

El cuaderno del astrónomo

Manuel Luis Ortiz Rojas

ETIG

José Juan Rodríguez

18/06/2012

A mis padres, por innumerables motivos.

Resumen

El presente Trabajo Fin de Carrera (TFC) tiene como objetivo el diseño y la implementación de una aplicación web basada en la plataforma J2EE que sirva como herramienta de trabajo al astrónomo aficionado.

En líneas generales, la aplicación deberá almacenar información relativa a un número de entidades relacionadas con la acción de realizar una observación astronómica o la realización de astrofotografía. Hablamos de material de observación como telescopios, oculares, monturas o cámaras. No menos importante es el lugar de observación, por lo que la aplicación será capaz de guardar información precisa y visual que facilite su gestión. Por último, como parte de una observación, la aplicación mantendrá un catálogo de objetos astronómicos con información de utilidad, como pueden ser planetas, galaxias, nebulosas, etc.

Como novedad, y enfocando el proyecto al boom de las redes sociales, se ha querido añadir una funcionalidad básica pero indispensable, como es el tratamiento de las peticiones de amistad y su aceptación y una consulta básica de las observaciones realizadas por tus amigos.

Para llegar al objetivo se ha prestado especial interés a la planificación y a la fase de diseño, realizando la codificación haciendo uso de JBoss Seam como framework.

Índice

1. Introducción	6
1.1. Justificación del TFC y contexto en el que se desarrolla.....	6
1.2. Objetivos del TFC.....	6
1.3. Enfoque y metodologías seguidas.....	7
1.4. Planificación del proyecto.....	8
1.4.1.1. Calendario de entregas.....	9
1.4.1.2. Organización de tareas.....	10
1.4.1.3. Diagrama de Gantt.....	11
1.5. Productos obtenidos.....	11
2. Análisis	13
2.1. Actores.....	13
2.2. Casos de uso por subsistema.....	14
2.2.1. Subsistema de registro.....	14
2.2.2. Subsistema de gestión de conexión.....	16
2.2.3. Subsistema de gestión de equipo.....	19
2.2.4. Subsistema de gestión de catálogo.....	23
2.2.5. Subsistema de gestión de lugares.....	27
2.2.6. Subsistema de gestión de observaciones.....	31
2.2.7. Subsistema de gestión de amigos.....	35
3. Diseño	39
3.1. Arquitectura.....	39
3.2. JBoss seam, un framework para unirlos a todos.....	40
3.2.1.1. JBoss Seam y EJB3.....	41
3.2.1.2. JBoss Seam y JSF.....	41
3.2.1.3. JBoss Seam y la seguridad.....	42
3.3. Entorno de ejecución.....	42
3.4. Herramientas de desarrollo.....	43
3.5. Diagrama de clases.....	44
3.6. Modelo de datos.....	45
4. Implementación	46
4.1. Creación de un nuevo proyecto y seam-gen.....	46

4.2.	Persistencia de entidades.....	48
4.3.	Listado de entidades.....	51
4.4.	Vista en XHTML	52
4.5.	Control de permisos y flujo de páginas	54
5.	Valoración económica.....	55
6.	Conclusiones	56
7.	Glosario de términos.....	57
8.	Bibliografía.....	59

1. Introducción

1.1. Justificación del TFC y contexto en el que se desarrolla.

El objetivo principal del TFC, y de este proyecto en concreto, es enfrentarnos a un problema real desde cero, pasando desde un análisis de requisitos, diseño, implementación, pruebas y entrega al cliente (en este caso nos faltaría mantenimiento).

De esta forma podemos afianzar conocimientos en asignaturas como:

- Base de datos 1 y 2.
- Ingeniería del software.
- Programación orientada a objetos.
- Estructura de la información.
- Técnicas de desarrollo del software.

La razón de elegir J2EE viene motivada por mi carrera profesional y por la necesidad de ampliar conocimientos. Y aquí entra JBoss Seam, framework que goza de una gran popularidad ya que cuenta con el soporte de Red Hat, es muy productivo gracias a la utilidad de *seam-gen*.

1.2. Objetivos del TFC

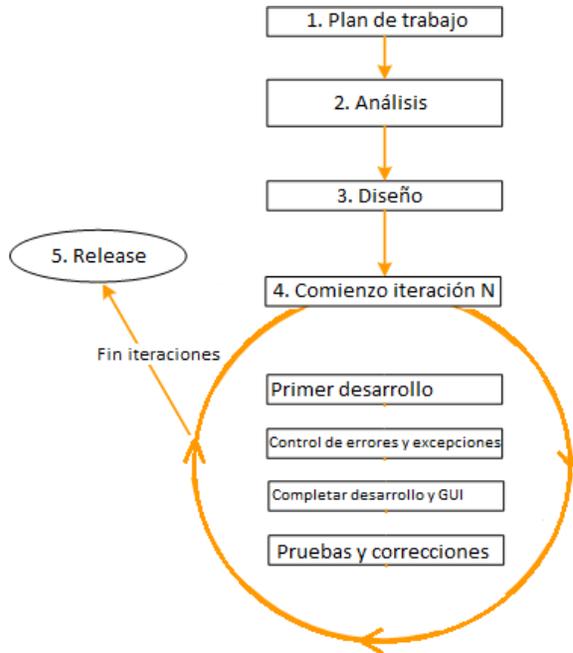
El objetivo del trabajo fin de carrera es proporcionar a la comunidad de astrónomos aficionados presente en la Red de una herramienta útil para la gestión de observaciones, así como fomentar el intercambio de conocimientos a través del aspecto social.

Para garantizar el tiempo y la calidad del desarrollo se establecen una serie de entregas al cliente (en este caso la *Universitat Oberta de Catalunya*):

- **PAC1:** Plan de Trabajo. Con el objetivo de establecer una descripción y una planificación realista del proyecto a desarrollar.
- **PAC2:** Análisis y diseño. Documentación necesaria para sentar la base de implementación del proyecto.
- **PAC3:** entrega de la aplicación a desarrollar y manual de instalación.
- **Memoria:** documento de síntesis final.
- **Presentación:** resumen en forma de presentación del trabajo realizado.

1.3. Enfoque y metodologías seguidas

El ciclo de vida que seguirá el proyecto será el desarrollo iterativo. Cada caso de uso pasará por varias iteraciones hasta llegar a la versión final.



Una vez aprobado el análisis y el diseño, pasaríamos al desarrollo de cada caso de uso.

Ejemplo de ciclo de vida iterativo sobre el primer caso de uso: registro de usuario.

- **Iteración 1:** implementación del caso de uso (registro de usuario) sin tener en cuenta validaciones, estilos. Se valorará la realización de la conexión con la base de datos y la creación del registro en la tabla correspondiente.
- **Iteración 2:** control de errores, excepciones y validaciones. Se controlará que se introducen los campos obligatorios, formato de dichos campos, uso de librerías opcionales para la presentación.
- **Iteración 3:** completar el caso de uso con todos los campos necesarios, mejorar la gestión de errores y usabilidad.
- **Iteración 4:** realizamos las pruebas necesarias y se acepta el caso de uso.

Finalizado el caso de uso, pasaríamos al siguiente; en este caso: autenticación y autorización, siguiendo el mismo procedimiento anteriormente descrito.

De esta forma nos acercamos más a metodologías ágiles como SCRUM para poder ajustarnos a las necesidades reales del proyecto; ya que en cada iteración se realiza una entrega al cliente y podemos ajustar el desarrollo a tiempo.

1.4. Planificación del proyecto

Para una correcta planificación, vamos a descomponer el desarrollo en distintas tareas. Por experiencia propia, esta es una de las fases donde más errores se suele cometer, ya que normalmente se realizan estimaciones en ambientes positivos y bajo presión y no se tienen en cuenta factores como: formación del personal, cambios en los requisitos por parte del cliente, elección del framework adecuado.

El rol de cliente, analista y desarrollador será realizado por una misma persona, aunque nos referiremos indiferentemente como si se trataran de personas distintas.

- **Plan de trabajo y formación.**

 - Elaboración del plan de trabajo.
 - Formación en las tecnologías necesarias.

- **Análisis funcional**

 - Requisitos funcionales.
 - Requisitos no funcionales.
 - Casos de uso.

- **Diseño**

 - Diagrama entidad-relación.
 - Diagrama de clases.
 - Diagrama de secuencia.
 - Descripción de la arquitectura.

- **Implementación**

 - Desarrollo de la aplicación siguiendo un ciclo de vida iterativo e incremental. Incluye la realización de las pruebas.

 - Manual de instalación.

- **Entrega**

 - Envío de los fuentes y binarios necesarios al consultor.

Memoria

Documento descriptivo acerca de la realización de este proyecto.
Presentación.

1.4.1.1. Calendario de entregas

Para el cumplimiento de la evaluación continua y un correcto seguimiento del trabajo fin de carrera; la UOC establece una serie de fechas para la consecución de cada hito que se detallan a continuación:

PAC 1 - Plan de trabajo	14 de Marzo de 2012
PAC 2 - Análisis y diseño	19 de Abril de 2012
PAC 3 - Implementación	4 de Junio de 2012
Memoria y presentación	18 de Junio de 2012

El cumplimiento de estos plazos es obligatorio, por lo que siempre se priorizará la entrega de los documentos y/o fuentes necesarios sobre cualquier tarea que se describe en el siguiente apartado.

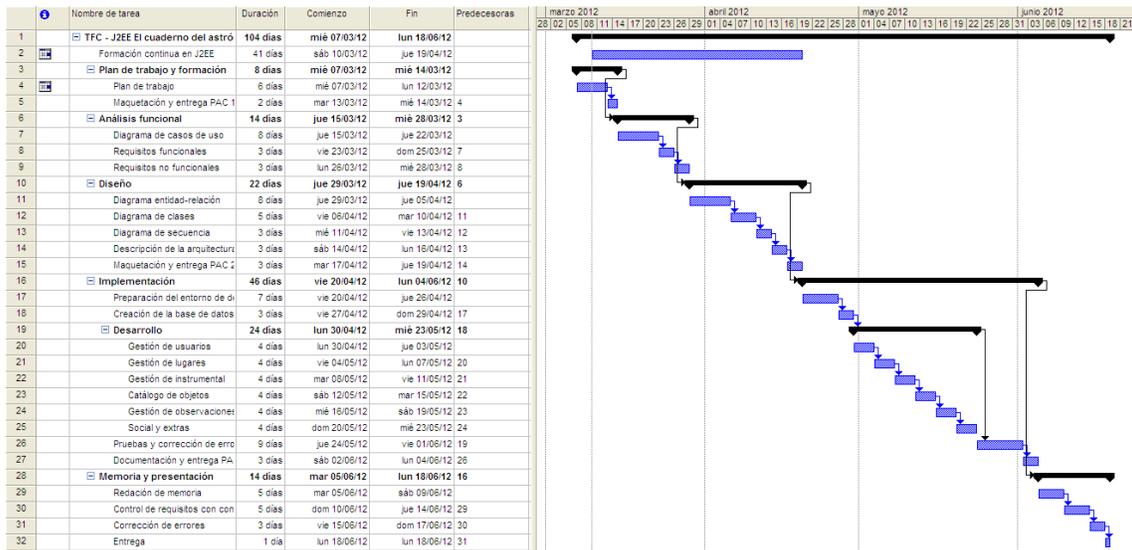
1.4.1.2. Organización de tareas

Se realizará una dedicación de 7 días a la semana, con aproximadamente 4 horas por jornada durante la tarde/noche.

		Nombre de tarea	Duración	Comienzo	Fin
1		 TFC - J2EE El cuaderno del astrónomo	104 días	mié 07/03/12	lun 18/06/12
2		Formación continua en J2EE	41 días	sáb 10/03/12	jue 19/04/12
3		 Plan de trabajo y formación	8 días	mié 07/03/12	mié 14/03/12
4		Plan de trabajo	6 días	mié 07/03/12	lun 12/03/12
5		Maquetación y entrega PAC 1	2 días	mar 13/03/12	mié 14/03/12
6		 Análisis funcional	14 días	jue 15/03/12	mié 28/03/12
7		Diagrama de casos de uso	8 días	jue 15/03/12	jue 22/03/12
8		Requisitos funcionales	3 días	vie 23/03/12	dom 25/03/12
9		Requisitos no funcionales	3 días	lun 26/03/12	mié 28/03/12
10		 Diseño	22 días	jue 29/03/12	jue 19/04/12
11		Diagrama entidad-relación	8 días	jue 29/03/12	jue 05/04/12
12		Diagrama de clases	5 días	vie 06/04/12	mar 10/04/12
13		Diagrama de secuencia	3 días	mié 11/04/12	vie 13/04/12
14		Descripción de la arquitectura	3 días	sáb 14/04/12	lun 16/04/12
15		Maquetación y entrega PAC 2	3 días	mar 17/04/12	jue 19/04/12
16		 Implementación	46 días	vie 20/04/12	lun 04/06/12
17		Preparación del entorno de desarrollo	7 días	vie 20/04/12	jue 26/04/12
18		Creación de la base de datos	3 días	vie 27/04/12	dom 29/04/12
19		 Desarrollo	24 días	lun 30/04/12	mié 23/05/12
20		Gestión de usuarios	4 días	lun 30/04/12	jue 03/05/12
21		Gestión de lugares	4 días	vie 04/05/12	lun 07/05/12
22		Gestión de instrumental	4 días	mar 08/05/12	vie 11/05/12
23		Catálogo de objetos	4 días	sáb 12/05/12	mar 15/05/12
24		Gestión de observaciones	4 días	mié 16/05/12	sáb 19/05/12
25		Social y extras	4 días	dom 20/05/12	mié 23/05/12
26		Pruebas y corrección de errores	9 días	jue 24/05/12	vie 01/06/12
27		Documentación y entrega PAC 3	3 días	sáb 02/06/12	lun 04/06/12
28		 Memoria y presentación	14 días	mar 05/06/12	lun 18/06/12
29		Redacción de memoria	5 días	mar 05/06/12	sáb 09/06/12
30		Control de requisitos con conector	5 días	dom 10/06/12	jue 14/06/12
31		Corrección de errores	3 días	vie 15/06/12	dom 17/06/12
32		Entrega	1 día	lun 18/06/12	lun 18/06/12

Durante la finalización de las distintas tareas se irán realizando revisiones con el consultor de forma asíncrona para verificar el trabajo realizado.

1.4.1.3. Diagrama de Gantt



1.5. Productos obtenidos

Tras realizar todo el proceso del Trabajo Fin de Carrera, se han obtenido una serie de productos que se corresponden con los entregables de cada PAC:

- **Plan de trabajo:** en esta fase se han establecido los objetivos que deberá cumplir el proyecto, así como una planificación de todo el trabajo a realizar teniendo en cuenta los recursos disponibles, formación y tiempo.
- **Análisis y diseño,** en este documento encontraremos la siguiente información:
 - Casos de uso por subsistema, como base del análisis. En este punto encontraremos la información relativa al funcionamiento que deberá cumplir la aplicación.
 - Diseño y arquitectura. Descripción del framework a utilizar, diagrama de clases y diseño de la base de datos.
 - Prototipo gráfico. El cual dará al cliente una idea aproximada de la distribución de los elementos en las pantallas más significativas,

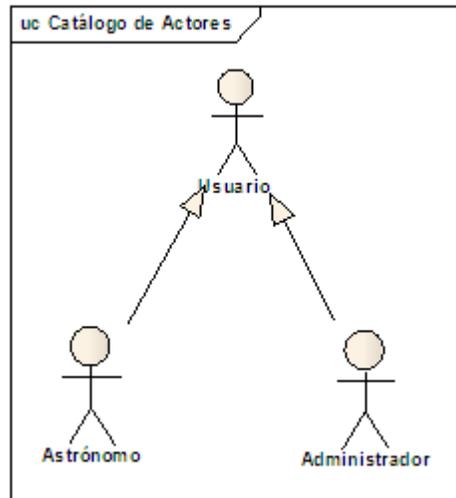
- **Implementación**, fase en la que se ha desarrollado el producto y además se ha generado la documentación para su correcta instalación y puesta en funcionamiento. La entrega consta principalmente de un fichero EAR y el script de base de datos.

- **Entrega final y memoria:**
 - Producto final: entrega definitiva del producto tal y como se ha descrito en el punto anterior.
 - Memoria: formada por este documento el cual sintetiza toda la información generada durante el transcurso de cada una de las fases de este trabajo.
 - Presentación: documento en forma de diapositivas el cual sintetizará todo el trabajo realizado.

2. Análisis

Este apartado recoge la descripción de los actores y de cada una de las funcionalidades de la aplicación a través de los casos de uso.

2.1. Actores



Catálogo de Actores

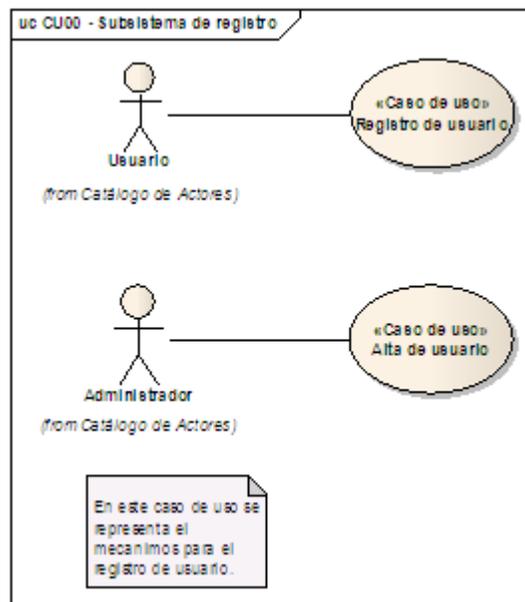
Nombre	Usuario
Versión	1.0
Dependencias	Astrónomo -- Usuario Usuario -- Registro de usuario Usuario -- Entrar Administrador -- Usuario Usuario -- Listar catálogo astronómico
Descripción	Representa de forma genérica a cualquier tipo de usuario de la aplicación. Este usuario sólo existe a nivel teórico y representa indistintamente a cualquier usuario que herede de él.
Importancia	Baja
Prioridad	Media
Estado	Aprobado
Comentarios	
Nombre	Administrador
Versión	1.0
Dependencias	Administrador -- Editar elemento astronómico Administrador -- Usuario Administrador -- Alta de usuario

Descripción	Representa a los administradores de la aplicación.
Importancia	Media
Prioridad	Media
Estado	Aprobado
Comentarios	

Nombre	Astrónomo
Versión	1.0
Dependencias	Astrónomo -- Usuario Astrónomo -- Listar lugares de observación Astrónomo -- Listar catálogo astronómico Astrónomo -- Listar equipo Astrónomo -- Listar observaciones Astrónomo -- Amistades pendientes Astrónomo -- Listar amigos
Descripción	Representa a los usuarios que se registrarán para poder hacer uso de la aplicación.
Importancia	Alta
Prioridad	Alta
Estado	Aprobado

2.2. Casos de uso por subsistema

2.2.1. Subsistema de registro



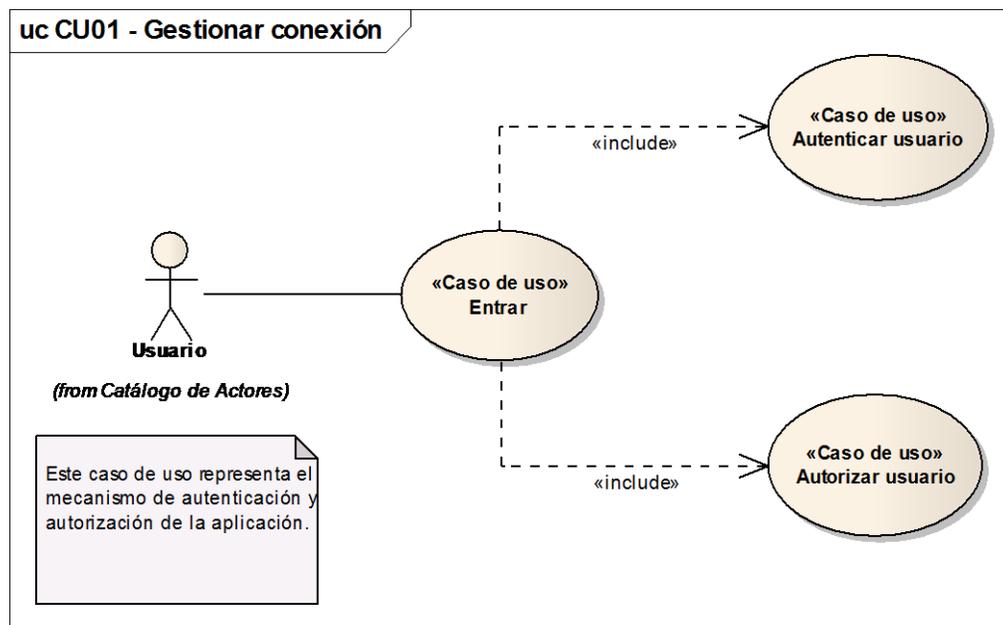
CU00 - Subsistema de registro

Nombre	Alta de usuario
Versión	1.0
Dependencias	Administrador -- Alta de usuario
Precondición/ postcondición	Precondición: El usuario debe de tener perfil administrador y estar autenticado en el sistema. El usuario a crear no debe de existir. Postcondición: El usuario queda almacenado en la base de datos.
Subsistema asociado	Subsistema de registro
Descripción	El sistema permitirá al usuario administrador dar de alta otros usuarios con cualquier perfil (Astrónomo o Administrador)
Escenarios	Basic Path - Alta de usuario 1. El usuario administrador selecciona la opción "Usuarios" => "Alta de usuario" del menú principal. 2. El sistema muestra un formulario con los campos que se muestran en el caso de uso "Registro de usuario" y además con la posibilidad de elegir el perfil (Astrónomo o Administrador) 3. El usuario introduce los datos requeridos y pulsa guardar. 4. El sistema almacena el usuario en la base de datos.
Importancia	Media
Prioridad	Media
Estado	Aprobado
Comentarios	

Nombre	Registro de usuario
Versión	1.0
Dependencias	Usuario -- Registro de usuario
Precondición/ postcondición	Precondición: El usuario no puede existir en la base de datos Postcondición: El usuario queda registrado en la base de datos con el perfil de "Astrónomo".
Subsistema asociado	Subsistema de registro
Descripción	El caso de uso recoge los pasos a seguir para que un usuario externo al sistema pueda registrarse como usuario con perfil astrónomo.
Escenarios	Basic Path - Registro de usuario 1. El usuario pulsa la opción registro.

	<p>2. El sistema muestra un formulario con los datos requeridos: Nombre de usuario (único y obligatorio) Contraseña (obligatorio) Nombre y apellidos (obligatorio) Correo electrónico (obligatorio) Fecha de nacimiento Sexo</p> <p>3. El usuario introduce al menos los datos obligatorios y pulsa aceptar.</p> <p>4. El sistema almacena los datos y muestra un mensaje al usuario con el resultado de la operación.</p>
Importancia	Alta
Prioridad	Alta
Estado	Aprobado
Comentarios	

2.2.2. Subsistema de gestión de conexión



CU01 - Gestionar conexión

Nombre	Entrar
Versión	1.0
Dependencias	Listar catálogo astronómico -- Entrar Entrar -- RF1.1 - Usuario y contraseña

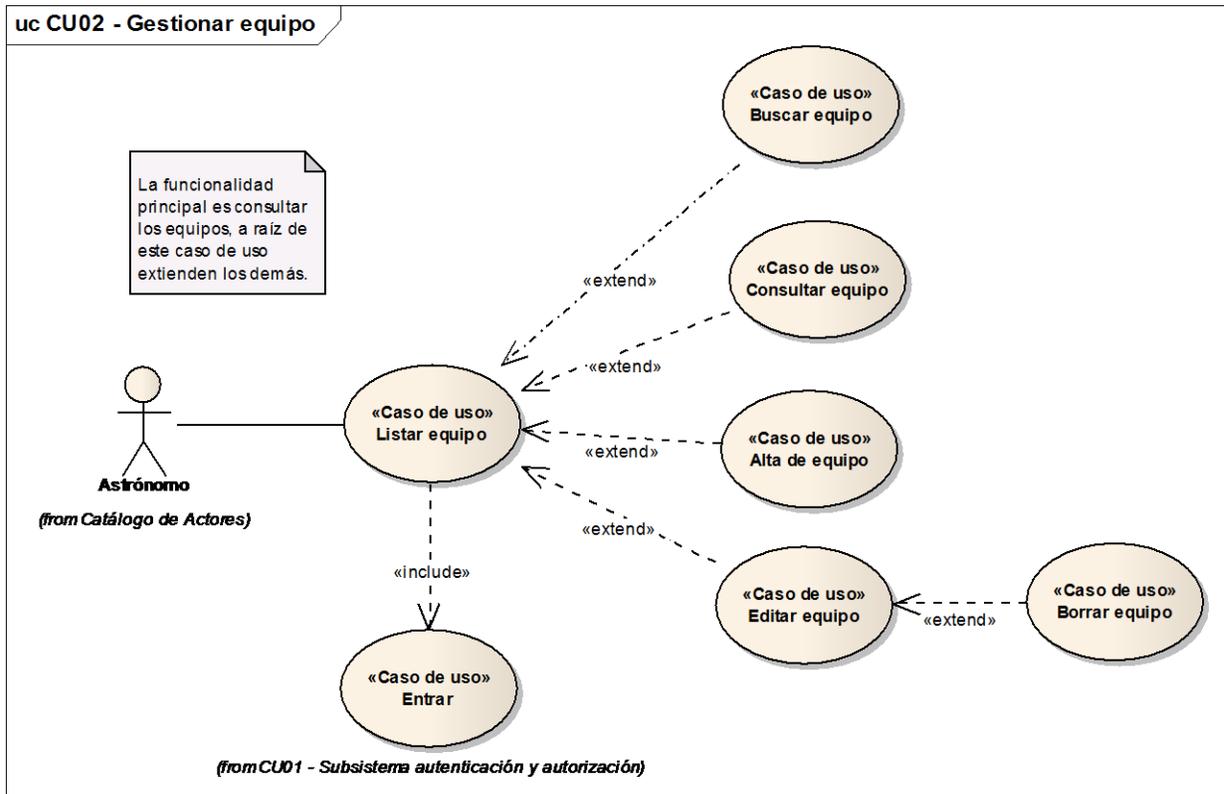
	<p>Listar equipo -- Entrar</p> <p>Usuario -- Entrar</p> <p>Entrar -- Listar observaciones</p> <p>Amistades pendientes -- Entrar</p> <p>Listar amigos -- Entrar</p> <p>Entrar -- RNF3.1 - Estándares web</p> <p>Entrar -- Autorizar usuario</p> <p>Entrar -- Autenticar usuario</p> <p>Entrar -- RF1.2 - Usuario y rol</p> <p>Listar lugares de observación -- Entrar</p>
Precondición/ postcondición	<p>Precondición: El usuario debe acceder a la aplicación desde la intranet de la Consejería de Salud o desde internet.</p> <p>Postcondición: El usuario accede a la aplicación si sus credenciales son válidas.</p> <p>Obtenemos los roles del usuario.</p>
Subsistema asociado	Subsistema de gestión de conexión
Descripción	El sistema sólo permitirá el acceso a la aplicación a los usuarios válidos y éstos sólo podrán acceder a las secciones a las que tienen permiso.
Escenarios	<p>Basic Path - Entrar</p> <ol style="list-style-type: none"> 1. El usuario introduce la dirección web de la aplicación en un navegador. 2. La aplicación muestra una ventana donde se le pregunta por sus credenciales (usuario y contraseña). 3. El usuario introduce sus credenciales y pulsa "Entrar". 4. La aplicación comprueba que las credenciales son correctas y se comprueban los permisos que tiene habilitados el usuario.
Importancia	Alta
Prioridad	Alta
Estado	Aprobado

Nombre	Autenticar usuario
Versión	1.0
Dependencias	<p>Entrar -- Autenticar usuario</p> <p>Autenticar usuario -- RF1.2 - Usuario y rol</p>
Precondición/ postcondición	<p>Precondición: El usuario debe estar dado de alta en la base de datos y tener concedidos permisos en la aplicación.</p> <p>Postcondición: El usuario es autenticado y se comprueban los permisos que tiene para interactuar con la aplicación.</p>
Subsistema asociado	Subsistema de gestión de conexión
Descripción	Se comprueba si las credenciales del usuario son correctas en la base de

	datos
Escenarios	Basic Path - Autenticar usuario 1. La aplicación obtiene el usuario relacionado con los datos introducidos. 2. La aplicación contrasta el usuario obtenido y la contraseña con el usuario obtenido de la base de datos.
Importancia	Alta
Prioridad	Alta
Estado	Aprobado

Nombre	Autorizar usuario
Versión	1.0
Dependencias	Autorizar usuario -- RF2.1 - Acceso a la aplicación Entrar -- Autorizar usuario
Precondición/ postcondición	Precondición: El usuario debe haber sido autenticado previamente y debe de estar dado de alta en el módulo de autorización de la aplicación. Postcondición: Se recuperan los permisos asignados al usuario.
Subsistema asociado	Subsistema de gestión de conexión
Descripción	Se comprueban los permisos que tiene habilitados el usuario.
Escenarios	Basic Path - Autorizar usuario 1. Se busca el usuario en el módulo de autorización de la aplicación y se recupera el permiso que tiene asignado. 2. El sistema asigna el perfil del usuario a la sesión abierta en la aplicación.
Importancia	Alta
Prioridad	Alta
Estado	Aprobado

2.2.3. Subsistema de gestión de equipo



CU02 - Gestionar equipo

Nombre	Listar equipo
Versión	1.0
Dependencias	Alta de equipo -- Listar equipo Listar equipo -- Entrar Editar equipo -- Listar equipo Buscar equipo -- Listar equipo Consultar equipo -- Listar equipo Astrónomo -- Listar equipo
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema muestra el equipo del usuario, la opción de dar de alta, un filtro de búsqueda, la opción de editar un elemento y otra para consultarlo.
Subsistema asociado	Subsistema de gestión de equipo
Descripción	El sistema permitirá al usuario astrónomo consultar en forma de listado, su equipo dado de alta en el sistema.
Escenarios	Basic Path - Consulta de equipo 1. El usuario selecciona la opción "Equipo" del menú principal.

	<p>2. El sistema consulta en la base de datos los instrumentos perteneciente al usuario actual.</p> <p>3. El sistema muestra un listado con dichos instrumentos.</p>
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Buscar equipo
Versión	1.0
Dependencias	Buscar equipo -- Listar equipo
Precondición/ postcondición	<p>Precondición: El usuario debe de estar autenticado y autorizado.</p> <p>Postcondición: El sistema muestra el equipo que cumple el criterio de la búsqueda</p>
Subsistema asociado	Subsistema de gestión de equipo
Descripción	El sistema permitirá la búsqueda del equipo del usuario autenticado.
Escenarios	<p>Basic Path - Buscar equipo</p> <p>1. El sistema muestra los siguientes filtros de búsqueda en la cabecera del caso de uso "Consultar equipo":</p> <p>Combo clasificación</p> <p>Marca</p> <p>Modelo</p> <p>Filtro AND/OR para los campos de búsqueda.</p> <p>2. El usuario introduce el criterio de búsqueda y pulsa en el botón "Buscar".</p> <p>3. El sistema muestra el resultado.</p>
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Consultar equipo
Versión	1.0
Dependencias	Consultar equipo -- Listar equipo
Precondición/ postcondición	<p>Precondición: El usuario debe de estar autenticado y autorizado.</p> <p>Deberá de existir algún elemento de equipo.</p> <p>Postcondición: El sistema muestra la información relacionada con el equipo seleccionado.</p>
Subsistema asociado	Subsistema de gestión de equipo
Descripción	El sistema permitirá consultar un elemento del equipo.

Escenarios	Basic Path - Consultar equipo 1. El usuario selecciona la opción "Ver" del listado del equipo. 2. El sistema muestra la información completa del equipo seleccionado.
Importancia	Media
Prioridad	Media
Estado	Aprobado

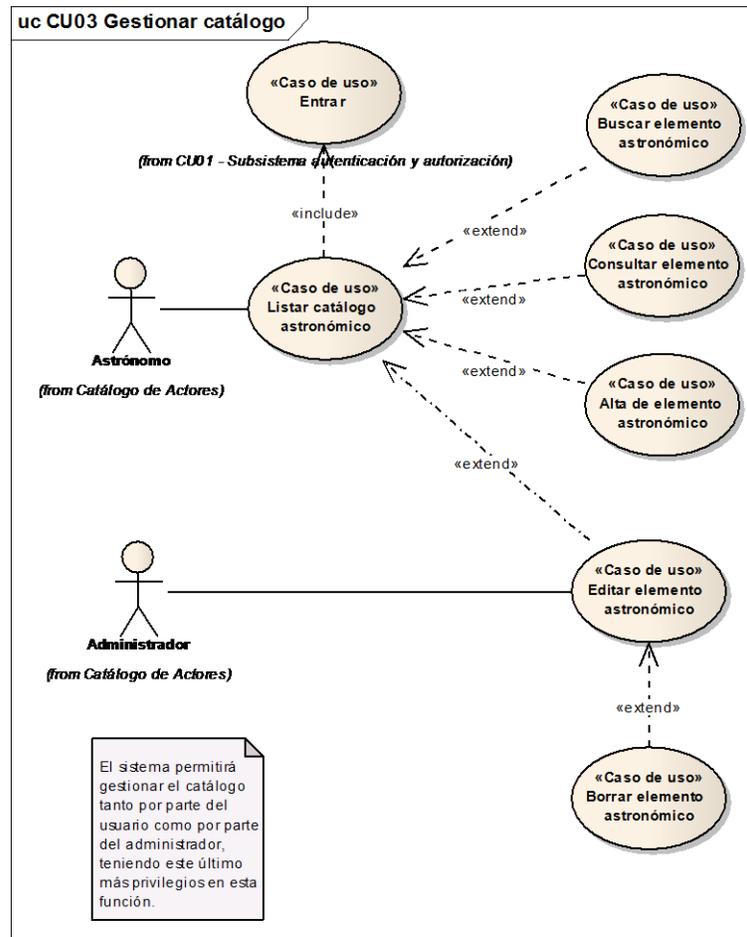
Nombre	Alta de equipo
Versión	1.0
Dependencias	Alta de equipo -- Listar equipo
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: Queda almacenado el nuevo equipo en el inventario.
Subsistema asociado	Subsistema de gestión de equipo
Descripción	El sistema permitirá el alta de equipo al inventario del usuario astrónomo.
Escenarios	Basic Path - Alta de equipo 1. El usuario pulsa el botón de "Nuevo Equipo" desde la pantalla del caso de uso "Consultar equipo". 2. El sistema muestra un formulario con los siguientes campos: <ul style="list-style-type: none"> • Clasificación (combo de valores - obligatorio) • Marca del equipo (obligatorio) • Modelo del equipo (obligatorio) • Comentario
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Editar equipo
Versión	1.0
Dependencias	Borrar equipo -- Editar equipo Editar equipo -- Listar equipo
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir algún ítem en el equipo del usuario. Postcondición: El ítem queda modificado.
Subsistema asociado	Subsistema de gestión de equipo
Descripción	El sistema permitirá modificar la información relativa a cualquier de los ítems de su equipo.
Escenarios	Basic Path - Editar equipo 1. El usuario selecciona la opción "Editar" de uno de los ítems de la lista

	<p>resultante del caso de uso "Consultar equipo".</p> <ol style="list-style-type: none"> 2. El sistema muestra el ítem del equipo con sus valores precargados. 3. El usuario realiza las modificaciones oportunas y pulsa el botón "Guardar". 4. El sistema guarda los cambios en base de datos y muestra al usuario el resultado de la operación.
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Borrar equipo
Versión	1.0
Dependencias	Borrar equipo -- Editar equipo
Precondición/ postcondición	<p>Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir algún elemento en el equipo del usuario.</p> <p>Postcondición: El ítem queda eliminado si aún no se ha realizado ninguna observación con él. En caso de haber sido usado en la base de datos para alguna observación, será dado de baja y no se mostrará en el equipo del usuario (pero sí en las observaciones donde se haya usado).</p>
Subsistema asociado	Subsistema de gestión de equipo
Descripción	El sistema permitirá dar de borrar un elemento del equipo del usuario.
Escenarios	<p>Basic Path - Borrar equipo</p> <ol style="list-style-type: none"> 1. El usuario ejecuta el caso de uso "Editar equipo". 2. El usuario selecciona la opción "Borrar". 3. El sistema elimina o da de baja de la base de datos el equipo seleccionado.
Importancia	Media
Prioridad	Media
Estado	Aprobado

2.2.4. Subsistema de gestión de catálogo



CU03 Gestionar catálogo

Nombre	Listar catálogo astronómico
Versión	1.0
Dependencias	Listar catálogo astronómico -- Entrar Buscar elemento astronómico -- Listar catálogo astronómico Editar elemento astronómico -- Listar catálogo astronómico Alta de elemento astronómico -- Listar catálogo astronómico Consultar elemento astronómico -- Listar catálogo astronómico Astrónomo -- Listar catálogo astronómico Usuario -- Listar catálogo astronómico
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema muestra un listado de los elementos del catálogo astronómico
Subsistema asociado	Subsistema de gestión de catálogo
Descripción	El sistema permitirá consultar el catálogo astronómico tanto al usuario

	administrador como al usuario astrónomo.
Escenarios	Basic Path - Listar catálogo astronómico 1. El usuario selecciona la opción "Catálogo" del menú principal. 2. El sistema muestra un listado con todos los elementos del sistema, un filtro de búsqueda, la opción alta y la opción editar si el usuario es administrador.
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Alta de elemento astronómico
Versión	1.0
Dependencias	Alta de elemento astronómico -- Listar catálogo astronómico
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El elemento queda almacenado en la base de datos disponible de forma pública.
Subsistema asociado	Subsistema de gestión de catálogo
Descripción	El sistema permitirá al usuario dar de alta nuevos elementos en el catálogo.
Escenarios	Basic Path - Alta de elemento astronómico 1. El usuario pulsa la opción "Alta" del listado del catálogo astronómico. 2. El sistema muestra un formulario con los siguientes campos: <ul style="list-style-type: none"> • Nombre (obligatorio) • Clasificación (combo - obligatorio) • Magnitud aparente (opcional) • Tamaño aparente (opcional) • Messier (opcional) • NGC (opcional) • Imagen (opcional) • Descripción (opcional) 3. El usuario introduce al menos los campos obligatorios. 4. El sistema almacena la información e informa al usuario del resultado.
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Buscar elemento astronómico
Versión	1.0
Dependencias	Buscar elemento astronómico -- Listar catálogo astronómico

Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema muestra los elementos que cumplen el criterio de la búsqueda.
Subsistema asociado	Subsistema de gestión de catálogo
Descripción	El sistema permitirá buscar elementos del catálogo astronómico.
Escenarios	Basic Path - Buscar elemento astronómico 1. El usuario introduce el criterio de búsqueda en el filtro de búsqueda mostrado en el caso de uso "Listar catálogo astronómico". <ul style="list-style-type: none"> • Nombre • Clasificación • Campo AND/OR para los campos de búsqueda 2. El sistema realiza la búsqueda según los valores introducidos. 3. El sistema muestra el resultado de la búsqueda.
Importancia	Media
Prioridad	Media
Estado	Aprobado

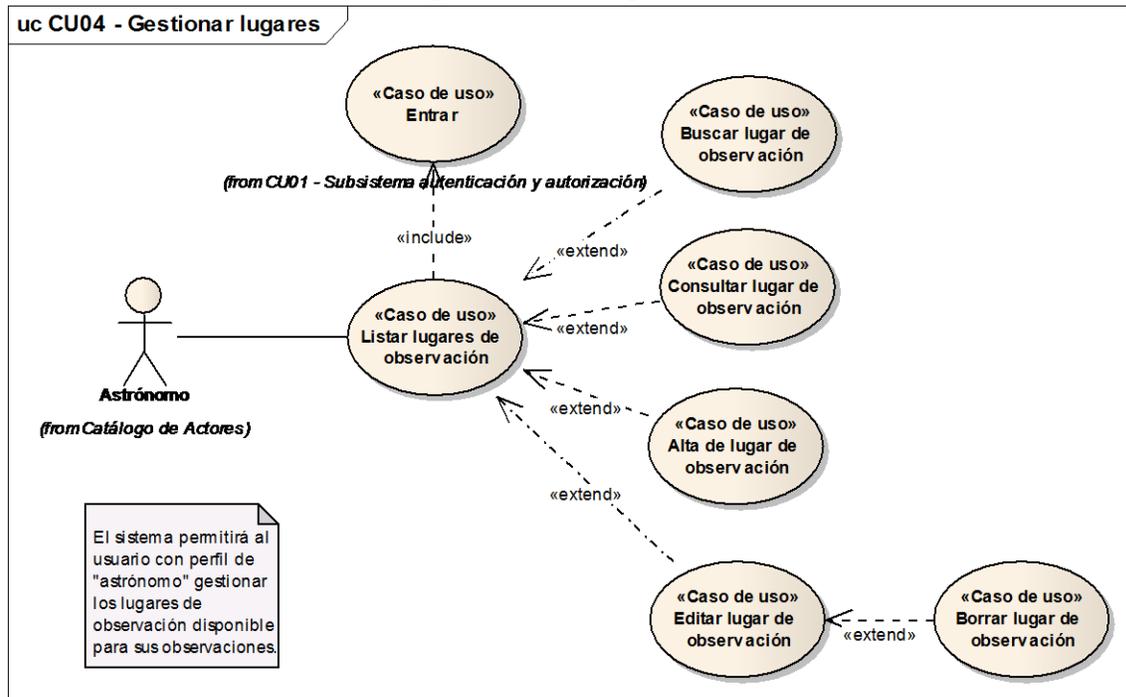
Nombre	Consultar elemento astronómico
Versión	1.0
Dependencias	Consultar elemento astronómico -- Listar catálogo astronómico
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir al menos un elemento en el catálogo astronómico. Postcondición: El sistema muestra la información completa relacionada con el elemento seleccionado.
Subsistema asociado	Subsistema de gestión de catálogo
Descripción	El sistema permitirá consultar los detalles de un elemento disponible en el catálogo astronómico de la aplicación.
Escenarios	Basic Path - Consultar elemento astronómico 1. El usuario pulsa la opción "Ver" relativa a un elemento del catálogo astronómico. 2. El sistema muestra la información relativa al elemento seleccionado.
Importancia	Media
Prioridad	Media
Estado	Aprobado
Nombre	Editar elemento astronómico
Versión	1.0
Dependencias	Administrador -- Editar elemento astronómico Editar elemento astronómico -- Listar catálogo astronómico

	Borrar elemento astronómico -- Editar elemento astronómico
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado como administrador. Debe de existir el elemento a editar. Postcondición: El elemento queda modificado en el sistema.
Subsistema asociado	Subsistema de gestión de catálogo
Descripción	El sistema permitirá al usuario administrador editar los elementos del catálogo astronómico.
Escenarios	Basic Path - Editar elemento astronómico 1. El usuario administrador selecciona la opción "editar" del listado del catálogo. 2. El sistema muestra un formulario con los campos descritos en el caso de uso "Alta de elemento astronómico". 3. El usuario introduce las modificaciones que considere oportunas y pulsa sobre la opción de "Guardar". 4. El sistema almacena los cambios y muestra el resultado de la operación.
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Borrar elemento astronómico
Versión	1.0
Dependencias	Borrar elemento astronómico -- Editar elemento astronómico
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado como administrador. Debe de existir algún elemento a eliminar. Postcondición: El elemento del catálogo queda eliminado del sistema en caso de no existir referencias a éste. En caso afirmativo, se le asignará una fecha de baja y ya no podrá consultarse en el catálogo.
Subsistema asociado	Subsistema de gestión de catálogo
Descripción	El sistema permitirá eliminar elementos del catálogo astronómico al usuario administrador.
Escenarios	Basic Path - Borrar elemento astronómico 1. El usuario ejecuta el caso de uso "Editar elemento astronómico" sobre el elemento que desea eliminar. 2. El usuario pulsa la opción "Borrar". 3. El sistema comprueba si existen referencias (observaciones) al objeto a

	eliminar.
	4. El sistema elimina el objeto en caso de no existir referencias en la base de datos; en caso de existir, se le asignará una fecha de baja.
Importancia	Media
Prioridad	Media
Estado	Aprobado

2.2.5. Subsistema de gestión de lugares



CU04 - Gestionar lugares

Nombre	Listar lugares de observación
Versión	1.0
Dependencias	Astrónomo -- Listar lugares de observación Buscar lugar de observación -- Listar lugares de observación Alta de lugar de observación -- Listar lugares de observación Editar lugar de observación -- Listar lugares de observación Consultar lugar de observación -- Listar lugares de observación Listar lugares de observación -- Entrar
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema mostrará el listado de lugares de observación del usuario astrónomo.
Subsistema asociado	Subsistema de gestión de lugares

Descripción	El sistema permitirá al usuario consultar sus lugares de observación.
Escenarios	Basic Path - Listar lugares de observación 1. El usuario selecciona la opción "Lugares" del menú principal. 2. El sistema muestra el listado de lugares de observación del usuario.
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Alta de lugar de observación
Versión	1.0
Dependencias	Alta de lugar de observación -- Listar lugares de observación
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema almacena en base de datos el lugar de observación.
Subsistema asociado	Subsistema de gestión de lugares
Descripción	El sistema permitirá dar de alta lugares de observación visibles sólo al usuario propietario.
Escenarios	Basic Path - Alta de lugar de observación 1. El usuario selecciona la opción "Nuevo lugar de observación". 2. El sistema muestra un formulario para introducir los campos del lugar de observación: <ul style="list-style-type: none"> • Nombre del lugar • Coordenadas (longitud y latitud) • Comentarios 3. El usuario introduce los datos y pulsa en "Guardar". 4. El sistema almacena el lugar e informa al usuario.
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Buscar lugar de observación
Versión	1.0
Dependencias	Buscar lugar de observación -- Listar lugares de observación
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema muestra los lugares de observación que cumplen el criterio de búsqueda.

Subsistema asociado	Subsistema de gestión de lugares
Descripción	El sistema permitirá al usuario astrónomo buscar lugares de observación en el listado propio de lugares de observación.
Escenarios	<p>Basic Path - Buscar lugar de observación</p> <p>1. El usuario introduce los valores en el formulario de búsqueda de la pantalla correspondiente al caso de uso: "Listar lugares de observación". Los campos disponibles son:</p> <ul style="list-style-type: none"> • Nombre <p>2. El sistema ejecuta la búsqueda y muestra los resultados que cumplen el criterio del usuario.</p>
Importancia	Media
Prioridad	Media
Estado	Aprobado

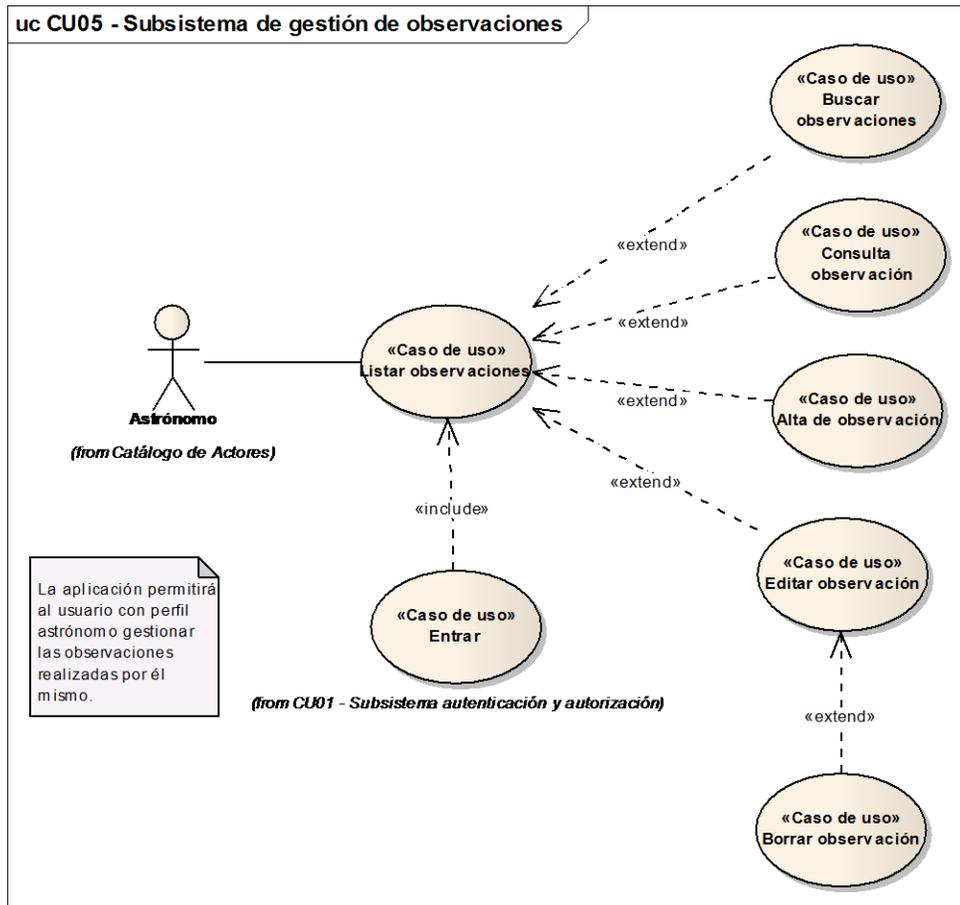
Nombre	Consultar lugar de observación
Versión	1.0
Dependencias	Consultar lugar de observación -- Listar lugares de observación
Precondición/postcondición	<p>Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir algún lugar de observación en el catálogo del usuario.</p> <p>Postcondición: El sistema muestra la información completa del lugar de observación del usuario. A partir de las coordenadas deberá pintar un mapa usando el servicio de Google Maps.</p>
Subsistema asociado	Subsistema de gestión de lugares
Descripción	El sistema permitirá al usuario consultar los detalles de sus lugares de observación.
Escenarios	<p>Basic Path - Consultar lugar de observación</p> <p>1. El usuario selecciona la opción "Ver" de uno de los elementos del listado de lugares de observación.</p> <p>2. El sistema muestra la información completa del lugar junto con un mapa de Google Maps.</p>
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Editar lugar de observación
Versión	1.0
Dependencias	<p>Borrar lugar de observación -- Editar lugar de observación</p> <p>Editar lugar de observación -- Listar lugares de observación</p>

Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir el lugar de observación a editar. Postcondición: El lugar de observación queda modificado en el sistema.
Subsistema asociado	Subsistema de gestión de lugares
Descripción	El sistema permitirá al usuario modificar sus propios lugares de observación.
Escenarios	Basic Path - Editar lugar de observación 1. El usuario selecciona la opción "Editar" del listado de lugares de observación. 2. El sistema muestra un formulario con los valores de la entidad tal y como se describe en el caso de uso "Alta de lugar de observación". 3. El usuario introduce los cambios que considera oportunos y pulsar "Guardar". 4. El sistema guarda los cambios y muestra al usuario el resultado de la operación.
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Borrar lugar de observación
Versión	1.0
Dependencias	Borrar lugar de observación -- Editar lugar de observación
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir el lugar a eliminar. Postcondición: El lugar de observación queda eliminado si no está referenciado por ninguna entidad del sistema o queda marcado con una fecha de baja en caso afirmativo. El lugar no se mostrará salvo en las observaciones que ya hayan sido registradas.
Subsistema asociado	Subsistema de gestión de lugares
Descripción	El sistema permitirá al usuario eliminar o dar de baja lugares de observación de su propio catálogo.
Escenarios	Basic Path - Borrar lugar de observación 1. El usuario astrónomo realiza el caso de uso "Editar lugar de observación" sobre el lugar de observación que desea eliminar. 2. El usuario pulsa la opción "Borrar". 3. El sistema borra o da de baja el lugar de observación.
Importancia	Media
Prioridad	Media
Estado	Aprobado

2.2.6. Subsistema de gestión de observaciones



CU05 - Subsistema de gestión de observaciones

Nombre	Listar observaciones
Versión	1.0
Dependencias	Consulta observación -- Listar observaciones Entrar -- Listar observaciones Buscar observaciones -- Listar observaciones Listar observaciones -- Consultar perfil de amigo Astrónomo -- Listar observaciones Editar observación -- Listar observaciones Alta de observación -- Listar observaciones
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema muestra todas las observaciones realizadas por el usuario.
Subsistema asociado	Subsistema de gestión de observaciones
Descripción	El sistema permitirá listar las observaciones realizadas por el usuario autenticado.

Escenarios	Basic Path - Listar observaciones 1. El usuario selecciona la opción "Observaciones" del menú principal. 2. El sistema consulta en la base de datos y muestra todas las observaciones realizadas por el usuario.
Importancia	Alta
Prioridad	Media
Estado	Aprobado

Nombre	Alta de observación
Versión	1.0
Dependencias	Alta de observación -- Listar observaciones
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir algún lugar de observación dado de alta por el usuario. Debe de existir el elemento observado en el catálogo astronómico. Postcondición: El sistema almacena la observación en la base de datos.
Subsistema asociado	Subsistema de gestión de observaciones
Descripción	El sistema permitirá al usuario dar de alta observaciones.
Escenarios	Basic Path - Alta de observación 1. El usuario selecciona el botón de "Nueva observación" desde la ventana del caso de uso "Listar observaciones". 2. El sistema muestra un formulario de alta de observación con los siguientes campos: <ul style="list-style-type: none"> • Objeto observado (obligatorio) • Lugar de observación (obligatorio) • Fecha de observación (puntual o rango - obligatorio) • Posición en el cielo (coordenadas altazimutales o ecuatoriales - opcional) • Comentario (opcional) • Ficheros adjuntos (imágenes y/o otro tipo de documentos - opcional)
Importancia	Alta
Prioridad	Media
Estado	Aprobado

Nombre	Buscar observaciones
Versión	1.0
Dependencias	Buscar observaciones -- Listar observaciones
Precondición/	Precondición: El usuario debe de estar autenticado y autorizado.

postcondición	Postcondición: El sistema muestra las observaciones que cumplen el criterio de búsqueda.
Subsistema asociado	Subsistema de gestión de observaciones
Descripción	El sistema permitirá realizar búsquedas sobre las observaciones realizadas por el usuario.
Escenarios	Basic Path – Buscar observaciones 1. El usuario introduce los criterios de búsqueda en el formulario de búsqueda de la pantalla "Listar observaciones" y pulsar "Buscar". 2. El sistema realiza la consulta y muestra los resultados que cumplen el criterio.
Importancia	Alta
Prioridad	Media
Estado	Aprobado

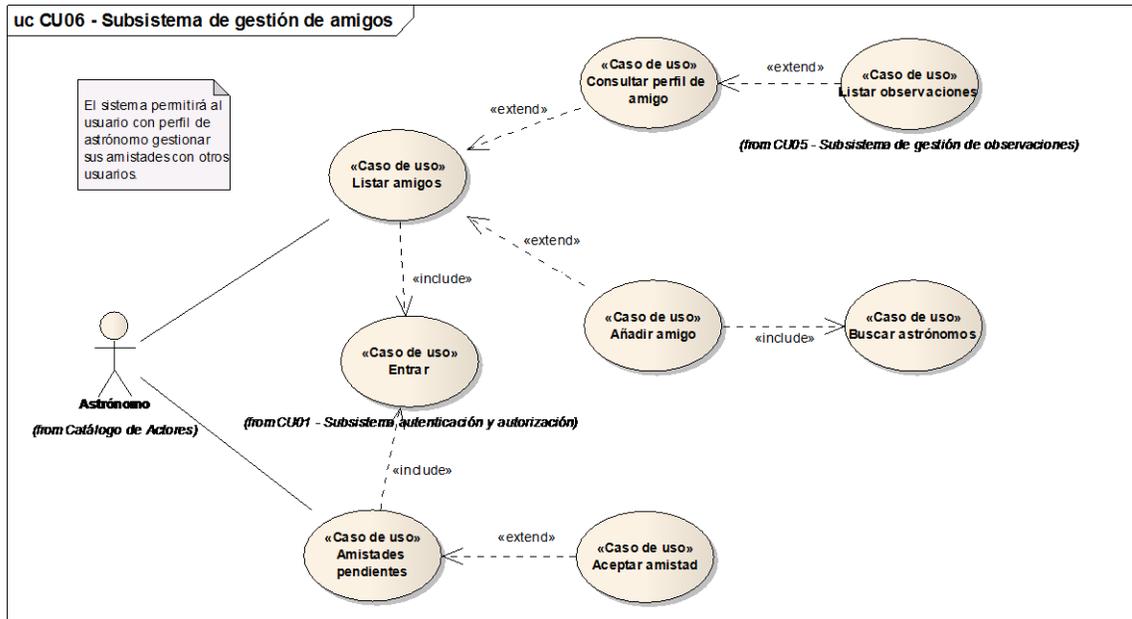
Nombre	Consulta observación
Versión	1.0
Dependencias	Consulta observación -- Listar observaciones
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir al menos una observación en el listado de observaciones del usuario. Postcondición: El sistema muestra la información detallada de la observación seleccionada.
Subsistema asociado	Subsistema de gestión de observaciones
Descripción	El sistema permitirá consultar la información detallada de cualquier de sus observaciones
Escenarios	Basic Path - Consulta observación 1. El usuario selecciona la opción "Ver" de una de las observaciones del listado de observaciones. 2. El sistema muestra la información detallada de la observación.
Importancia	Alta
Prioridad	Media
Estado	Aprobado

Nombre	Editar observación
Versión	1.0
Dependencias	Borrar observación -- Editar observación Editar observación -- Listar observaciones

Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir alguna observación en el listado de observaciones del usuario. Postcondición: La observación queda editada y guardada en la base de datos.
Subsistema asociado	Subsistema de gestión de observaciones
Descripción	El sistema permitirá al usuario editar los datos de sus observaciones.
Escenarios	Basic Path - Editar observación 1. El usuario selecciona la opción "Editar" de una de las observaciones realizadas por el usuario. 2. El sistema muestra un formulario con los campos que se describen en el caso de uso "Alta de observación". 3. El usuario introduce los cambios que considera oportuno y pulsa la opción "Guardar". 4. El sistema almacena los cambios y muestra el resultado.
Importancia	Media
Prioridad	Media
Estado	Aprobado

Nombre	Borrar observación
Versión	1.0
Dependencias	Borrar observación -- Editar observación
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Debe de existir al menos una observación en el listado de observaciones del usuario. Postcondición: La observación queda eliminada definitivamente del sistema.
Subsistema asociado	Subsistema de gestión de observaciones
Descripción	El sistema permitirá al usuario eliminar observaciones que haya realizado.
Escenarios	Basic Path - Borrar obsevación 1. El usuario ejecuta el caso de uso "Editar observación" sobre la observación que desea eliminar. 2. El usuario selecciona la opción "Borrar". 3. El sistema elimina la observación de la base de datos.
Importancia	Alta
Prioridad	Media

2.2.7. Subsistema de gestión de amigos



CU06 - Subsistema de gestión de amigos

Nombre	Listar amigos
Versión	1.0
Dependencias	Listar amigos -- Entrar Consultar perfil de amigo -- Listar amigos Añadir amigo -- Listar amigos Astrónomo -- Listar amigos
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema muestra la lista de amigos del usuario.
Subsistema asociado	Subsistema de gestión de amigos
Descripción	El sistema permitirá listar los amigos de un usuario. Dos usuarios son amigos si uno de ellos ha realizado una solicitud de amistad y la otra persona la ha aceptado.
Escenarios	Basic Path - Listar amigos 1. El usuario selecciona la opción "Ver amigos" del menú "Amigos" situado en el menú principal. 2. El sistema muestra un listado con los amigos del usuario y un botón para añadir amigos.
Importancia	Baja

Prioridad	Baja
Estado	Aprobado
Nombre	Consultar perfil de amigo
Versión	1.0
Dependencias	Listar observaciones -- Consultar perfil de amigo Consultar perfil de amigo -- Listar amigos
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. El usuario debe de ser amigo del usuario a consultar. Postcondición: El sistema muestra una pantalla con los datos personales del usuario amigo y un enlace a su catálogo de observaciones.
Subsistema asociado	Subsistema de gestión de amigos
Descripción	El sistema permitirá consultar el perfil de los usuarios amigos con sus datos personales, así como un enlace a sus observaciones.
Escenarios	Basic Path - Consultar perfil de amigo 1. El usuario pulsa la opción "Ver" de la lista de amigos. 2. El sistema muestra una pantalla con las observaciones realizadas por su amigo.
Importancia	Baja
Prioridad	Baja
Estado	Aprobado

Nombre	Buscar astrónomos
Versión	1.0
Dependencias	Añadir amigo -- Buscar astrónomos
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema muestra un listado de usuarios astrónomos que cumple el criterio de búsqueda.
Subsistema asociado	Subsistema de gestión de amigos
Descripción	El sistema permitirá al usuario buscar otros usuarios con perfil de astrónomo para enviarle solicitud de amistad.
Escenarios	Basic Path - Buscar astrónomos 1. El sistema muestra un listado completo de usuarios astrónomos .
Importancia	Baja
Prioridad	Baja
Estado	Aprobado

Nombre	Añadir amigo
Versión	1.0
Dependencias	Añadir amigo -- Buscar astrónomos Añadir amigo -- Listar amigos
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Deben de existir al menos dos usuarios astrónomos en la base de datos. Postcondición: El primer usuario registra un envío de solicitud, que es recibida al instante en la bandeja de amistades pendientes del usuario destinatario.
Subsistema asociado	Subsistema de gestión de amigos
Descripción	El sistema permitirá enviar solicitudes de amistad a otros usuarios
Escenarios	Basic Path - Añadir amigo 1. El usuario selecciona la opción "Añadir amigo" de la pantalla "Listar amigos". 2. El sistema muestra el listado de astrónomos del sistema y un filtro de búsqueda. 3. El usuario ejecuta el caso de uso "Buscar astrónomos". 4. El usuario selecciona el icono correspondiente a "Enviar solicitud de amistad" al usuario deseado. 5. El sistema envía una solicitud de amistad al usuario destinatario, y se registra la solicitud en el caso de uso "Amistades pendientes".
Importancia	Baja
Prioridad	Baja
Estado	Aprobado

Nombre	Amistades pendientes
Versión	1.0
Dependencias	Amistades pendientes -- Entrar Aceptar amistad -- Amistades pendientes Astrónomo -- Amistades pendientes
Precondición/ postcondición	Precondición: El usuario debe de estar autenticado y autorizado. Postcondición: El sistema muestra las solicitudes enviadas y recibidas del usuario.
Subsistema asociado	Subsistema de gestión de amigos
Descripción	El sistema permitirá consultar las solicitudes de amistad enviadas a otros usuarios, así como las solicitudes de amistad recibidas.

Escenarios	<p>Basic Path - Amistades pendientes</p> <p>1. El usuario selecciona la opción "Amistades pendientes" de la opción "Amigos" del menú principal.</p> <p>2. El sistema muestra un listado de solicitudes enviadas y recibidas; con su correspondiente estado:</p> <ul style="list-style-type: none"> • ENVIADA: si es el usuario el que se la ha enviado a otro usuario. • RECIBIDA: si es otro usuario el que la ha enviado al usuario autenticado.
Importancia	Baja
Prioridad	Baja
Estado	Aprobado

Nombre	Aceptar amistad
Versión	1.0
Dependencias	Aceptar amistad -- Amistades pendientes
Precondición/ postcondición	<p>Precondición: El usuario debe de estar autenticado y autorizado. La amistad debe de estar en estado RECIBIDA.</p> <p>Postcondición: Los dos usuarios quedan registrados como amigos.</p>
Subsistema asociado	Subsistema de gestión de amigos
Descripción	El sistema permitirá al usuario aceptar amistades que haya recibido.
Escenarios	<p>Basic Path - Aceptar amistad</p> <p>1. El usuario selecciona la opción "Aceptar amistad", relativa a una amistad recibida en su lista de amistades pendientes.</p> <p>2. El sistema registra la amistad entre los dos astrónomos y el usuario aceptado está disponible en la lista de amigos del usuario autenticado y viceversa.</p>
Importancia	Baja
Prioridad	Baja
Estado	Aprobado

3. Diseño

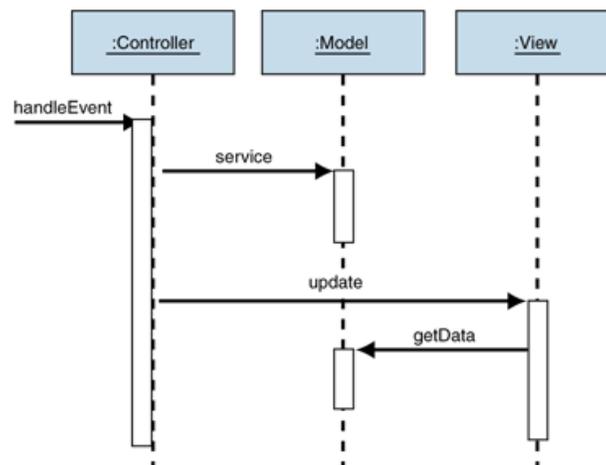
3.1. Arquitectura

El Cuaderno del Astrónomo es una aplicación enfocada claramente a que el usuario pueda manejar los datos generados en las observaciones, lo cual puede ser a través de un ordenador de sobremesa, un smartphone, un ordenador portátil, una tablet, etc.

Dada la cantidad de sistemas operativos y plataformas distintas, se decidió que El Cuaderno del Astrónomo sería una aplicación web basada en J2EE. Los motivos de optar por esta plataforma para su desarrollo son varios:

- Lenguaje Java, el cual es un lenguaje orientado a objetos y multiplataforma (*"Write once, run anywhere"*), lo cual nos permite desarrollar casi en cualquier sistema operativo.
- Gran cantidad de *frameworks* y herramientas que abstraen al programador de las tareas de más bajo nivel como por ejemplo el acceso a base de datos.
- Comunidad muy activa, la cual genera gran cantidad de información y que cualquier programador puede requerir en algún momento del desarrollo.

Actualmente, quizás el patrón de diseño más utilizado en el software empresarial es el **MVC**, que se adapta perfectamente a las necesidades de la aplicación.



Este patrón divide el desarrollo en tres partes, que de forma resumida podríamos definir como:

- **Controlador:** capa intermediaria entre el usuario y las capas modelo y vista. Encargada de llevar a cabo las acciones requeridas y el control de flujo de la aplicación.

En JBoss Seam la función de controlador es realizada por los *action beans*; los cuales son clases que extienden de *EntityHome*, *EntityQuery* y *EntityController*. Estas clases son las encargadas de realizar las operaciones de persistencia y consultas entre otras funciones.

Encontraremos estas clases en el paquete:
uoc.edu.cuadernoastronomo.session.

- **Modelo:** información (los datos) sobre la que la aplicación trabaja.

En Seam, el modelo es recogido en los *entity beans*, los cuales son mapeados a través de anotaciones de hibernate con los campos de las tablas de la base de datos.

Encontraremos los vean de entidad en el paquete:
uoc.edu.cuadernoastronomo.entity.

- **Vista:** interfaz de usuario.

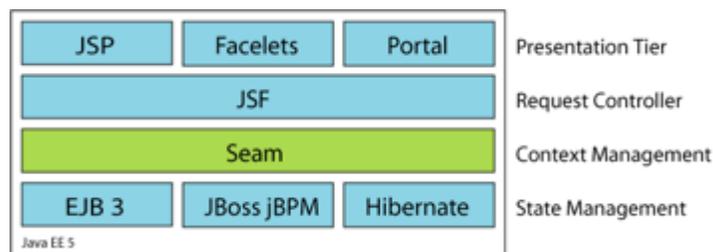
En Seam la interfaz queda recogida en ficheros **xhtml**, los cuales soportan una gran cantidad de librerías para mostrar datos como puede ser JSF o Richfaces.

3.2. JBoss seam, un framework para unirlos a todos

JBoss Seam es un *application framework* muy usado actualmente y que pertenece a una de las mayores empresas de software libre del mundo: Red Hat.

Podríamos definir JBoss Seam como una telaraña perfectamente tejida de tecnologías, herramientas y *frameworks*, que coge lo mejor de cada una de éstas para llevar al mercado una completa solución para desarrollo empresarial.

Para definir JBoss Seam tenemos que hablar de **Enterprise JavaBeans 3 (EJB3)** y **JavaServer Faces (JSF)**.



<http://docs.jboss.org/seam/2.0.2.SP1/reference/en-US/html/Book-Preface.html>

3.2.1.1. JBoss Seam y EJB3

Con EJB3, JBoss Seam consigue tener una serie de componentes que pueden ser usados en cualquier momento, dependiendo de su visibilidad y que podemos descomponer en:

- **Beans de entidad**, que representan nuestro modelo de datos y están gestionados por un *Entity Manager*, el cual, a través de *Java Persistence API* (JPA) nos facilita la tarea de persistencia de datos. Más concretamente, JBoss Seam integra *Hibernate*, una implementación de JPA.
- **Beans de sesión**, los cuales implementan la lógica de la aplicación y realizan las acciones que la vista solicita: instanciar otros beans, gestionar su ciclo de vida, etc.

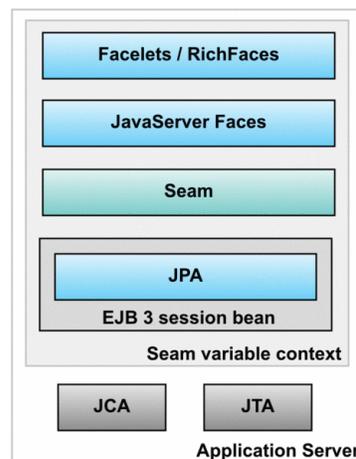
De esta forma abarcamos tanto el modelador de datos como el controlador de la aplicación.

3.2.1.2. JBoss Seam y JSF

Para la capa de presentación, JBoss Seam incluye **Richfaces** y **Icefaces** que son implementaciones de **JavaServer Faces** (JSF).

De esta forma tendremos acceso a una rica librería de componentes para usar con EJB de forma sencilla y con soporte para Ajax.

Richfaces es la implementación de JSF de JBoss; la cual será la que se use en la mayor parte de la aplicación; pero si se necesita usar otros componentes, JBoss Seam permite integrar distintas librerías de JSF en un mismo proyecto.



Tecnologías incluidas en JBoss Seam

3.2.1.3. JBoss Seam y la seguridad

Para la seguridad, Seam implementa su propia versión de **JAAS** (Java Authentication and Authorization Service) que proporciona funcionalidades de autenticación y autorización basados en permisos y roles.

Seam permite definir una serie de reglas de acceso dependiendo del rol del usuario. Estas reglas se pueden restringir ya sea para los beans de entidad, páginas o componentes de una página.

3.3. Entorno de ejecución

A continuación se detallan las herramientas que se utilizarán para el desarrollo de la aplicación, así como la descripción del sistema:

- **JDK 1.6.** Se ha decidido **no** utilizar la versión más reciente de Java (JDK 1.7) debido a que no se utilizará ninguna de las mejoras que ofrece en este proyecto y además lleva poco tiempo en el mercado.
- **Seam 2.2.0. G.A.** Siguiendo la misma filosofía que en el paso anterior, se ha optado por no escoger la última versión de Seam (Seam 3), y utilizar la última versión de **Seam 2** que es G.A.

Las siglas G.A. en los productos de JBoss hacen referencia a que es una versión General Availability, o lo que es lo mismo, versión de lanzamiento. Lo cual nos da más seguridad sobre su estabilidad.

- **JBoss 4.2.3. G.A.** Contenedor de aplicaciones J2EE muy utilizado con JBoss Seam, y que al ser de la misma compañía nos garantiza mayor facilidad de integración. Por otro lado, JBoss es un servidor multiplataforma, muy potente y fácil de administrar.
- **MySQL Server 5.5.21.** Última versión a fecha de hoy de este popular gestor de base de datos de libre distribución; que además de disponer de versiones para Windows y Linux, se puede complementar con una serie de herramientas de usuario para la creación, gestión y manejo de la base de datos; en este caso particular MySQL Workbench 5.2 CE.
- **Conector MySQL para JAVA 5.0.8.** Librería Java para la conexión entre la aplicación y la base de datos.

Para el desarrollo y la ejecución de toda la aplicación se ha montado una máquina virtual con **VMware** con las siguientes características:

- Procesador AMD Phenom II 2.8 Ghz
- Windows XP SP3 32 bits
- 2 GB RAM
- 40 GB DD

3.4. Herramientas de desarrollo

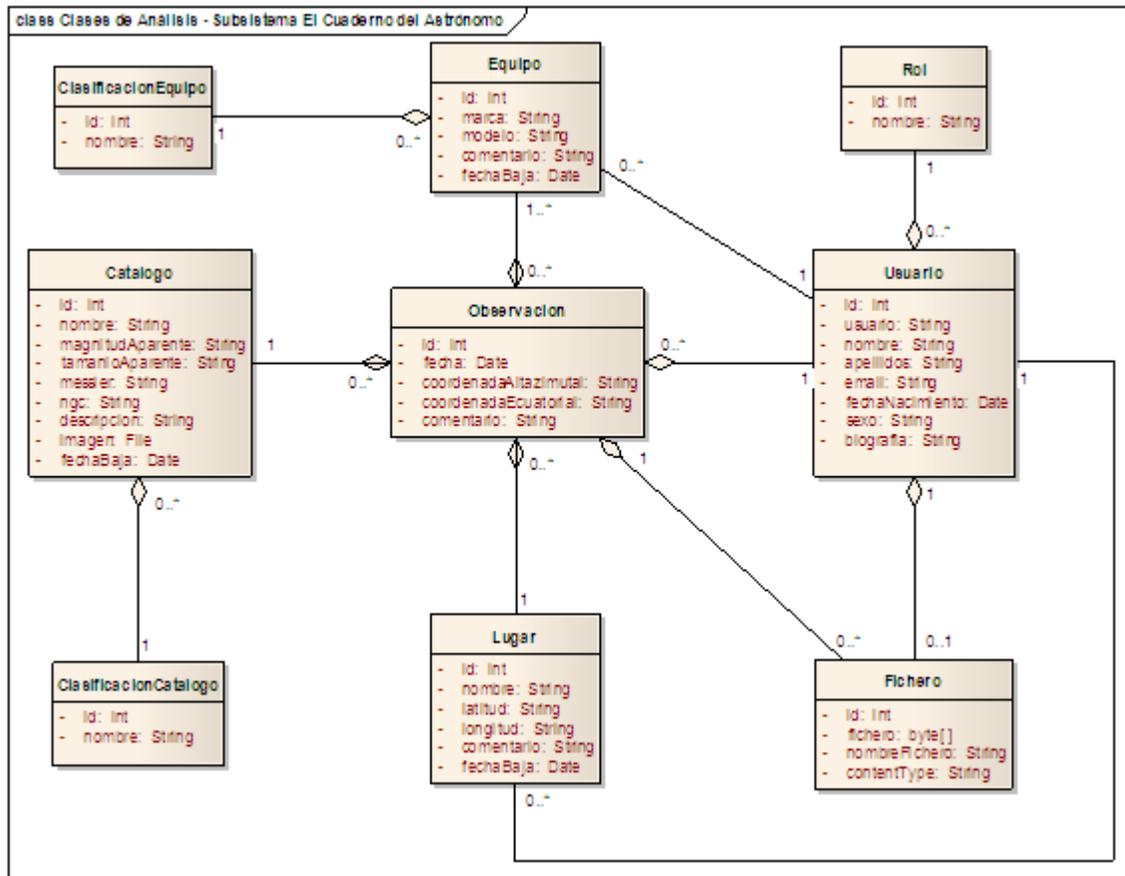
Para el desarrollo de la aplicación se hará uso de 2 aplicaciones:

- **MySQL Workbench 5.2 CE:** para la creación y gestión de la base de datos.
- **Eclipse Helios:** integrado con el plugin de Seam que facilita la configuración del framework e incluye opciones útiles para el desarrollo de la aplicación.

Otra opción gratuita sería **jDeveloper** que ya integra las funcionalidades de desarrollo para Seam, pero dado que es el mismo IDE con la personalización de JBoss, se usará Eclipse.

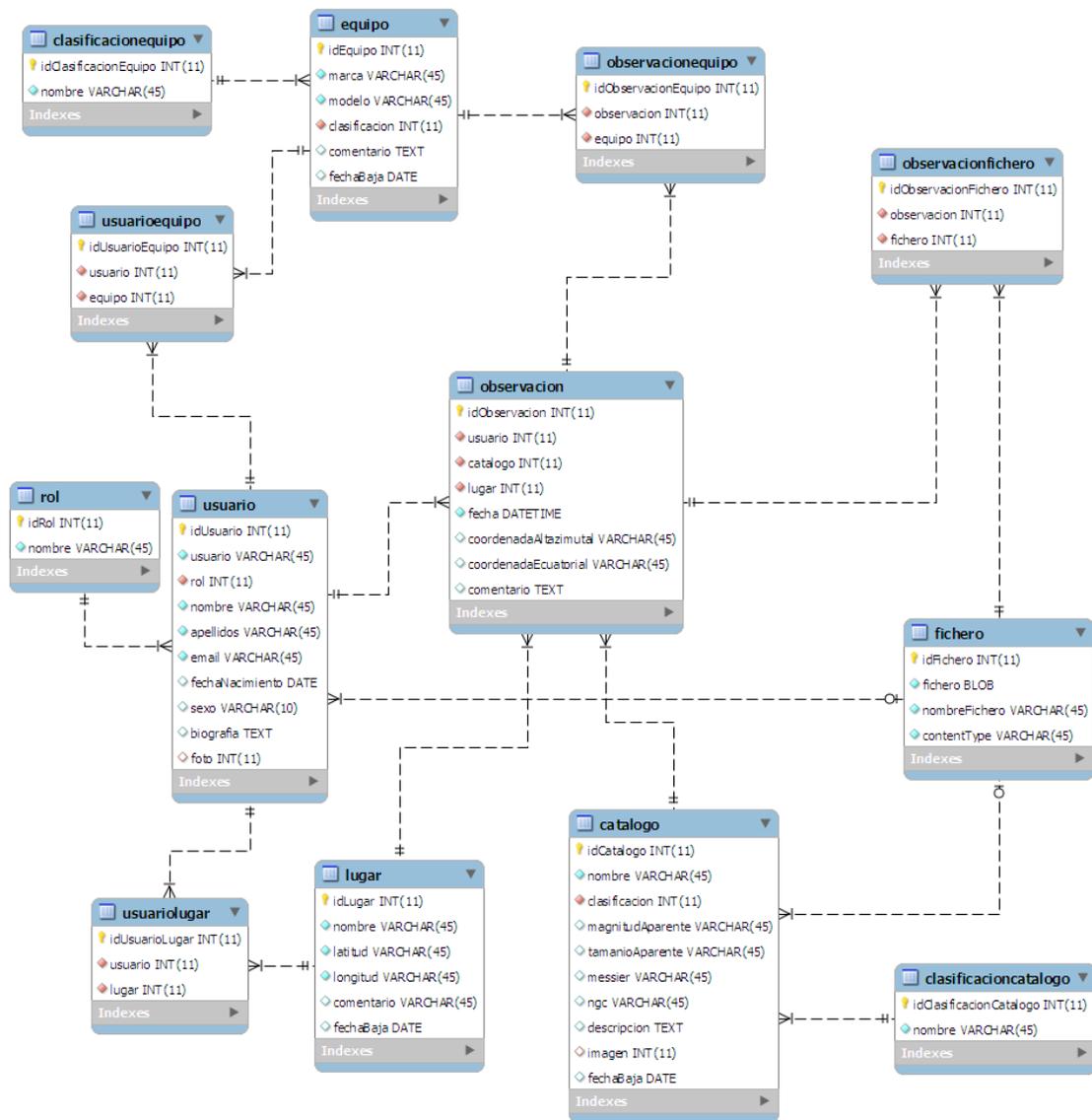
3.5. Diagrama de clases

A continuación se presenta el diagrama de **clases del negocio**.



Los atributos implícitos de una relación entre clases no se transcriben. Por ejemplo, de la relación Usuario – Fichero se deduce que existe un atributo de este último tipo en la clase Usuario (que será la foto del usuario). Estos atributos aparecerán en las clases llegado el momento de la implementación.

3.6. Modelo de datos



4. Implementación

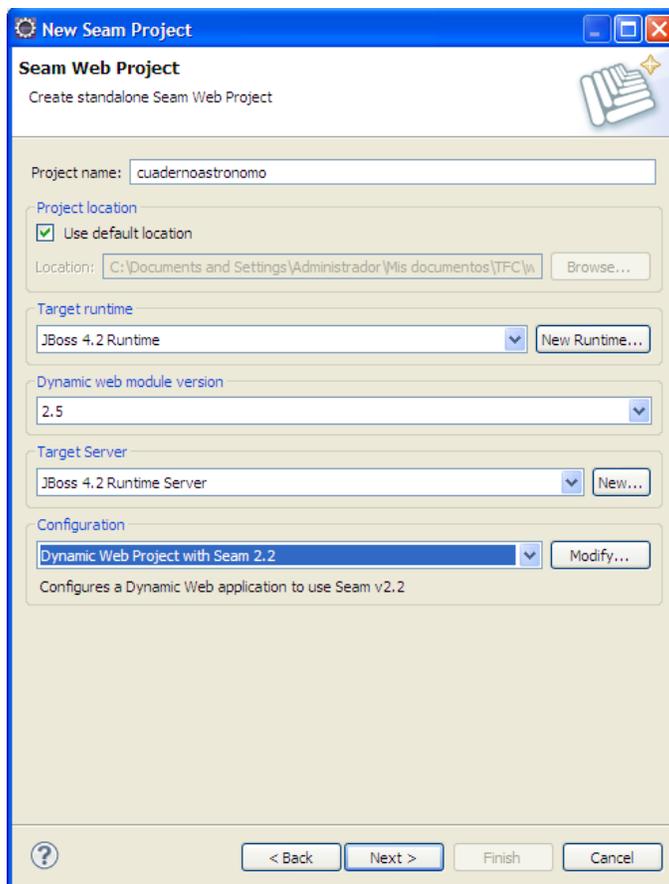
4.1. Creación de un nuevo proyecto y seam-gen

Una de las grandes bondades de JBoss Seam y sus 'JBoss Tools' para Eclipse, es la posibilidad de realizar ingeniería inversa y construir el esqueleto de una aplicación a partir de la base de datos gracias a **seam-gen**.

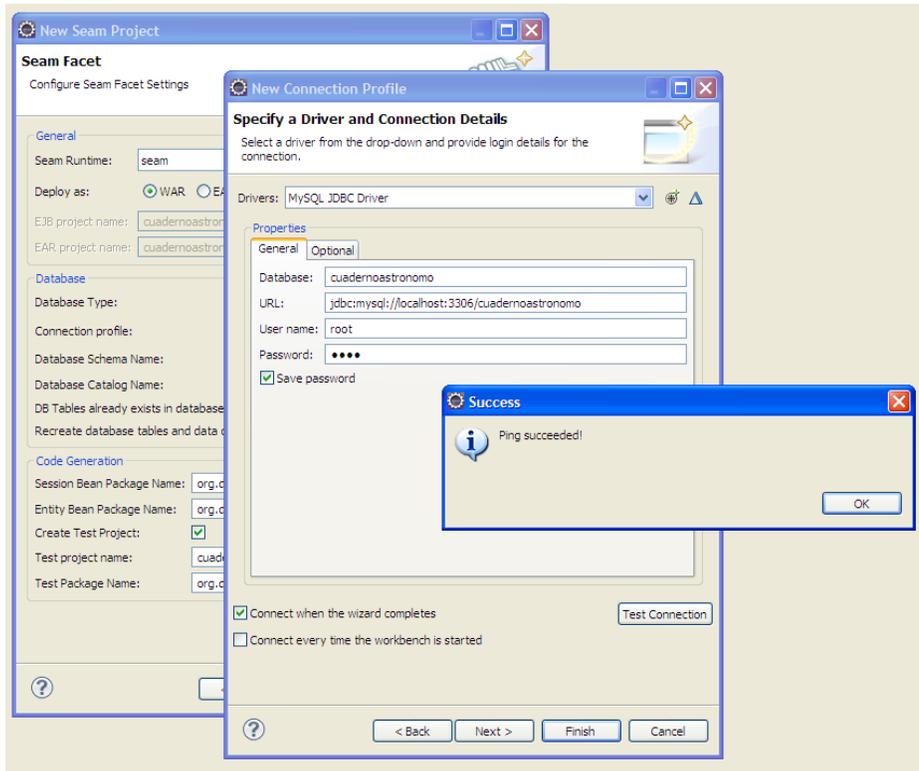
Este proceso nos ahorra un trabajo sencillo pero tedioso, como es la creación de los bean de entidad, las clases controladoras y XHTML para las vistas. Evidentemente habrá muchas persistencias que sobrescribir o listados que rehacer, permisos que establecer, vistas en XHTML que crear y/o cambiar... en definitiva, el proceso de desarrollo no ha hecho más que empezar.

Los pasos a dar son:

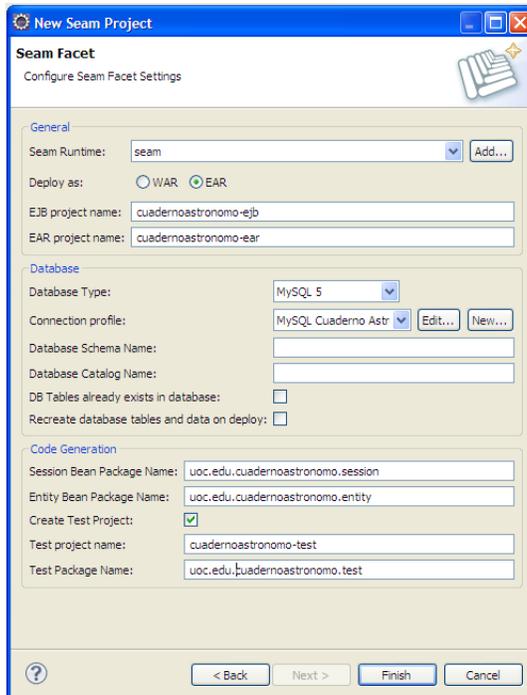
- File → New → Seam web Project



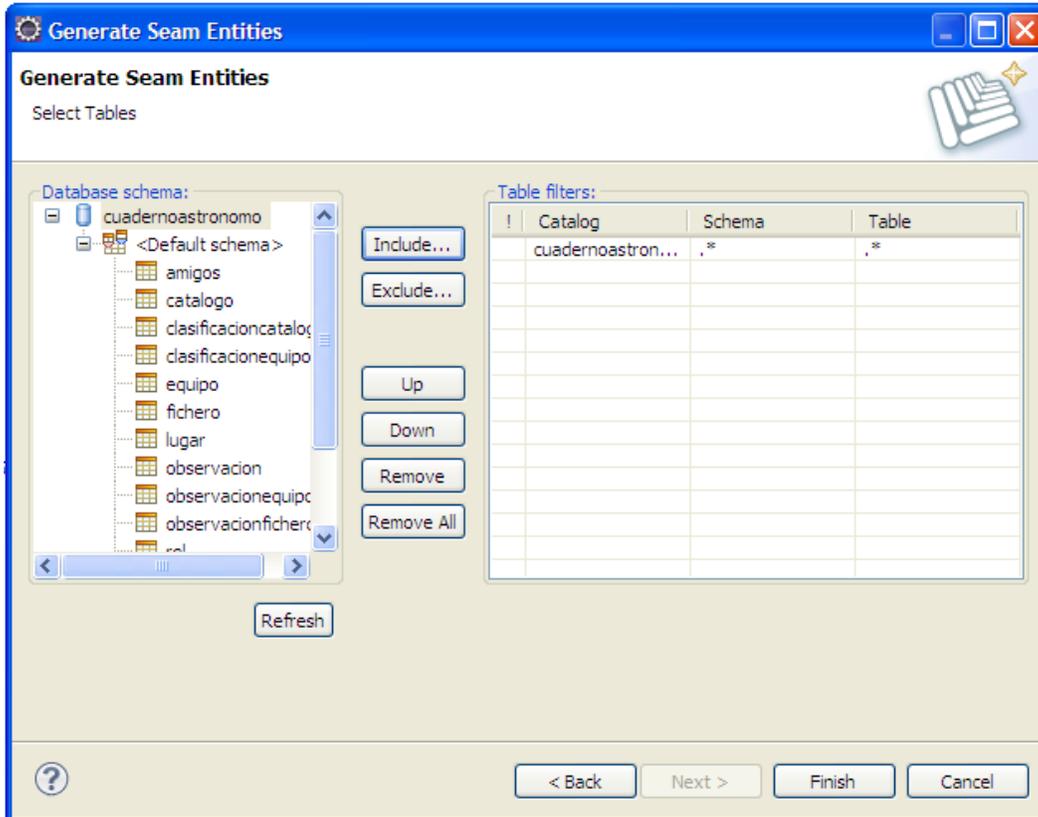
- Configuramos una nueva conexión de base de datos



- New Seam Facet (donde configuramos opciones de despliegue, paquete o conexión)



- Pulsamos FINISH y se generarán los 4 proyectos.
 - cuadernoastrologo
 - cuadernoastrologo-ear
 - cuadernoastrologo-ejb
 - cuadernoastrologo-test
- Creación de entidades: New → Seam generate entities (reverse engineer from db)



Seleccionamos todas las tablas y al pulsar 'Finish' obtendremos el esqueleto de nuestra aplicación.

4.2. Persistencia de entidades

En Seam, por cada entidad tendremos una clase Home que es la encargada de la persistencia. Para ello se hace uso de *Hibernate* y el *EntityManager*, incluido en el *core* de Seam para realizar las transacciones.

A continuación se muestra un ejemplo con la clase **UsuarioHome**. En este caso hemos sobrescrito los métodos de **persist** (guardar por primera vez), **update** y hemos creado **registrar**.

```

@Name("usuarioHome")
public class UsuarioHome extends CuadernoAstronomoEntityHome<Usuario> {

    private static final long serialVersionUID = -2071095645794319914L;
    @In(create = true)
    RolHome rolHome;
    @In(create = true)
    FicheroHome ficheroHome;

    private Fichero fichero = new Fichero();

    public void setUsuarioIdUsuario(Integer id) {
        setId(id);
    }

    public Integer getUsuarioIdUsuario() {
        return (Integer) getId();
    }

    @Override
    protected Usuario createInstance() {
        Usuario usuario = new Usuario();
        return usuario;
    }

    public void load() {
        if (isIdDefined()) {
            wire();
        }
    }

    @Override
    public String persist(){
        String result = PERSIST;
        try {
            FicheroHome ficheroHome = new FicheroHome();
            ficheroHome.setInstance(fichero);
            ficheroHome.getEntityManager().persist(ficheroHome.getInstance());
            ficheroHome.getEntityManager().flush();

            Usuario usuario = getInstance();
            usuario.setFichero(fichero);
            getEntityManager().persist(usuario);
            getEntityManager().flush();

            addInfo("Usuario guardado correctamente.");
            result = PERSIST;
        } catch (EntityExistsException eeex){
            addError("Error, el usuario ya existe.");
            result = ERROR;
        }
        catch(Exception e){
            addError("Error al guardar el usuario");
            result = ERROR;
        }
        return result;
    }
}

```

```

@Override
public String update() {
    String result = "";
    try {
        FicheroHome ficheroHome = new FicheroHome();
        ficheroHome.setInstance(fichero);
        ficheroHome.getEntityManager().persist(ficheroHome.getInstance());
        ficheroHome.getEntityManager().flush();

        Usuario usuario = getInstance();
        usuario.setFichero(fichero);
        getEntityManager().persist(usuario);
        getEntityManager().flush();

        addInfo("Usuario guardado correctamente.");
        result = UPDATE;
    } catch (EntityExistsException eeex) {
        addError("Error, el usuario ya existe.");
        result = ERROR;
    }
    catch (Exception e) {
        addError("Error al guardar el usuario");
        result = ERROR;
    }
    return result;
}

```

```

}

```

```

@SuppressWarnings("deprecation")
public String registrar() {
    try {
        PasswordHash pHash = new PasswordHash();
        pHash.setHashAlgorithm(PasswordHash.ALGORITHM_MD5);
        instance.setPassword(pHash.generateHash(instance.getPassword()));
        facesMessages.clear();

        //Asignamos el Rol ASTRONOMO
        RolController rolController = new RolController();
        instance.setRol(rolController.getRolByNombre(Constants.ROL_ASTRONOMO));

        FicheroHome ficheroHome = new FicheroHome();
        ficheroHome.setInstance(fichero);
        ficheroHome.getEntityManager().persist(ficheroHome.getInstance());
        ficheroHome.getEntityManager().flush();

        instance.setFichero(fichero);

        super.persist();
        this.getEntityManager().flush();

        addInfo("Usuario registrado");
    } catch (Exception e) {
        addWarn("No se puede registrar el usuario");
        return ERROR;
    }
    return PERSIST;
}
}

```

4.3. Listado de entidades

Para listar entidades en Seam haremos uso de dos tipos de clase: *List* y *Controller*. Las clases *List* contienen la lógica para las búsquedas y listados por defecto. Si queremos añadir listados personalizados, en nuestro caso haremos uso de las clases *Controller*, siendo éste un escenario más cómodo para programar nuestras propias consultas.

Veamos un ejemplo sencillo con el listado de Equipo:

EquipoList

```
@Name("equipoList")
public class EquipoList extends EntityQuery<Equipo> {

    private static final long serialVersionUID = -4997771748922952026L;

    private static final String EJBQL = "select equipo from Equipo equipo where equipo.fechaBaja is null";

    private static final String[] RESTRICTIONS = {
        "lower(equipo.marca) like lower(concat(#{equipoList.equipo.marca},'%'))",
        "lower(equipo.modelo) like lower(concat(#{equipoList.equipo.modelo},'%'))",
        "lower(equipo.comentario) like lower(concat(#{equipoList.equipo.comentario},'%'))", };

    private Equipo equipo = new Equipo();

    public EquipoList() {
        setEjbql(EJBQL);
        setRestrictionExpressionStrings(Arrays.asList(RESTRICTIONS));
        setMaxResults(25);
    }

    public Equipo getEquipo() {
        return equipo;
    }

    public List<Equipo> getEquipoByUsuario(String usuario){
        EquipoController eController = new EquipoController();
        return eController.getEquipoByUsuario(usuario);
    }
}
```

EquipoController

```
public class EquipoController extends CuadernoAstronomoEntityController{

    private static final long serialVersionUID = -6216325279593457786L;
    private final String SQL_EQUIPO_BY_USUARIO = "SELECT equipo FROM Equipo equipo " +
        "WHERE equipo.fechaBaja IS NULL AND equipo.usuario.usuario = :usuario";

    public List<Equipo> getEquipoByUsuario(String usuario){
        HashMap<String, Object> params = new HashMap<String, Object>();
        params.put("usuario", usuario);
        return getList(Equipo.class, SQL_EQUIPO_BY_USUARIO, params);
    }
}
```

4.4. Vista en XHTML

En los ficheros XHTML se incrustará el código encargado de mostrar cada página, siendo una mezcla de HTML y las *taglib* de las librerías incluidas como *Richfaces* y *JSF*.

Como ejemplo, mostramos el código de la vista **Catalogo.xhtml**

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/template.xhtml">

    <ui:define name="navigation">
        <rich:panel styleClass="rastroMiga">
            <s:link view="#{navigationType.home}"
                value="Inicio" includePageParams="false"/>
            <h:outputText value=" » " />
            <s:link view="#{navigationType.catalogoList}"
                value="Catálogo" includePageParams="false"/>
            <h:outputText value=" » " />
            <h:outputText value="Objeto del catálogo" />
        </rich:panel>
    </ui:define>

    <ui:define name="body">
        <rich:panel>
            <f:facet name="header">Detalles del elemento del Catálogo</f:facet>

            <table>
            <tr>
                <td width="55%">
                    <s:decorate id="nombre" template="/layout/display.xhtml">
                        <ui:define name="label">Nombre</ui:define>
                        <h:outputText value="#{catalogoHome.instance.nombre}"/>
                    </s:decorate>

                    <s:decorate id="magnitudAparente" template="/layout/display.xhtml">
                        <ui:define name="label">Magnitud aparente</ui:define>
                        <h:outputText value="#{catalogoHome.instance.magnitudAparente}"/>
                    </s:decorate>

                    <s:decorate id="tamanoAparente" template="/layout/display.xhtml">
                        <ui:define name="label">Tamaño aparente</ui:define>
                        <h:outputText value="#{catalogoHome.instance.tamanoAparente}"/>
                    </s:decorate>

                    <s:decorate id="messier" template="/layout/display.xhtml">
                        <ui:define name="label">Messier</ui:define>
                        <h:outputText value="#{catalogoHome.instance.messier}"/>
                    </s:decorate>

                    <s:decorate id="ngc" template="/layout/display.xhtml">
                        <ui:define name="label">NGC</ui:define>
                        <h:outputText value="#{catalogoHome.instance.ngc}"/>
                    </s:decorate>
                </td>
            </tr>
            </table>
        </rich:panel>
    </ui:define>
</ui:composition>
```

```

        <s:decorate id="descripcion" template="/layout/display.xhtml">
            <ui:define name="label">Descripción</ui:define>
            <h:inputTextarea value="#{catalogoHome.instance.descripcion}"
                disabled="true" rows="20" cols="90"/>
        </s:decorate>
    </td>
    <td>
        <s:decorate id="fichero" template="/layout/display.xhtml">
            <s:graphicImage rendered="#{catalogoHome.instance.fichero ne null}"
                style="border: 2px solid green"
                value="#{catalogoHome.instance.fichero.fichero}">
                <s:transformImageSize height="400" maintainRatio="true"/>
            </s:graphicImage>
        </s:decorate>

        <s:decorate id="fechaBaja" template="/layout/display.xhtml"
            rendered="#{catalogoHome.instance.fechaBaja ne null}">
            <ui:define name="label">Fecha baja</ui:define>
            <h:outputText value="#{catalogoHome.instance.fechaBaja}">
                <s:convertDateTime type="date" dateStyle="short"/>
            </h:outputText>
        </s:decorate>
    </td>
</tr>
</table>
<div style="clear:both"/>

</rich:panel>

<div class="actionButtons">

    <s:button view="/catalogo/CatalogoEdit.xhtml"
        id="edit"
        rendered="#{identity.hasRole(rolType.ADMIN)}"
        value="Editar"/>

    <s:button view="/catalogo/#{empty catalogoFrom ? 'CatalogoList' : catalogoFrom}.xhtml"
        id="done"
        value="Hecho"/>

</div>

</ui:define>

</ui:composition>

```

4.5. Control de permisos y flujo de páginas

Cada fichero XHTML puede llevar asociado un fichero con el mismo nombre, pero con la terminación **.page** en el nombre y formato **XML**.

Estos ficheros tienen varias utilidades, entre las que destacan:

- Control de paso de parámetros entre los formularios XHTML y las clases de la lógica.
- Gestión de permisos y autorización por perfiles.
- Flujo de páginas dependiendo del resultado de la operación realizada a través de las reglas de navegación.

Sirva como ejemplo `CatalogoEdit.page.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<page xmlns="http://jboss.com/products/seam/pages"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://jboss.com/products/seam/pages http://jboss.com/products/seam/pages-2.2.xsd"
      no-conversation-view-id="/catalogo/CatalogoList.xhtml"
      login-required="true">
  <restrict>#{identity.hasRole(rolType.ADMIN)}</restrict>
  <begin-conversation join="true" flush-mode="MANUAL"/>

  <action execute="#{catalogoHome.wire}"/>

  <param name="catalogoFrom"/>
  <param name="catalogoIdCatalogo" value="#{catalogoHome.catalogoIdCatalogo}"/>
  <param name="clasificacioncatalogoFrom"/>
  <param name="clasificacioncatalogoIdClasificacionCatalogo"
        value="#{clasificacioncatalogoHome.clasificacioncatalogoIdClasificacionCatalogo}"/>

  <param name="ficheroFrom"/>
  <param name="ficheroIdFichero" value="#{ficheroHome.ficheroIdFichero}"/>

  <navigation from-action="#{catalogoHome.persist}">
    <rule if-outcome="persisted">
      <redirect view-id="/catalogo/Catalogo.xhtml"/>
    </rule>
  </navigation>

  <navigation from-action="#{catalogoHome.update}">
    <rule if-outcome="updated">
      <redirect view-id="/catalogo/Catalogo.xhtml"/>
    </rule>
  </navigation>

  <navigation from-action="#{catalogoHome.remove}">
    <rule if-outcome="removed">
      <redirect view-id="/catalogo/CatalogoList.xhtml"/>
    </rule>
    <rule if-outcome="error">
      <redirect view-id="/catalogo/CatalogoEdit.xhtml"/>
    </rule>
  </navigation>
</page>
```

Podemos comprobar cómo en la etiqueta <restrict> le indicamos el único perfil que tiene permisos para mostrar esta página; los parámetros con los que trabajará la página y por último las reglas de navegación definidas.

5. Valoración económica

Para la realización del proyecto se consideran los siguientes perfiles y precios por jornada:

Perfil	Coste Jornada 8h (IVA incluido)
Administrador de Bases de Datos	80 €
Analista	100 €
Programador Senior	90 €

El número de jornadas estimadas por perfil basándonos en el plan de trabajo es:

Perfil	Jornadas	Tareas
Administrador de Bases de Datos	2	Creación de la base de datos
Analista	29	Análisis, diseño y documentación
Programador Senior	21	Implementación de la aplicación

Las jornadas del plan de trabajo son de 4 horas, por lo que se ha reducido a la mitad el número de jornadas completas necesarias.

Teniendo en cuenta el precio por jornada y perfil y el número de días obtenemos el coste en personal para la creación de la aplicación.

Perfil	Jornadas	Coste Jornada	€
Administrador de Bases de Datos	2	80 €	160
Analista	29	100 €	2.900
Desarrollador	21	90 €	1.890
		Total	4.950

Los costes del software utilizado para la aplicación es 0, dado que el proyecto está basado en software libre y gratuito.

6. Conclusiones

Es el momento de recapitular, mirar atrás y sacar conclusiones. Han sido varios meses de intenso trabajo, pero el resultado ha cumplido los objetivos que nos marcamos en los inicios.

Este hito, en el mundo de la construcción de software, lejos de ser lo habitual, es una de los mayores retos al que un equipo de desarrollo se enfrenta cuando se coge un nuevo proyecto.

El primer paso importante lo establecemos en las primeras reuniones con el cliente ¿sabemos qué necesita el cliente?, ¿sabe el cliente lo que necesita?, ¿contamos con el equipo técnico y humano para darle solución?

En caso afirmativo, tenemos mucho ganado, ya que uno de los principales problemas a la hora de desarrollar son los cambios. Es muy importante tener claro lo que vamos a desarrollar y lo que el cliente va a obtener; si una de las partes no está segura, el proyecto puede acabar con retrasos y pérdidas.

Otro punto, que por experiencia propia he podido vivir en entornos ‘profesionales’, es la poca importancia que se le da a veces a la fase de análisis y diseño; siendo habitual encontrarte como programador con un simple diagrama de flujos, un diagrama de clases o peor aún; una reunión informal de dos horas para explicar lo que tienes que hacer. Cuando el equipo está formado por varias personas, el caos puede ser permanente; es por ello que se le ha dedicado especial atención a definir todos y cada uno de los casos de uso existentes y a modelar correctamente las clases del negocio y el diseño relacional.

Teniendo las bases bien sentadas para empezar a desarrollar, no nos hemos olvidado de cuidar la formación en JBoss Seam a lo largo de todo el proyecto. Este es otro punto que a veces las empresas no cuidan: la formación de sus empleados. No existe el programador que conozca todos los lenguajes y todos los *frameworks*; por lo que es importante disponer de los recursos humanos ya preparados en cada fase del proyecto.

Esta fase es la que sin duda ha requerido de mayor esfuerzo, ya que el adaptarse a las directrices de un nuevo framework siempre es una tarea que requiere dedicación; y aunque Seam usa otras herramientas o *frameworks* muy populares, de todas ellas sólo conocía *Hibernate*.

Si tuviese que volver a seleccionar framework, sin duda elegiría Seam, el cual considero, después de la experiencia, más intuitivo y productivo que otros que he podido probar como *Spring MVC* o *Struts 2*.

Como punto negativo, creo que Seam es un *framework* que requiere más tiempo de formación que otros como *Struts 2*, que de cara a un primer proyecto *J2EE* considero que puede ser más recomendable por resultar más didáctico.

7. Glosario de términos.

- **Catálogo Messier:** famoso catálogo de objetos celestes realizado por el astrónomo Charles Messier.
- **Catálogo NGC:** catálogo de nebulosas y cúmulos de estrellas con más de 7800 objetos registrados.
- **Cúmulo estelar:** agrupación de estrellas que está dentro u orbita una galaxia.
- **EJB3:** es como se denomina a Enterprise JavaBeans 3. Parte de la API que reúne los estándares para el desarrollo de aplicaciones Java EE.
- **Framework:** herramienta tecnológica compuesta por módulos de software y una serie de reglas que sirve como base para la construcción de otro software.
- **Hibernate:** herramienta de mapeo de entidades Java con campos de tablas de bases de datos relacionales.
- **JBoss:** servidor de aplicaciones J2EE multiplataforma implementado en Java perteneciente a Red Hat (También es el nombre de la división de desarrollo de productos de Red Hat).
- **JBoss Seam:** framework desarrollado por la división JBoss que combina los frameworks EJB3 y JSF entre otras características.
- **JSF:** JavaServer Faces, estándar de la API de Java para la construcción de interfaces de usuario.
- **Latitud:** distancia angular entre la línea ecuatorial, el ecuador, y un punto determinado del planeta.
- **Longitud:** distancia angular entre un punto dado de la superficie terrestre y el meridiano que se tome como 0° (es decir el meridiano base), tomando como centro angular el centro de la Tierra.
- **Magnitud aparente:** índice para medir el brillo de un objeto celeste visto desde nuestro planeta.
- **Modo noche:** funcionalidad especial de algunas aplicaciones que reducen el brillo o modifican su propio estilo para minimizar la cantidad de luz emitida.

- **Montura:** soporte para aguantar el tubo telescópico. Se sitúa entre el tubo y el trípode.
- **Montura altazimutal:** soporte usado para mover un telescopio o una cámara fotográfica a lo largo de dos ejes perpendiculares de movimiento (horizontal y vertical). Estos movimientos son medidos en relación al observador (que tiene posición 0° Azimut, 0° altura).
- **Montura Dobson:** soporte usado para mover un telescopio de forma manual de forma altazimutal.
- **Montura ecuatorial:** soporte usado para mover un telescopio o una cámara fotográfica a lo largo de dos ejes perpendiculares de movimiento. Estos dos ejes se conocen como ascensión recta y declinación
- **Nebulosa:** regiones del medio interestelar constituidas por gases (principalmente hidrógeno y helio) y elementos químicos pesados en forma de polvo cósmico.
- **Ocular:** lente que transforma la luz recogida por el telescopio en una imagen observable por el observador.
- **Richfaces:** herramienta de JBoss para la construcción de interfaces de usuario construida sobre JSF.
- **Tamaño aparente:** cantidad de campo que ocupa un objeto en el cielo. Se mide en grados.

8. Bibliografía

Dan Allen (2008): Seam in action. Manning.

Primeros pasos con Jboss Seam: disponible en la web

http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=JBossSeam_primeros_pasos

Event-Driven MVC with JSF and JBoss Seam: disponible en la web

<http://java.dzone.com/articles/event-driven-mvc-jsf-and-jboss>

Seam Framework community documentation: disponible en la web

<http://docs.jboss.org/seam/2.2.0.GA/reference/en-US/html/tutorial.html>

Seam API documentation: disponible en la web

<http://docs.jboss.org/seam/2.2.0.GA/api/overview-summary.html>

Seam community: disponible en la web

<https://community.jboss.org/en/seam/seam2?view=discussions>