGroup1}" id="
47         tableRowGroup1" rows="10"
48         selected="#{ListarPginas.selectedState}" sourceData="#{
49             SessionBean1.listaP}" sourceVar="currentRow">
50         <webuijsf:tableColumn headerText="Nombre" id="tableColumn1" sort="
51             nombrePagina">
52             <webuijsf:staticText id="staticText1" text="#{currentRow.value['
53                 nombrePagina']}" />
54         </webuijsf:tableColumn>
55         <webuijsf:tableColumn headerText="Descripcin" id="tableColumn2" sort="
56             descripcionPagina">
57             <webuijsf:staticText id="staticText2" text="#{currentRow.value['
58                 descripcionPagina']}" />
59         </webuijsf:tableColumn>
60         <webuijsf:tableColumn headerText="Seleccionar" id="tableColumn5"
61             onClick="setTimeout('initAllRows()',0)."
62             selectId="#{ListarPginas.radioButton2.id}" width="76">
63         <webuijsf:radioButton binding="#{ListarPginas.radioButton2}" id="
64             radioButton2" label=""
65             name="#{ListarPginas.radioButton2.id}" selected="#{
66                 ListarPginas.selected}" selectedValue="#{ListarPginas
67                     .selectedValue}" />
68         </webuijsf:tableColumn>
69     </webuijsf:tableRowGroup>
70 </webuijsf:table>
71 <webuijsf:button actionExpression="#{ListarPginas.button1_action}" binding="
72     #{ListarPginas.button1}" id="button1"
73     style="font-family: Verdana,Arial,Helvetica,sans-serif; font-size:
74         10px; height: 22px; left: 287px; top: 216px; position:
75         absolute; width: 72px" text="Visualizar"/>
76 <webuijsf:button actionExpression="#{ListarPginas.button2_action}" binding="
77     #{ListarPginas.button2}" id="button2"
78     style="font-family: Verdana,Arial,Helvetica,sans-serif; font-size:
79         10px; height: 22px; left: 95px; top: 216px; position:
80         absolute; width: 96px" text="Administrar"/>
81 <webuijsf:label binding="#{ListarPginas.label7}" id="label7"
82     style="border-width: 1px; border-style: solid; border-color: rgb
83         (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
84         background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
85         font-family: 'Verdana','Arial','Helvetica',sans-serif; font
86         -size: 12px; font-style: oblique; font-weight: bold; height:
87         44px; left: 408px; top: 216px; position: absolute; text-
88         align: center; vertical-align: baseline; width: 262px"
89     text="Lo siento no tienes permisos para modificar esta pgina"
90     visible="false"/>
91 <webuijsf:label binding="#{ListarPginas.label8}" id="label8"
92     style="border-width: 1px; border-style: solid; border-color: rgb
93         (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);
94         background-color: rgb(255, 204, 102); color: rgb(102, 0, 0);
95         font-family: 'Verdana','Arial','Helvetica',sans-serif; font
96         -size: 12px; font-style: oblique; font-weight: bold; height:
97         24px; left: 408px; top: 216px; position: absolute; text-
98         align: center; vertical-align: baseline; width: 214px" visible=
99     "false"/>
100 <webuijsf:image height="204" id="image1" style="left: 672px; top: 48px;
101     position: absolute"
102     url="http://www.mrboo.fr/wp-content/uploads/dotclear/
103     illustrations/wiki.jpg" width="271"/>
104 </webuijsf:form>
105 </webuijsf:body>

```

```

74         </webuijsf.html>
75         </webuijsf:page>
76     </f:view>
77 </jsp:root>

```

ModificarPagWiki.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package wiki;
7
8  import com.sun.rave.web.ui.appbase.AbstractPageBean;
9  import com.sun.webui.jsf.component.Button;
10 import com.sun.webui.jsf.component.Label;
11 import com.sun.webui.jsf.component.TextField;
12 import javax.faces.FacesException;
13 import javax.faces.component.html.HtmlInputTextarea;
14
15 /**
16  * <p>Page bean that corresponds to a similarly named JSP page. This
17  * class contains component definitions (and initialization code) for
18  * all components that you have defined on this page, as well as
19  * lifecycle methods and event handlers where you may add behavior
20  * to respond to incoming events.</p>
21  *
22  * @version ModificarPagWiki.java
23  * @version Created on 22-may-2009, 20:33:51
24  * @author ZoZe
25  */
26
27 public class ModificarPagWiki extends AbstractPageBean {
28     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
29
30     /**
31      * <p>Automatically managed component initialization. <strong>WARNING:</strong>
32      * This method is automatically generated, so any user-specified code inserted
33      * here is subject to being replaced.</p>
34      */
35     private void _init() throws Exception {
36     }
37     private HtmlInputTextarea contArea1 = new HtmlInputTextarea();
38
39     public HtmlInputTextarea getContArea1() {
40         return contArea1;
41     }
42
43     public void setContArea1(HtmlInputTextarea hit) {
44         this.contArea1 = hit;
45     }
46     private Button button1 = new Button();
47
48     public Button getButton1() {
49         return button1;
50     }
51
52     public void setButton1(Button b) {
53         this.button1 = b;
54     }
55     private TextField textField1 = new TextField();
56
57     public TextField getTextField1() {

```



```

58         return textField1;
59     }
60
61     public void setTextField1(TextField tf) {
62         this.textField1 = tf;
63     }
64     private TextField textField2 = new TextField();
65
66     public TextField getTextField2() {
67         return textField2;
68     }
69
70     public void setTextField2(TextField tf) {
71         this.textField2 = tf;
72     }
73     private Label label10 = new Label();
74
75     public Label getLabel10() {
76         return label10;
77     }
78
79     public void setLabel10(Label l) {
80         this.label10 = l;
81     }
82     private Label label11 = new Label();
83
84     public Label getLabel11() {
85         return label11;
86     }
87
88     public void setLabel11(Label l) {
89         this.label11 = l;
90     }
91
92     // </editor-fold>
93
94     /**
95      * <p>Construct a new Page bean instance.</p>
96      */
97     public ModificarPagWiki() {
98     }
99
100    /**
101     * <p>Callback method that is called whenever a page is navigated to,
102     * either directly via a URL, or indirectly via page navigation.
103     * Customize this method to acquire resources that will be needed
104     * for event handlers and lifecycle methods, whether or not this
105     * page is performing post back processing.</p>
106     *
107     * <p>Note that, if the current request is a postback, the property
108     * values of the components do <strong>not</strong> represent any
109     * values submitted with this request. Instead, they represent the
110     * property values that were saved for this view when it was rendered.</p>
111     */
112     @Override
113     public void init() {
114         // Perform initializations inherited from our superclass
115         super.init();
116         // Perform application initialization that must complete
117         // *before* managed components are initialized
118         // TODO — add your own initiation code here
119
120         // <editor-fold defaultstate="collapsed" desc="Managed Component Initialization">
121         // Initialize automatically managed components
122         // *Note* — this logic should NOT be modified
123         try {
124             _init();

```

```

125     } catch (Exception e) {
126         log("ModificarPagWiki Initialization Failure", e);
127         throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
128     }
129
130     // </editor-fold>
131     // Perform application initialization that must complete
132     // *after* managed components are initialized
133     // TODO – add your own initialization code here
134 }
135
136 /**
137  * <p>Callback method that is called after the component tree has been
138  * restored, but before any event processing takes place. This method
139  * will <strong>only</strong> be called on a postback request that
140  * is processing a form submit. Customize this method to allocate
141  * resources that will be required in your event handlers.</p>
142  */
143 @Override
144 public void preprocess() {
145 }
146
147 /**
148  * <p>Callback method that is called just before rendering takes place.
149  * This method will <strong>only</strong> be called for the page that
150  * will actually be rendered (and not, for example, on a page that
151  * handled a postback and then navigated to a different page). Customize
152  * this method to allocate resources that will be required for rendering
153  * this page.</p>
154  */
155 @Override
156 public void prerender() {
157 }
158
159 /**
160  * <p>Callback method that is called after rendering is completed for
161  * this request, if <code>init()</code> was called (regardless of whether
162  * or not this was the page that was actually rendered). Customize this
163  * method to release resources acquired in the <code>init()</code>,
164  * <code>preprocess()</code>, or <code>prerender()</code> methods (or
165  * acquired during execution of an event handler).</p>
166  */
167 @Override
168 public void destroy() {
169 }
170
171 /**
172  * <p>Return a reference to the scoped data bean.</p>
173  *
174  * @return reference to the scoped data bean
175  */
176 protected RequestBean1 getRequestBean1() {
177     return (RequestBean1) getBean("RequestBean1");
178 }
179
180 /**
181  * <p>Return a reference to the scoped data bean.</p>
182  *
183  * @return reference to the scoped data bean
184  */
185 protected ApplicationBean1 getApplicationBean1() {
186     return (ApplicationBean1) getBean("ApplicationBean1");
187 }
188
189 /**
190  * <p>Return a reference to the scoped data bean.</p>
191  *

```

```

192     * @return reference to the scoped data bean
193     */
194     protected SessionBean1 getSessionBean1() {
195         return (SessionBean1) getBean("SessionBean1");
196     }
197
198     public String button1_action() {
199
200         PaginaAyuda ayudap = new PaginaAyuda();
201
202
203
204         String nombre = (String) textField1.getText();
205         String descrip = (String) textField2.getText();
206         String contenido = contArea1.getValue().toString();
207
208
209
210         getSessionBean1().getPagina().setNombrePagina(nombre);
211         getSessionBean1().getPagina().setDescripcionPagina(descrip);
212         getSessionBean1().getPagina().setContenido(contenido);
213         Paginas page=ayudap.buscarPaginaNom(nombre);
214
215         if(page!=null && page.getIdPagina()!=getSessionBean1().getPagina().getIdPagina() && page.
            getNombrePagina().equals(getSessionBean1().getPagina().getNombrePagina())){
216             textField1.setText("");
217             label11.setVisible(true);
218             label10.setVisible(false);
219             label11.setText("El nombre que quieres insertar ya existe. Elige otro");
220         }
221         else{
222             ayudap.modificarPagina(getSessionBean1().getPagina());
223             label10.setVisible(true);
224             label11.setVisible(false);
225             label10.setText("Pgina modificada con xito");
226
227         }
228         // TODO: Process the action. Return value is a navigation
229         // case name where null will return to the same page.
230         return null;
231     }
232
233     public String hyperlink2_action()
234     {
235         if(getSessionBean1().getRolUsr().equals("AD"))
236             return "GoToPageAd";
237         else
238             return "GoToPageUR";
239
240     }
241
242     public String hyperlink3_action() {
243         // TODO: Replace with your code
244         return "GoToListarP";
245     }
246
247     public String hyperlink1_action() {
248         // TODO: Replace with your code
249         return "GoToIndex";
250     }
251
252
253
254 }

```

ModificarPagWiki.jsp

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3     Document : ModificarPagWiki
4     Created on : 22-may-2009, 20:33:51
5     Author : ZoZe
6 -->
7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
  html" xmlns:jsp="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
  webuijsf">
8 <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9 <f:view>
10     <webuijsf:page id="page1">
11         <webuijsf:html id="html1">
12             <webuijsf:head id="head1">
13                 <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
14                 <webuijsf:script type="text/javascript" url="/nicEdit.js"/>
15                 <webuijsf:script type="text/javascript">
16                     bkLib.onDomLoaded(function() {
17                         new nicEditor({fullPanel : true}).panelInstance('form1:contAreal');
18                     });
19                 </webuijsf:script>
20             </webuijsf:head>
21             <webuijsf:body id="body1" style="-rave-layout: grid">
22                 <webuijsf:form id="form1">
23                     <webuijsf:label id="label1"
24                         style="font-family: 'Times New Roman','Times', serif; font-size: 36
25                             px; font-style: oblique; font-weight: lighter; left: 0px;
26                             top: 24px; position: absolute; text-decoration: underline;
27                             width: 408px" text="Wiki 4Proyect - Andaluca"/>
28                     <webuijsf:label id="label2"
29                         style="font-family: 'Verdana','Arial','Helvetica', sans-serif; font
30                             -size: 12px; font-style: oblique; font-weight: lighter;
31                             height: 22px; left: 264px; top: 120px; position: absolute;
32                             text-decoration: underline; width: 118px" text="Modificando
33                             pgina"/>
34                     <webuijsf:label id="label3"
35                         style="background-color: rgb(255, 255, 255); height: 22px; left:
36                             672px; top: 72px; position: absolute; width: 166px" text="Flix
37                             Reyes Fernndez"/>
38                     <webuijsf:label id="label4"
39                         style="background-color: rgb(204, 204, 204); height: 22px; left:
40                             672px; top: 48px; position: absolute; width: 166px" text="
41                             Sergio Puerta Aragn"/>
42                     <webuijsf:label id="label5"
43                         style="background-color: rgb(153, 153, 153); height: 22px; left:
44                             672px; top: 24px; position: absolute; width: 166px" text="
45                             Lorena Gutirrez Madroal"/>
46                     <webuijsf:label id="label6"
47                         style="background-color: rgb(102, 102, 102); height: 22px; left:
48                             672px; top: 0px; position: absolute; width: 166px" text="Jos
49                             Alberto Cordero Ros"/>
50                     <webuijsf:hyperlink actionExpression="#{ModificarPagWiki.hyperlink1_action}"
51                         id="hyperlink1"
52                         style="background-color: rgb(102, 102, 102); color: white; font-
53                             family: 'Verdana','Arial','Helvetica', sans-serif; font-size:
54                             12px; font-style: oblique; font-weight: bold; height: 22px;
55                             left: 24px; top: 168px; position: absolute; text-
56                             decoration: none; width: 168px"
57                         text="Ir a Bienvenida" url="/faces/index.jsp"/>
58                     <webuijsf:hyperlink actionExpression="#{ModificarPagWiki.hyperlink2_action}"
59                         id="hyperlink2"
60                         style="color: black; font-family: 'Verdana','Arial','Helvetica',
61                             sans-serif; font-size: 12px; font-style: oblique; font-
62                             weight: bold; height: 24px; left: 24px; top: 192px;

```

```

        position: absolute; text-decoration: none; width: 168px" text=
        "Ir a mi pgina"/>
40 <webuijsf:hyperlink actionExpression="#{ModificarPagWiki.hyperlink3_action}"
        id="hyperlink3"
41 style="background-color: rgb(204, 204, 204); color: black; font-
        family: 'Verdana','Arial','Helvetica',sans-serif; font-size:
        12px; font-style: oblique; font-weight: bold; height: 24px;
        left: 24px; top: 216px; position: absolute; text-
        decoration: none; width: 168px" text="Listar pginas"/>
42 <webuijsf:label id="label17"
43 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: oblique; font-weight: bold; left:
        360px; top: 216px; position: absolute; width: 214px" text="
        Escriba el nombre de la pgina"/>
44 <webuijsf:textField binding="#{ModificarPagWiki.textField1}" id="textField1
        " maxLength="20"
45 style="height: 24px; left: 648px; top: 216px; position: absolute;
        width: 120px" text="#{SessionBean1.pagina.nombrePagina}"/>
46 <webuijsf:label id="label18"
47 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: oblique; font-weight: bold; left:
        360px; top: 264px; position: absolute; width: 238px" text="
        Escriba la descripcin de la pgina"/>
48 <webuijsf:textField binding="#{ModificarPagWiki.textField2}" id="textField2
        " maxLength="50"
49 style="left: 648px; top: 264px; position: absolute" text="#{
        SessionBean1.pagina.descripcionPagina}"/>
50 <webuijsf:label id="label19"
51 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: oblique; font-weight: lighter;
        height: 46px; left: 96px; top: 336px; position: absolute;
        width: 502px" text="Si se desea cambiar el nombre de la pgina
        tendr que volver a comprobar que no existe dicho nombre.
        Recuerde utilizar un lenguaje correcto y contrastar la
        informacin que va a aadir antes de realizar los cambios."/>
52 <webuijsf:staticText id="staticText1"
53 style="font-family: 'Verdana','Arial','Helvetica',sans-serif; font
        -size: 12px; font-style: oblique; font-weight: bold; height:
        18px; left: 384px; top: 120px; position: absolute" text="#{
        SessionBean1.pagina.idPagina}"/>
54 <div style="background-color:white; left: 150px; top: 500px; position:
        absolute;">
55 <h:inputTextarea binding="#{ModificarPagWiki.contAreal}" cols="150" id="
        contAreal" rows="70"
56 style="left: 0px; top: 0px; position: absolute" value="#{
        SessionBean1.pagina.contenido}"/>
57 </div>
58 <webuijsf:image height="96" id="image1" style="left: 552px; top: 72px;
        position: absolute"
59 url="http://upload.wikimedia.org/wikipedia/commons/thumb/f/f7/
        P_wiki_letter_w.svg/400px-P_wiki_letter_w.svg.png" width="120
        "/>
60 <webuijsf:button actionExpression="#{ModificarPagWiki.button1_action}"
        binding="#{ModificarPagWiki.button1}" id="button1"
61 style="height: 24px; left: 575px; top: 432px; position: absolute;
        width: 95px" text="Guardar Cambios"/>
62 <webuijsf:label binding="#{ModificarPagWiki.label10}" id="label10"
63 style="border: 1px solid navy; background-color: rgb(204, 204,
        255); color: navy; font-family: 'Verdana','Arial','Helvetica
        ',sans-serif; font-size: 12px; font-style: oblique; font-
        weight: bold; height: 22px; left: 576px; top: 408px;
        position: absolute; text-align: center; vertical-align:
        baseline"
64 text="Label" visible="false"/>
65 <webuijsf:label binding="#{ModificarPagWiki.label11}" id="label11"
66 style="border-width: 1px; border-style: solid; border-color: rgb
        (153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0) rgb(153, 0, 0);

```

```

67         background-color: rgb(255, 204, 102); color: rgb(153, 0, 0);
68         font-family: 'Verdana','Arial','Helvetica',sans-serif; font
69         -size: 12px; font-style: oblique; font-weight: bold; height:
70         24px; left: 504px; top: 408px; position: absolute; text-
71         align: center; vertical-align: baseline; width: 46px"
72         text="Label" visible="false"/>
73     </webuijsf:form>
    </webuijsf:body>
</webuijsf:html>
</webuijsf:page>
</f:view>
</jsp:root>

```

PaginaWiki.java

```

1  /*
2   * PginaWiki.java
3   *
4   * Created on 16-may-2009, 18:41:02
5   * Copyright WIND MOTEBOOK
6   */
7  package wiki;
8
9  import com.sun.rave.web.ui.appbase.AbstractPageBean;
10 import javax.faces.FacesException;
11 import javax.faces.component.html.HtmlInputTextarea;
12
13
14 /**
15  * <p>Page bean that corresponds to a similarly named JSP page. This
16  * class contains component definitions (and initialization code) for
17  * all components that you have defined on this page, as well as
18  * lifecycle methods and event handlers where you may add behavior
19  * to respond to incoming events.</p>
20  */
21 public class PginaWiki extends AbstractPageBean {
22     // <editor-fold defaultstate="collapsed" desc="Managed Component Definition">
23     private int __placeholder;
24
25     /**
26     * <p>Automatically managed component initialization. <strong>WARNING:</strong>
27     * This method is automatically generated, so any user-specified code inserted
28     * here is subject to being replaced.</p>
29     */
30     private void _init() throws Exception {
31     }
32     private HtmlInputTextarea contArea1 = new HtmlInputTextarea();
33
34     public HtmlInputTextarea getContArea1() {
35         return contArea1;
36     }
37
38     public void setContArea1(HtmlInputTextarea hit) {
39         this.contArea1 = hit;
40     }
41
42     // </editor-fold>
43
44     /**
45     * <p>Construct a new Page bean instance.</p>
46     */
47     public PginaWiki() {
48     }
49

```

```

50  /**
51  * <p>Callback method that is called whenever a page is navigated to,
52  * either directly via a URL, or indirectly via page navigation.
53  * Customize this method to acquire resources that will be needed
54  * for event handlers and lifecycle methods, whether or not this
55  * page is performing post back processing.</p>
56  *
57  * <p>Note that, if the current request is a postback, the property
58  * values of the components do <strong>not</strong> represent any
59  * values submitted with this request. Instead, they represent the
60  * property values that were saved for this view when it was rendered.</p>
61  */
62  @Override
63  public void init() {
64      // Perform initializations inherited from our superclass
65      super.init();
66      // Perform application initialization that must complete
67      // *before* managed components are initialized
68      // TODO – add your own initialization code here
69
70      // <editor–fold defaultstate="collapsed" desc="Managed Component Initialization">
71      // Initialize automatically managed components
72      // *Note* – this logic should NOT be modified
73      try {
74          _init();
75      } catch (Exception e) {
76          log("PginaWiki Initialization Failure", e);
77          throw e instanceof FacesException ? (FacesException) e: new FacesException(e);
78      }
79
80      // </editor–fold>
81      // Perform application initialization that must complete
82      // *after* managed components are initialized
83      // TODO – add your own initialization code here
84  }
85
86  /**
87  * <p>Callback method that is called after the component tree has been
88  * restored, but before any event processing takes place. This method
89  * will <strong>only</strong> be called on a postback request that
90  * is processing a form submit. Customize this method to allocate
91  * resources that will be required in your event handlers.</p>
92  */
93  @Override
94  public void preprocess() {
95  }
96
97  /**
98  * <p>Callback method that is called just before rendering takes place.
99  * This method will <strong>only</strong> be called for the page that
100  * will actually be rendered (and not, for example, on a page that
101  * handled a postback and then navigated to a different page). Customize
102  * this method to allocate resources that will be required for rendering
103  * this page.</p>
104  */
105  @Override
106  public void prerender() {
107  }
108
109  /**
110  * <p>Callback method that is called after rendering is completed for
111  * this request, if <code>init()</code> was called (regardless of whether
112  * or not this was the page that was actually rendered). Customize this
113  * method to release resources acquired in the <code>init()</code>,
114  * <code>preprocess()</code>, or <code>prerender()</code> methods (or
115  * acquired during execution of an event handler).</p>
116  */

```

```

117  @Override
118  public void destroy() {
119  }
120
121  /**
122   * <p>Return a reference to the scoped data bean.</p>
123   *
124   * @return reference to the scoped data bean
125   */
126  protected RequestBean1 getRequestBean1() {
127      return (RequestBean1) getBean("RequestBean1");
128  }
129
130  /**
131   * <p>Return a reference to the scoped data bean.</p>
132   *
133   * @return reference to the scoped data bean
134   */
135  protected SessionBean1 getSessionBean1() {
136      return (SessionBean1) getBean("SessionBean1");
137  }
138
139  /**
140   * <p>Return a reference to the scoped data bean.</p>
141   *
142   * @return reference to the scoped data bean
143   */
144  protected ApplicationBean1 getApplicationBean1() {
145      return (ApplicationBean1) getBean("ApplicationBean1");
146  }
147
148
149
150  public String hyperlink3_action() {
151      // TODO: Process the action. Return value is a navigation
152      // case name where null will return to the same page.
153      return "GoToListarP";
154  }
155
156  public String hyperlink2_action() {
157      // TODO: Process the action. Return value is a navigation
158      // case name where null will return to the same page.
159      if(getSessionBean1().getRolUsr().equals("AD"))
160          return "GoToPageAd";
161      else
162          return "GoToPageUR";
163  }
164
165  public String hyperlink1_action() {
166      // TODO: Process the action. Return value is a navigation
167      // case name where null will return to the same page.
168      return "GoToIndex";
169  }
170
171  }

```

PaginaWiki.jsp

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!--
3      Document : PginaWiki2
4      Created on : 18-may-2009, 22:50:44
5      Author : ZoZe
6  -->

```



```

7 <jsp:root version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/
  html" xmlns:jsf="http://java.sun.com/JSP/Page" xmlns:webuijsf="http://www.sun.com/webui/
  webuijsf">
8 <jsp:directive.page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"/>
9 <f:view>
10     <webuijsf:page id="page1">
11         <webuijsf:html id="html1">
12             <webuijsf:head id="head1">
13                 <webuijsf:link id="link1" url="/resources/stylesheet.css"/>
14                 <webuijsf:script type="text/javascript" url="/tiny_mce/tiny_mce.js"/>
15                 <webuijsf:script type="text/javascript">
16                     tinyMCE.init({
17                         mode : "textareas",
18                         theme : "advanced",
19                         readonly : true
20                     });
21                 </webuijsf:script>
22             </webuijsf:head>
23             <webuijsf:body id="body1" style="-rave-layout: grid">
24                 <webuijsf:form id="form1">
25                     <webuijsf:label id="label1"
26                         style="font-family: 'Times New Roman','Times', serif; font-size: 36
27                             px; font-style: oblique; font-weight: lighter; left: 48px;
28                             top: 24px; position: absolute; text-decoration: underline;
29                             width: 408px" text="Wiki 4Proyect - Andaluca"/>
30                     <webuijsf:staticText id="staticText1"
31                         style="font-family: Verdana,Arial,Helvetica,sans-serif; font-size:
32                             12px; font-style: oblique; font-weight: lighter; height: 24
33                             px; left: 312px; top: 264px; position: absolute; width: 24px
34                             " text="#{SessionBean1.pagina.nombrePagina}"/>
35                     <webuijsf:hyperlink actionExpression="#{PginaWiki.hyperlink1_action}" id="
36                         hyperlink1"
37                         style="background-color: rgb(102, 102, 102); color: white; font-
38                             family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
39                             font-style: oblique; font-weight: bold; height: 22px; left:
40                             72px; top: 120px; position: absolute; text-decoration: none
41                             ; width: 168px"
42                         text="Ir a Bienvenida" url="/faces/index.jsp"/>
43                     <webuijsf:hyperlink actionExpression="#{PginaWiki.hyperlink2_action}" id="
44                         hyperlink2"
45                         style="color: black; font-family: Verdana,Arial,Helvetica,sans-
46                             serif; font-size: 12px; font-style: oblique; font-weight:
47                             bold; height: 24px; left: 72px; top: 144px; position:
48                             absolute; text-decoration: none; width: 168px" text="Ir a mi
49                             pgina"/>
50                     <webuijsf:hyperlink actionExpression="#{PginaWiki.hyperlink3_action}" id="
51                         hyperlink3"
52                         style="background-color: rgb(204, 204, 204); color: black; font-
53                             family: Verdana,Arial,Helvetica,sans-serif; font-size: 12px;
54                             font-style: oblique; font-weight: bold; height: 24px; left:
55                             72px; top: 168px; position: absolute; text-decoration: none
56                             ; width: 168px" text="Listar pginas"/>
57                     <webuijsf:label id="label2"
58                         style="font-family: 'Verdana','Arial','Helvetica', sans-serif; font
59                             -size: 12px; font-style: oblique; font-weight: lighter;
60                             left: 288px; top: 144px; position: absolute; width: 454px"
61                         text="Esta pgina slo muestra el contenido de la temtica de la
62                             misma y una breve descripcin. Si deseamos modificarla,
63                             tendremos que ir a "Listar pginas", seleccionarla
64                             y darle a "Administrar". En ese momento tendremos
65                             la opcin de modificarla."/>
66                     <webuijsf:label id="label3"
67                         style="background-color: rgb(255, 255, 255); font-family: 'Arial
68                             ','Helvetica', sans-serif; font-size: 12px; font-weight: bold
69                             ; height: 22px; left: 744px; top: 96px; position: absolute;
70                             width: 166px" text="Flix Reyes Fernndez"/>
71                     <webuijsf:label id="label4"

```

```

41         style="background-color: rgb(204, 204, 204); font-family: 'Arial
42         ', 'Helvetica', sans-serif; font-size: 12px; font-weight: bold
43         ; height: 22px; left: 744px; top: 72px; position: absolute;
44         width: 166px" text="Sergio Puerta Aragn"/>
45     <webuijsf:label id="label15"
46     style="background-color: rgb(153, 153, 153); font-family: 'Arial
47     ', 'Helvetica', sans-serif; font-size: 12px; font-weight: bold
48     ; height: 22px; left: 744px; top: 48px; position: absolute;
49     width: 166px" text="Lorena Gutierrez Madroal"/>
50     <webuijsf:label id="label16"
51     style="background-color: rgb(102, 102, 102); font-family: 'Arial
52     ', 'Helvetica', sans-serif; font-size: 12px; font-weight: bold
53     ; height: 22px; left: 744px; top: 24px; position: absolute;
54     width: 166px" text="Jos Alberto Cordero Ros"/>
55     <webuijsf:label id="label17"
56     style="font-size: 12px; font-style: oblique; font-weight: lighter;
57     left: 168px; top: 264px; position: absolute; text-
58     decoration: underline; width: 118px" text="Nombre pgina:"/>
59     <div style="background-color:white; left: 150px; top: 400px; position:
60     absolute; ">
61         <h:inputTextarea binding="#{PginaWiki.contArea1}" cols="150" id="
62         contArea1" readonly="true" rows="70"
63         style="left: 0px; top: 0px; position: absolute" value="#{
64             SessionBean1.pagina.contenido}" />
65     </div>
66     <webuijsf:label id="label18"
67     style="font-size: 12px; font-style: oblique; font-weight: lighter;
68     height: 22px; left: 168px; top: 312px; position: absolute;
69     text-decoration: underline; width: 118px" text="Descripcin"/>
70     <webuijsf:staticText id="staticText2"
71     style="font-family: Verdana,Arial,Helvetica,sans-serif; font-size:
72     12px; font-style: oblique; font-weight: lighter; height: 24
73     px; left: 312px; top: 312px; position: absolute" text="#{
74         SessionBean1.pagina.descripcionPagina}" />
75     <webuijsf:image height="177" id="image1" style="left: 672px; top: 216px;
76     position: absolute" url="/resources/mundo.png" width="288"/>
77 </webuijsf:form>
78 </webuijsf:body>
79 </webuijsf:html>
80 </webuijsf:page>
81 </f:view>
82 </jsp:root>

```



Figura 7.1: nicEditorIcons.gif

8

Capítulo

Wiki de desarrollo

Como se puede observar, nuestra Wiki está estructurada en varios puntos que servirán de guía de lo que hemos estado realizando durante el desarrollo del proyecto.

Por una parte, tenemos la sección de Reuniones de Equipo, donde describimos las distintas reuniones que hemos tenido, junto a su fecha, lugar y descripción de los puntos tratados.

También, tenemos una parte correspondiente a los distintos entregables que hemos estado desarrollando y que se van actualizando en cada fase (Entregables 1-9).

Y para dejar, a modo de histórico, todas las actividades que vamos realizando, tenemos la sección de descripción de tareas de cada fase, donde cada integrante del equipo va comentando que es lo que ha estado desarrollando, que resultados ha obtenido, con que problemas se ha encontrado, etc.

Por último, creímos oportuno abrir una sección de Sugerencias, para discutir, plantear, debatir todos aquellos aspectos que pudiesen prestar confusión o que fuesen interesantes tratarlos.

Este es el enlace a la página de inicio de nuestra Wiki: <https://www.assembla.com/wiki/show/dRk-s6kEar3Pi6eJe5afGb>

9

Capítulo

Manual de instalación

Para la instalación del wiki “Wiki 4-Proyect Andalucia” tenemos que dar los siguientes pasos que se detallan en el siguiente manual. Primero vamos a observar el diagrama de despliegue para echarle un vistazo general a los elementos y dispositivos necesarios para utilizar nuestro wiki.

9.1. Diagrama de despliegue

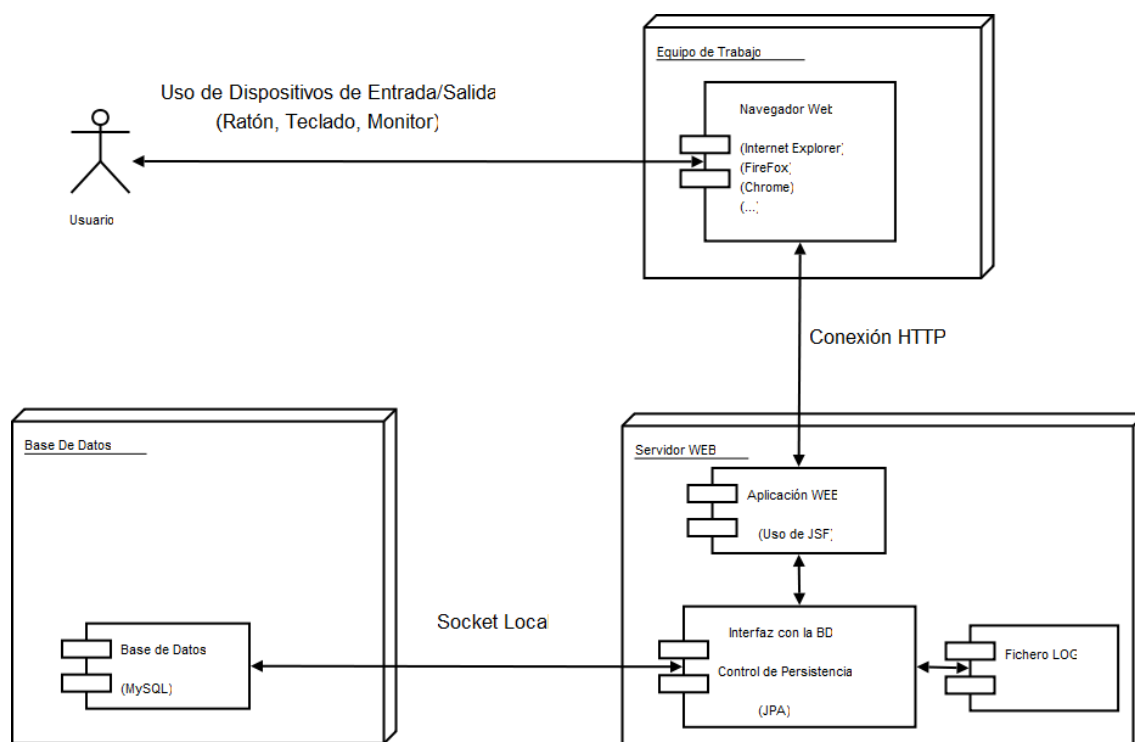


Figura 9.1: nicEditorIcons.gif

9.2. Instalación y configuración de MySQL

Como sistema de gestión de bases de datos, vamos a usar MySQL, ya que es gratuito y potente. Para instalar y configurar nuestro SGBD, vamos a seguir los siguientes pasos:

1. En primer lugar, debemos descargarnos el programa de instalación. Para ello, vamos a la página <http://www.mysql.com/> y entramos en la sección de descargas (Downloads). La versión que vamos a usar es plenamente funcional y completa, además de gratuita. Seleccionamos la versión MySQL Community Server.
2. Una vez descargado el archivo, comenzamos con su instalación. Al ejecutarlo, nos aparecerá la siguiente pantalla:



Figura 9.2: Primer paso en la instalación MySQL

Seleccionamos el tipo de instalación que queremos hacer, en nuestro caso Custom para configurar las opciones que queremos.

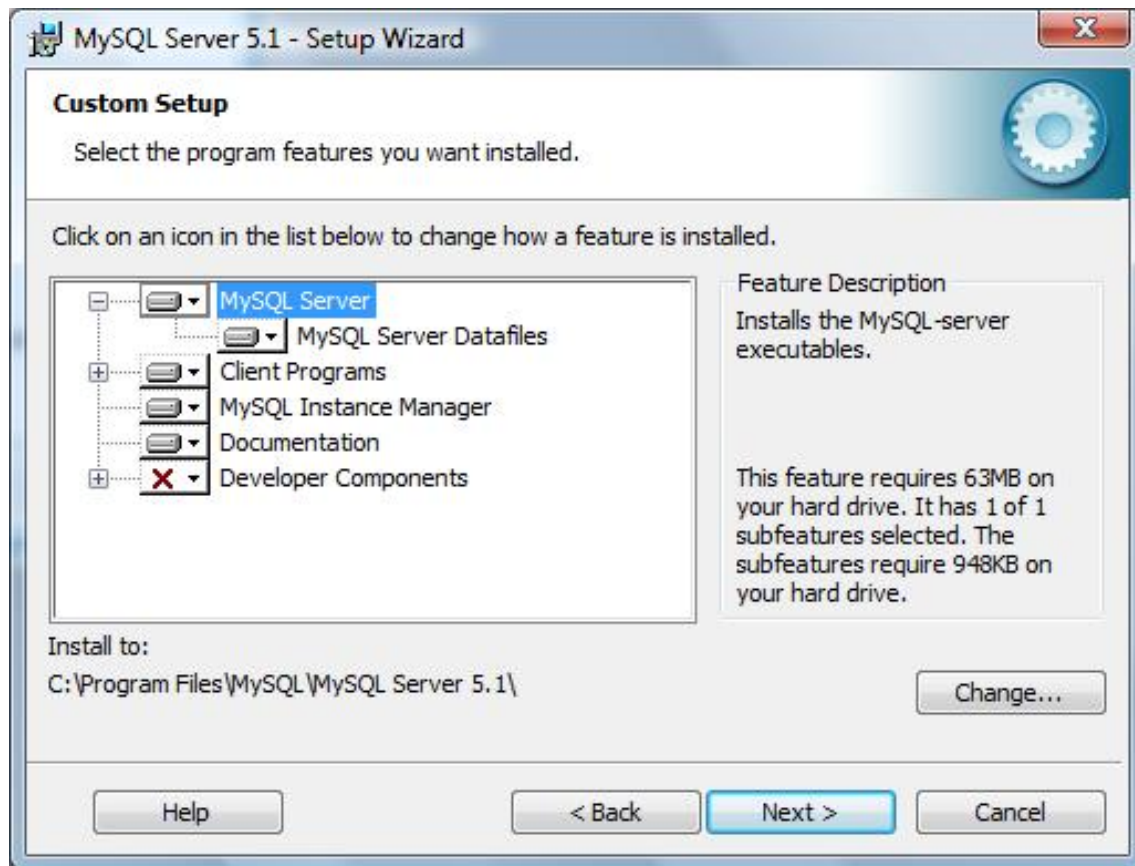


Figura 9.3: Segundo paso en la instalación - Tipo de instalación

Después de configurarlo, instalamos el programa, pero en la última ventana antes de salir del asistente, nos dan la opción de que sea el propio asistente quien configure la instancia de MySQL con las opciones por defecto. En nuestro caso, desactivamos esta casilla y lo instalamos nosotros mismos.

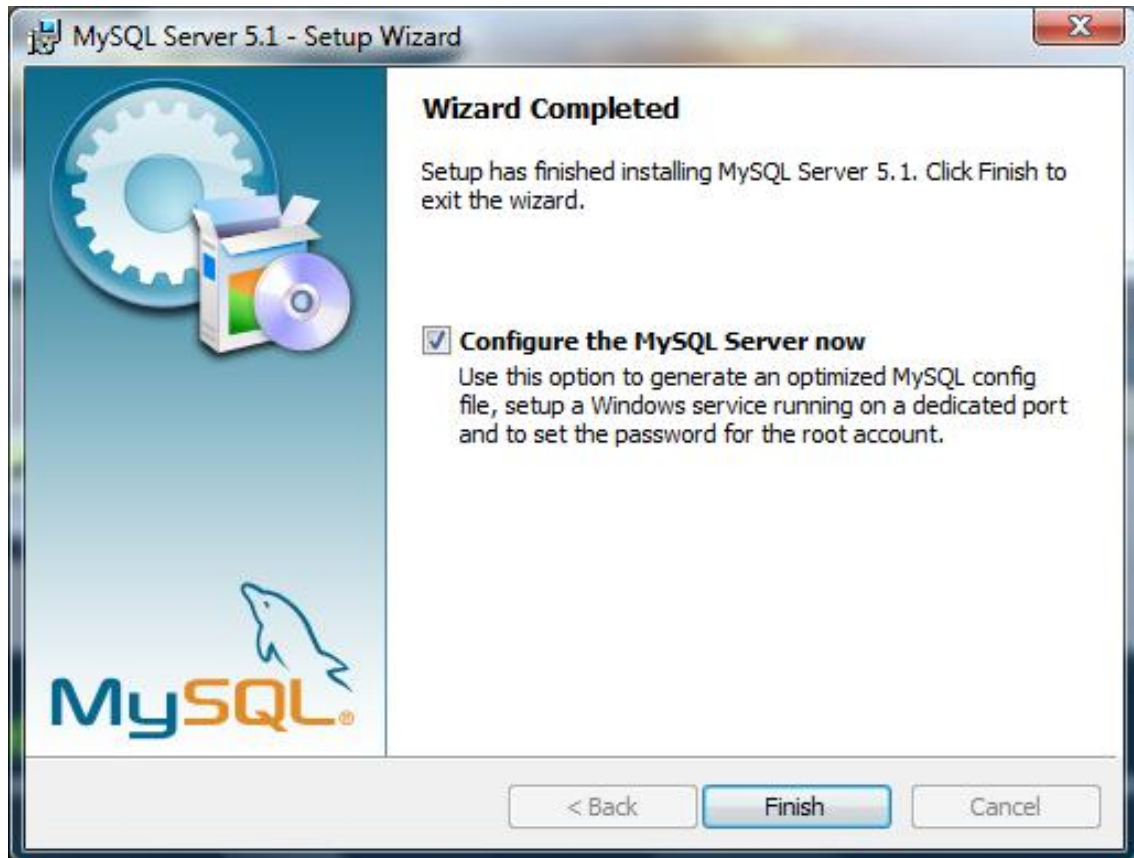


Figura 9.4: Tercer paso en la instalación - ¿Asistente?

3. A continuación vamos a configurar la instancia de MySQL. Nos vamos al grupo de programa creado, e iniciamos "MySQL Server Instance Config Wizard".



Figura 9.5: Primer paso en la configuración MySQL

4. El siguiente paso es escoger entre la configuración Standard o Detailed. La primera es una configuración general, mientras que la segunda nos permite configurar el SGBD a nuestro gusto. Optamos por esta segunda opción.

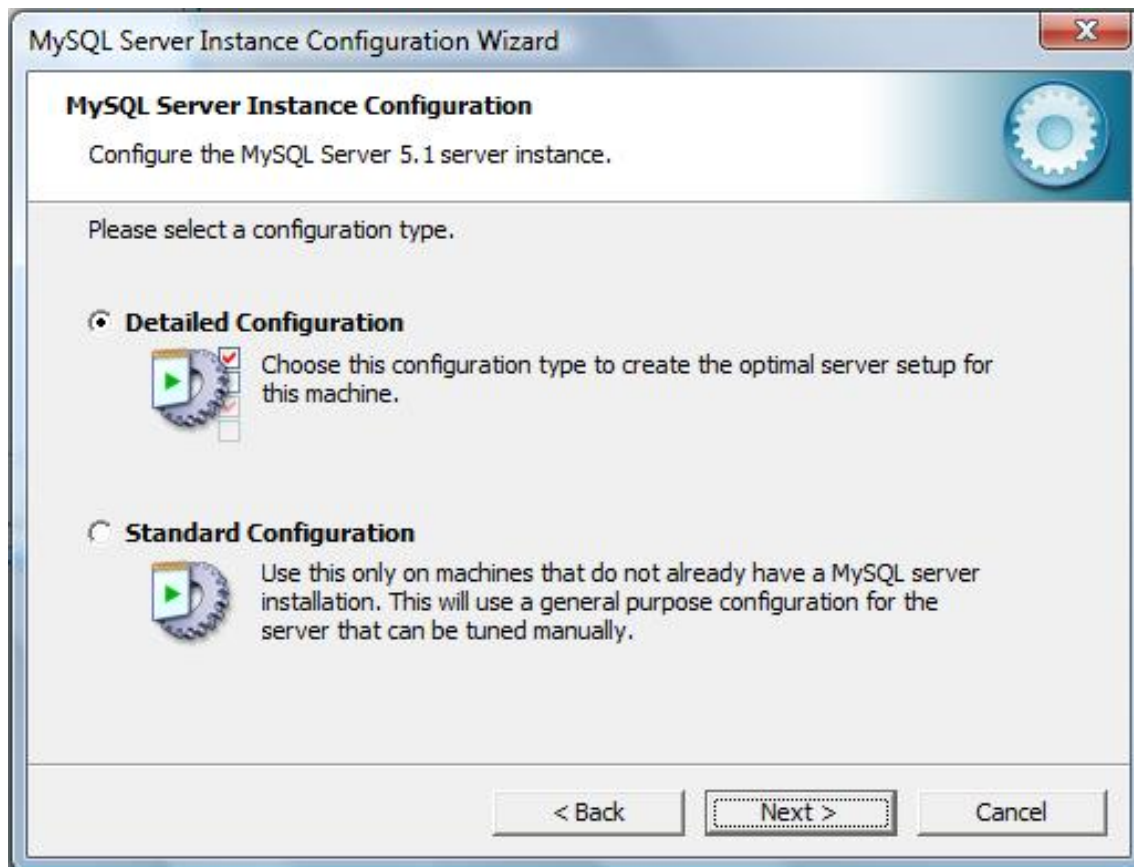


Figura 9.6: Segundo paso en la configuración MySQL

5. A continuación se nos pregunta qué tipo de servidor vamos a tener. Entre las opciones que nos ofrece, optamos por “Developer Machine”, suficiente para nuestro caso.



Figura 9.7: Tercer paso en la configuración MySQL - Servidor

6. El siguiente punto es escoger el tipo de base de datos a usar. En este caso, vamos a optar por una base de datos Transaccional únicamente (Transactional Database only), ya que nos proporcionará la funcionalidad suficiente para nuestro proyecto.

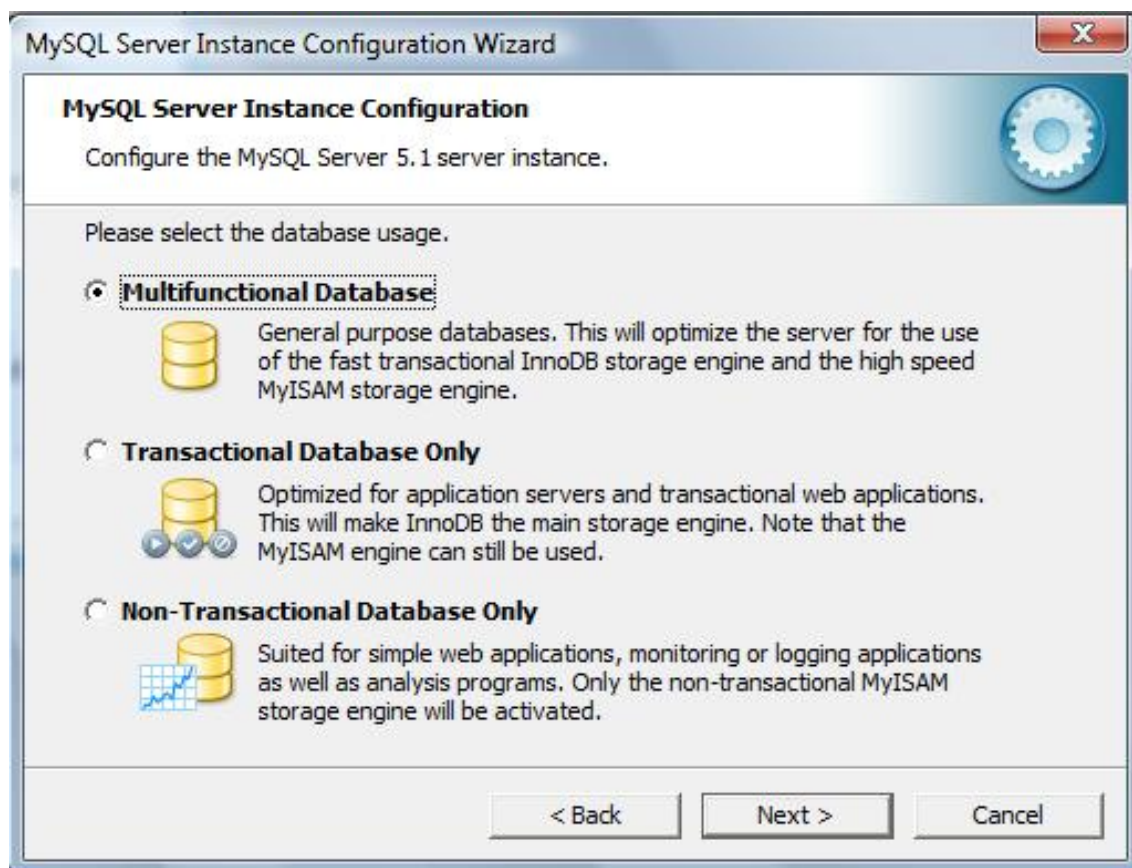


Figura 9.8: Cuarto paso en la configuración MySQL - Tipo de BD

7. La siguiente pantalla nos pregunta por las opciones de la red: si vamos a usar conexiones externas o únicamente conexiones internas desde nuestra máquina. En nuestro caso, optamos por conexiones externas. Seleccionamos el puerto por el que nos vamos a conectar y le indicamos que cree una excepción para el firewall. También dejamos marcado la opción "Enable Strict Mode", que ofrece más funcionalidad que un servidor de base de datos solamente.

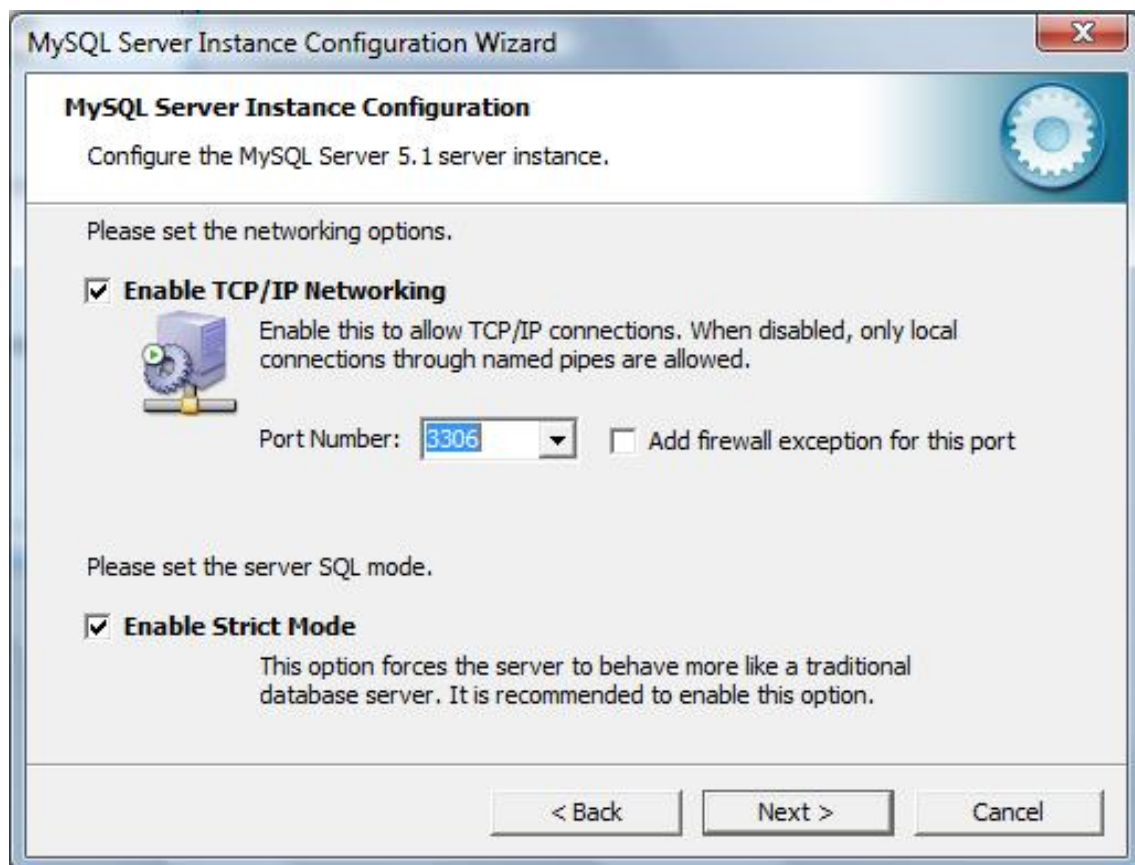


Figura 9.9: Quinto paso en la configuración MySQL - Opciones de red

8. A continuación le indicamos que instale el servicio del sistema MySQL, que lo inicie automáticamente cuando encendamos la máquina, y que cree incluya en la variable PATH el directorio de ejecución del servicio para poder llamarlo desde la consola del sistema.

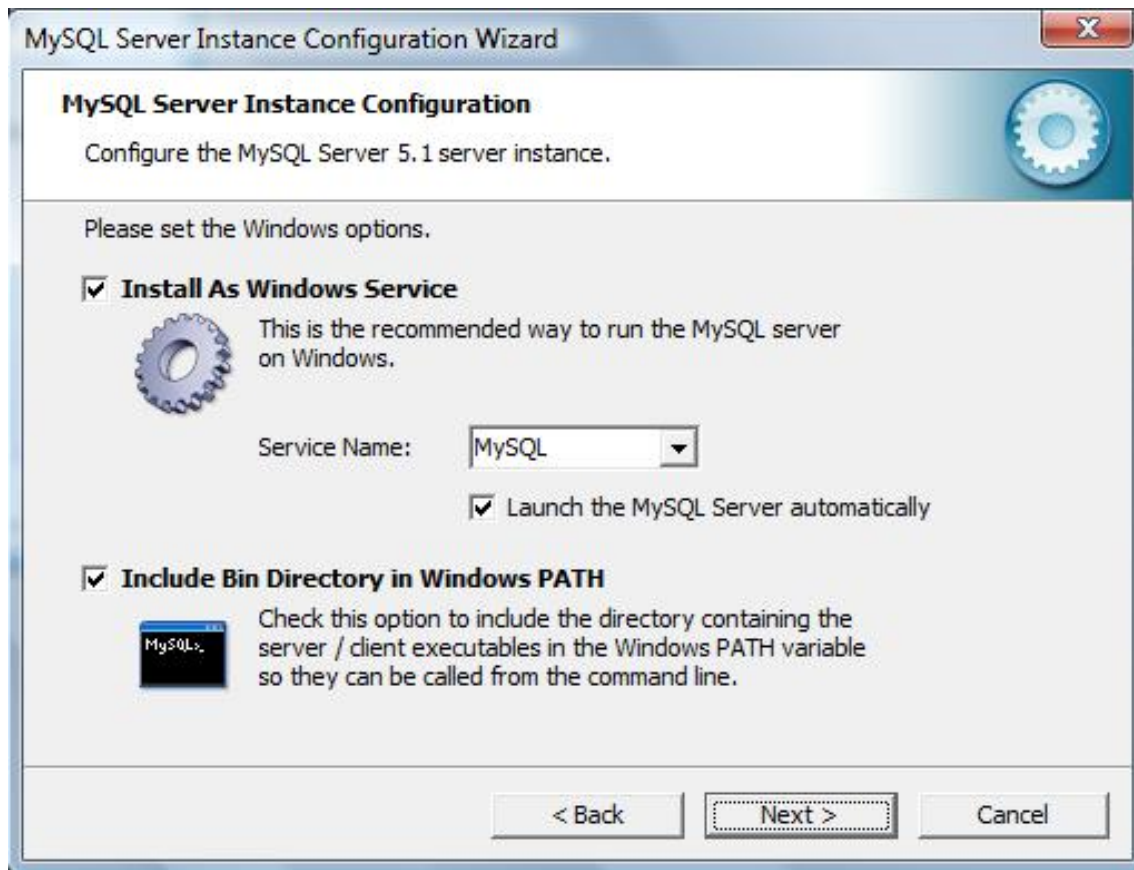


Figura 9.10: Sexto paso en la configuración MySQL

9. A continuación le indicamos que instale el servicio del sistema MySQL, que lo inicie automáticamente cuando encendamos la máquina, y que cree incluya en la variable PATH el directorio de ejecución del servicio para poder llamarlo desde la consola del sistema.
10. A continuación tendremos que introducir una contraseña para el usuario root(En nuestro caso sería “comandog”, para que exista conexión entre el wiki y la base de datos a través de hibernate). Y finalmente, si todo ha ido bien, en la siguiente pantalla ejecutará todas las opciones que hemos indicado y configurará nuestro sistema.

9.3. Creación de la Base de Datos

Una vez que está instalada la aplicación MySQL, vamos a crearnos nuestra Base de Datos. Aquí se adjuntan los scripts necesarios que se utilizarán para la creación de nuestro wiki.

Primero nos tendremos que dirigir a la consola de MySQL, tendremos que introducir la contraseña que se nos pide “comandog” y crearemos nuestra Base de Datos wiki dando los siguientes pasos (el texto que aparece a continuación, se podrá copiar y pegar directamente en consola o escribirlo en ella):

1. Script para la creación de la base de datos: createdb.sql

```

1  drop database if exists wiki;
2  create database wiki;
3  use wiki;
4  commit;
```

2. Script para la creación de usuarios, páginas y etiquetas, insercción de usuario Admin: insertdb.sql

```

1  CREATE TABLE wiki.usuarios (
2      login varchar(20) NOT NULL DEFAULT ' ',
3      password varchar(20) NOT NULL,
4      rol_usr varchar(2) NOT NULL,
5      nombre_usr varchar(30) NOT NULL,
6      apellidos_usr varchar(50) NOT NULL,
7      email_usr varchar(30) NOT NULL,
8      país_usr varchar(20) DEFAULT NULL,
9      ciudad_usr varchar(20) DEFAULT NULL,
10     telefono_usr varchar(9) DEFAULT NULL,
11     PRIMARY KEY (login)
12 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
13
14 CREATE TABLE wiki.paginas (
15     id_pagina bigint(20) NOT NULL,
16     nombre_pagina varchar(20) NOT NULL,
17     descripcion_pagina varchar(50) NOT NULL,
18     contenido longtext,
19     publicada bit(1) NOT NULL,
20     PRIMARY KEY (id_pagina)
21 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
22
23 CREATE TABLE wiki.administracion_paginas (
24     login varchar(20) NOT NULL,
25     id_pagina bigint(20) NOT NULL,
26     PRIMARY KEY ('login','id_pagina')
27 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
28
29 CREATE TABLE wiki.etiquetas (
30     id_etiqueta varchar(20) NOT NULL,
31     PRIMARY KEY ('id_etiqueta')
32 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
33
34 CREATE TABLE wiki.etiquetas_pagina (
35     id_pagina bigint(20) NOT NULL,
36     id_etiqueta varchar(20) NOT NULL,
37     PRIMARY KEY (id_pagina,id_etiqueta)
38 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
39
40 insert into wiki.usuarios values
41 ('Administrador','12345','AD','Administrador','Administrador','administrador@wiki.com',
42     null,null,null);
43
44 commit;
```

3. Script para borrar las tablas de usuarios, páginas y etiquetas: cleandb.sql

```
1 drop table if exists wiki.usuarios;
2 drop table if exists wiki.paginas;
3 drop table if exists wiki.administracion\_paginas;
4 drop table if exists wiki.etiquetas;
5 drop table if exists wiki.etiquetas\_pagina;
6 commit;
```

9.4. Apache Tomcat 6.0.18

Por último necesitaremos instalar un servidor Web para poder usar nuestra aplicación para ello hemos elegido Apache Tomcat. El proceso de instalación es bastante sencillo:

1. Visitaremos la página <http://apache.tomcat.org> e iremos a la sección download versión 6.x
2. En nuestro caso elegiremos la versión para windows, Windows Service Installer y descargaremos el archivo.

Una vez descargado el archivo le daremos a ejecutar y comenzaremos la instalación.

- Para comenzar la instalación pulsamos Next

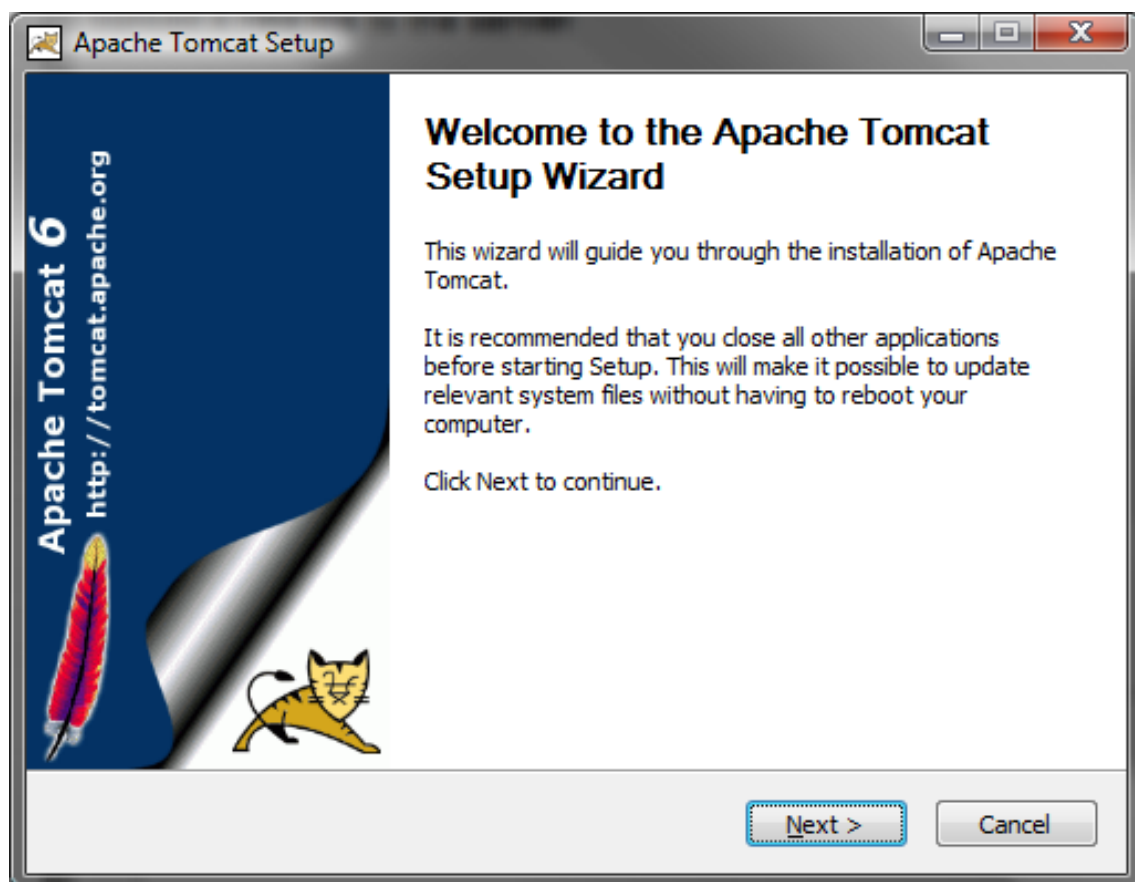


Figura 9.11: Instalación Tomcat paso 1

- Debemos aceptar la licencia de Tomcat

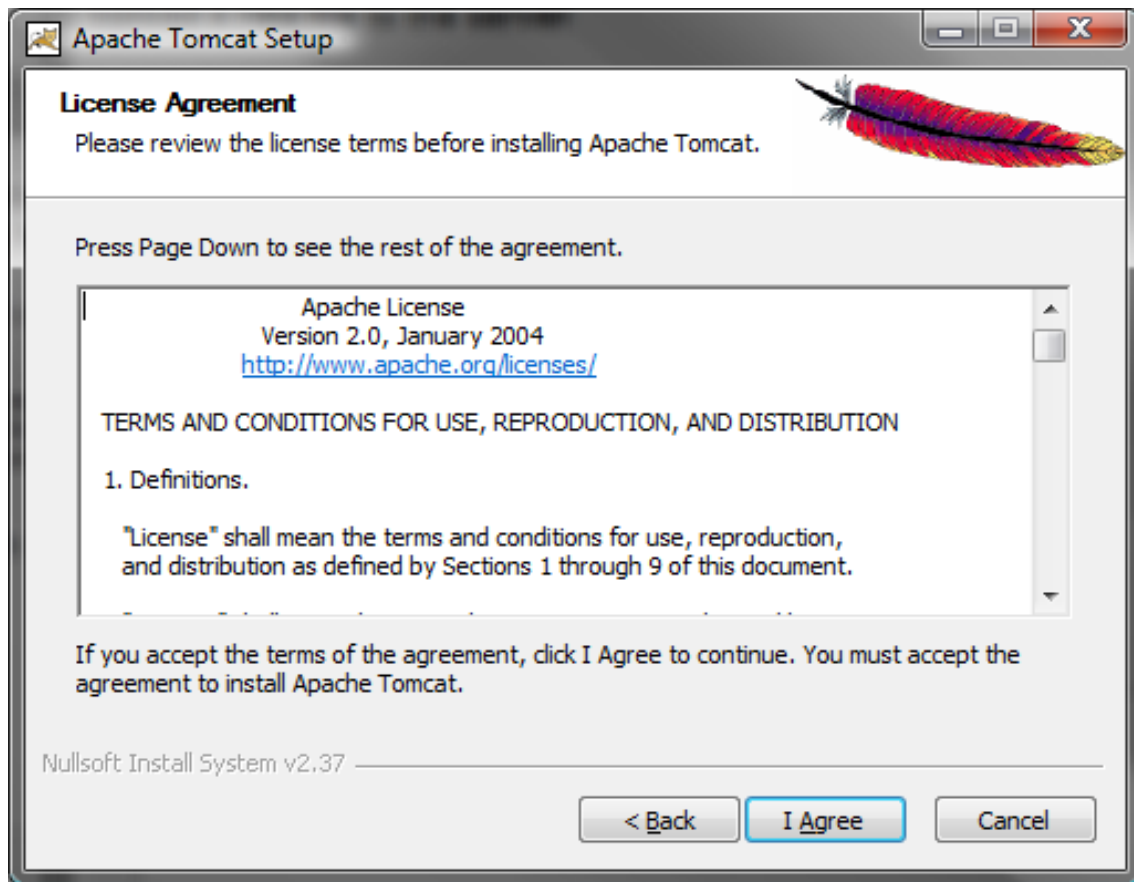


Figura 9.12: Instalación Tomcat paso 2

- Hay que elegir el directorio donde queremos instalar Tomcat

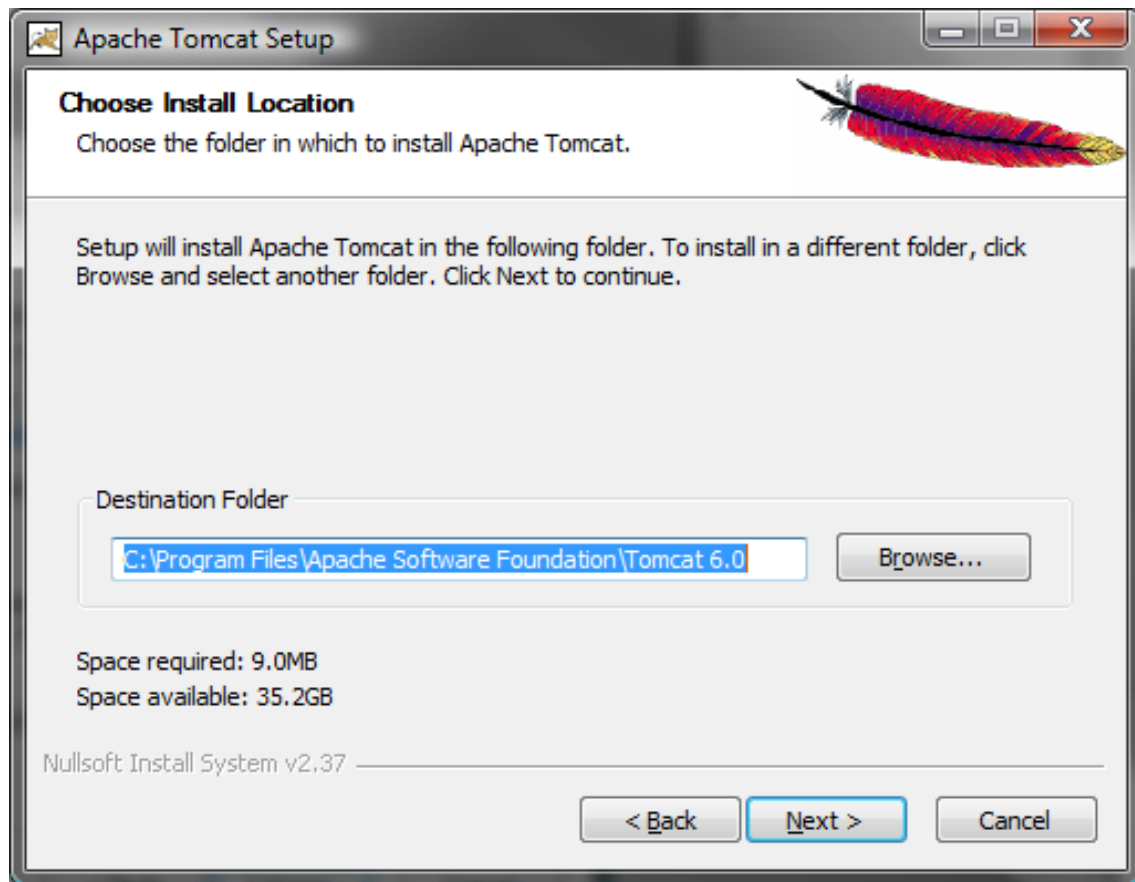


Figura 9.13: Instalación Tomcat paso 3

- Aquí debemos elegir el puerto por el cual queramos que funcione nuestro servidor, por defecto viene el 8080. Otro punto importante es elegir el usuario y clave para la administración de nuestro servidor Tomcat, eso queda a elección del usuario.

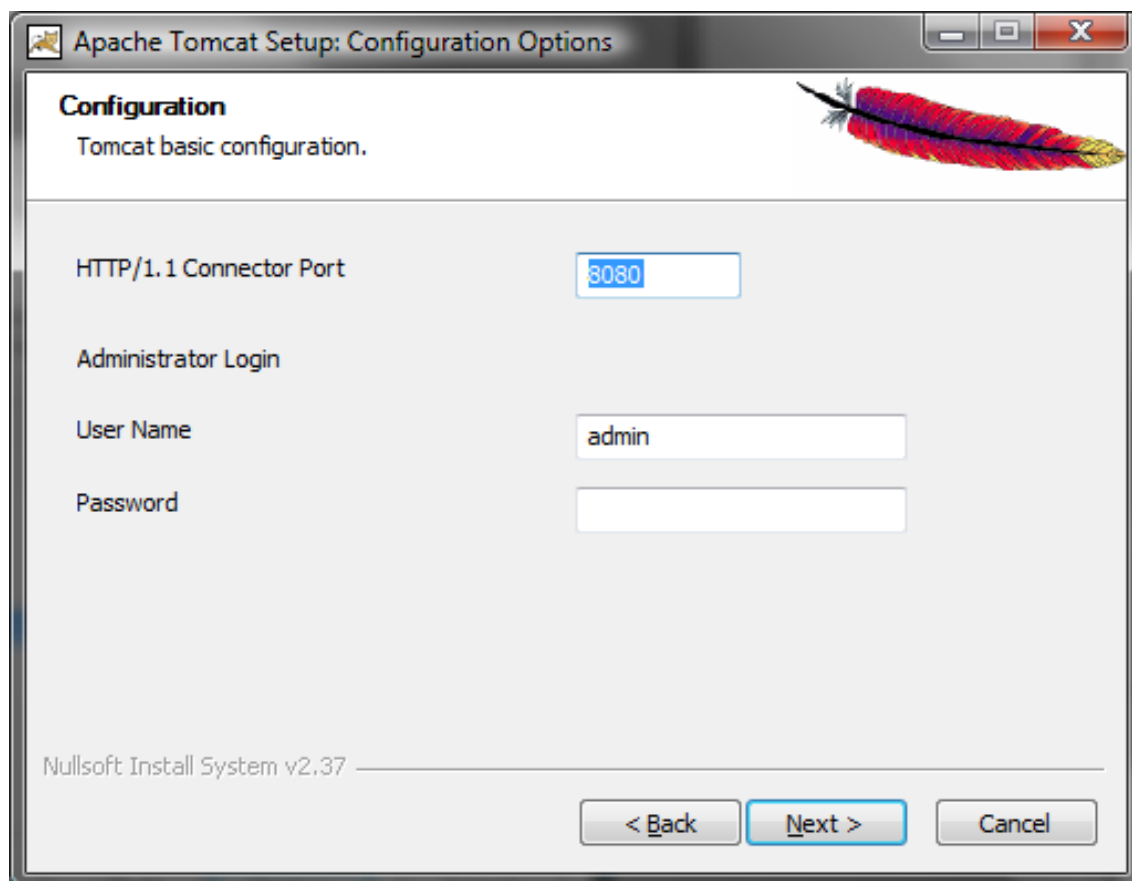


Figura 9.14: Instalación Tomcat paso 4

- Por último debemos proporcionar la ruta donde se encuentra instalado la JVM dentro de nuestra máquina. En caso de no tenerla instalada, pruebe con visitar esta página <http://www.java.com/es/download/manual.jsp> para descargarlo.

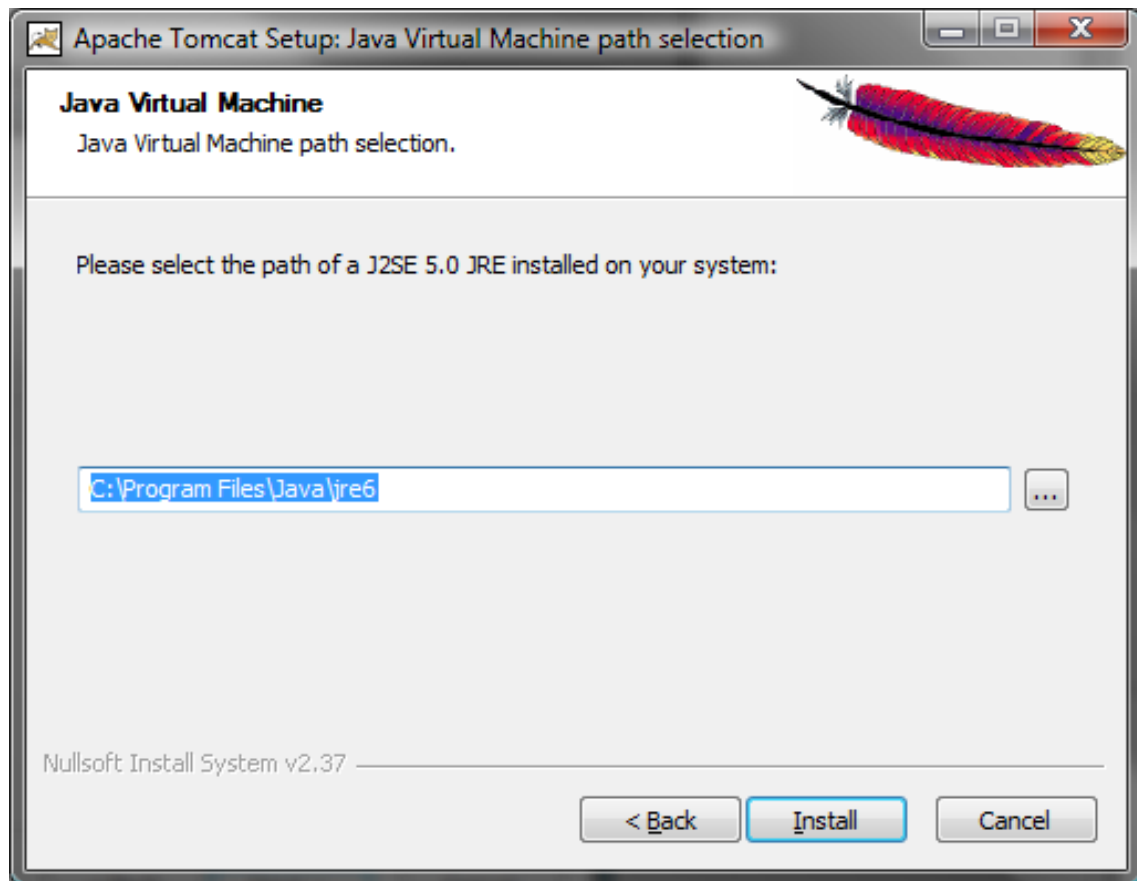


Figura 9.15: Instalación Tomcat paso 5

- Una vez hecho esto ya tenemos instalado apache Tomcat en nuestra máquina, el siguiente y último paso es desplegar nuestra aplicación en el servidor.
- Ahora en nuestra barra de tareas que se encuentra la parte inferior derecha de windows aparecerá un icono como éste:

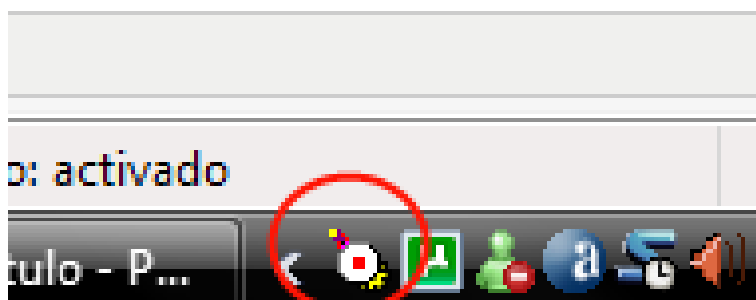


Figura 9.16: Instalación Tomcat paso 6

- Esto significa que nuestro servidor aún no está funcionando, por tanto le daremos al botón derecho y le daremos a Start Service tras lo cual arrancaremos nuestro servidor web. Una vez arrancado nos dirigiremos a nuestro navegador web y escribiremos la siguiente dirección `http://localhost:8080` y nos saldrá una pantalla como la siguiente:

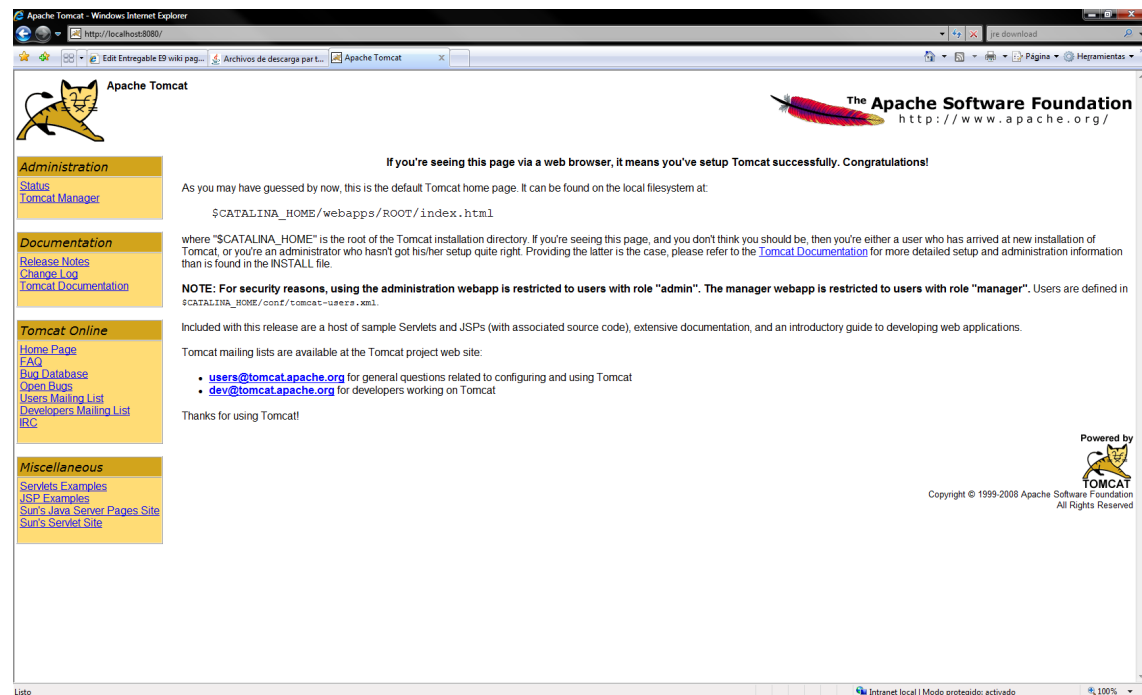


Figura 9.17: Instalación Tomcat paso 7

- Tras esto le daremos a Tomcat Manager en el menú de la izquierda y nos aparecerá que introduzcamos el login y el password de administración que hemos puesto en uno de los pasos anteriores para poder acceder a esta sección del sistema. Tras introducirlo nos encontramos en la siguiente página:

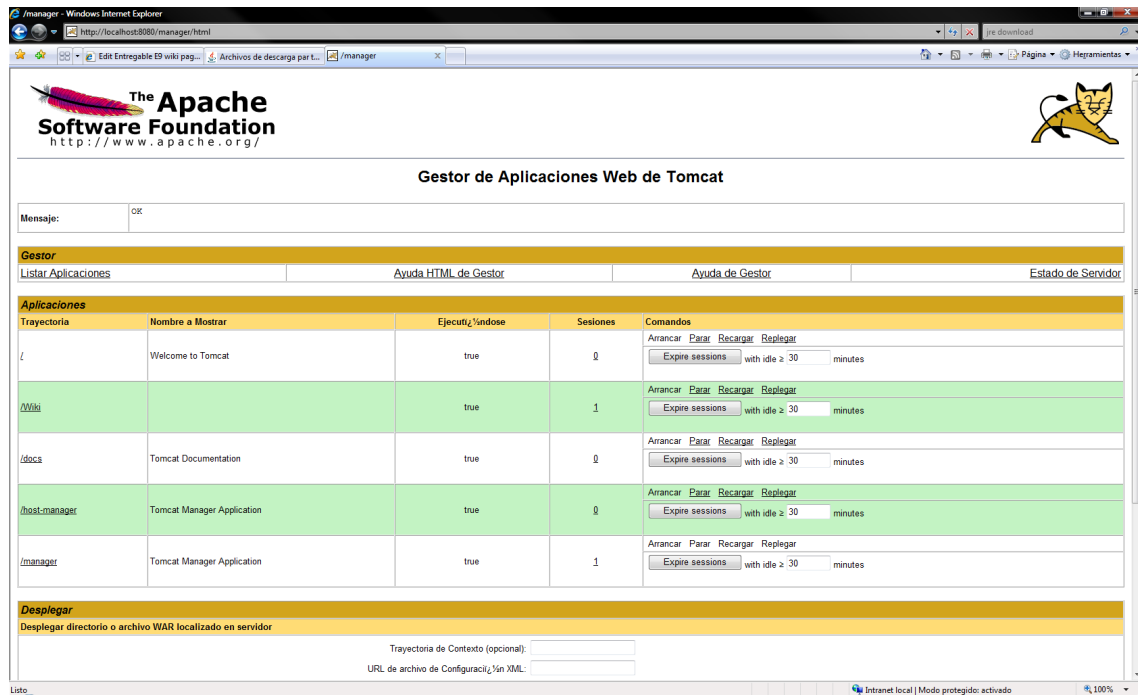


Figura 9.18: Instalación Tomcat paso 8

- En este paso nos iremos a la parte inferior de esta página donde pone Seleccione el archivo WAR a cargar: y le daremos a examinar. Nos saldrá un navegador de carpetas en el cual tenemos que buscar el archivo wiki.war que será suministrado a través del aula virtual para la entrega final junto con su documento y tras esto le daremos a desplegar. Cuando el servidor termine de desplegar, ya podremos arrancar nuestra aplicación, tan solo escribiendo esta dirección `http://localhost:8080/Wiki`.

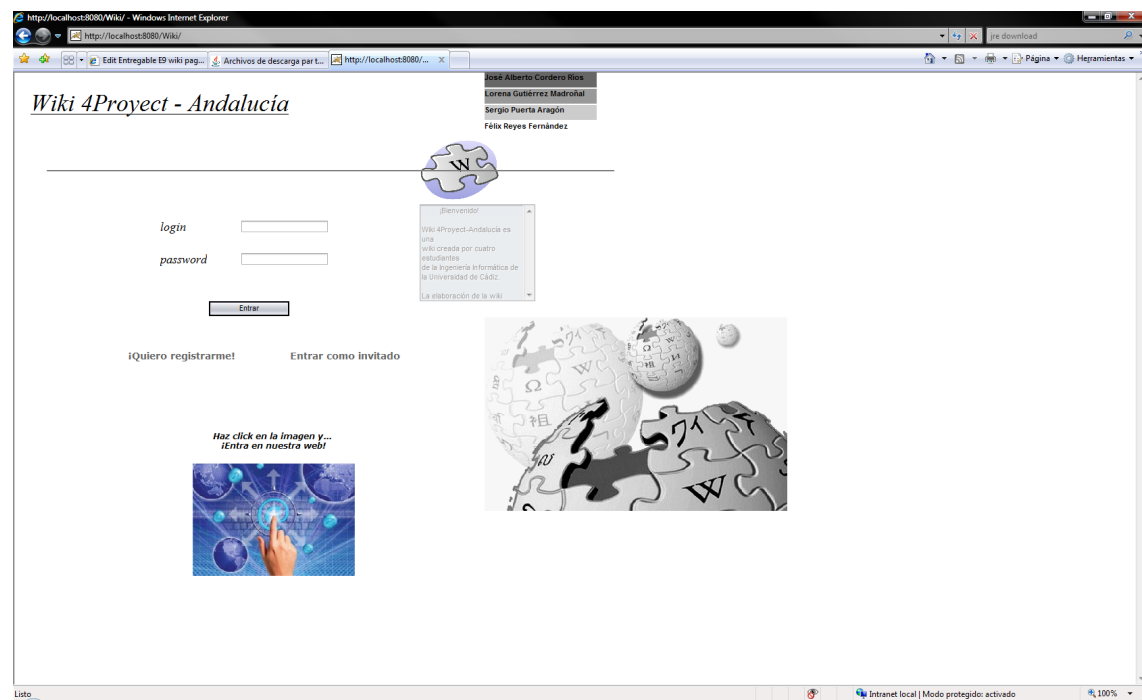


Figura 9.19: Instalación Tomcat paso 9

¡A wikear!