



Ornithoculture: aviary management.

Rafael Rodríguez Sánchez
Grado de Ingeniería Informática
Java EE

Albert Grau Perisé
Santi Caballe Llobet

13 de junio de 2018



Esta obra está sujeta a una licencia de [Creative Commons: Attribution - Non Commercial - No Derivate Works 3.0 Spain](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Ornithoculture: aviary management</i>
Nombre del autor:	<i>Rafael Rodríguez Sánchez</i>
Nombre del consultor:	<i>Albert Grau Perisé</i>
Nombre del PRA:	<i>Santi Caballe Llobet</i>
Fecha de entrega:	<i>06/2018</i>
Titulación:	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>Java EE</i>
Idioma del trabajo:	<i>Inglés</i>
Palabras clave	<i>breeding canary birds</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo</i>	
<p>Este proyecto pretende ser una herramienta con un objetivo bastante específico. Eso sí, ello no nos ha desviado de la dirección en que el análisis, el diseño y la implementación del mismo pretenden un enfoque general, en el sentido de que el resultado final pueda ser reutilizado en situaciones similares.</p> <p>Así pues, se ha pretendido que el análisis y el diseño del proyecto siguieran una formalización basada en el paradigma de la Orientación a Objetos. Y, basándonos en que la arquitectura tuviese un diseño de tres capas. Por tanto, tratándose de una arquitectura distribuida en ciertos aspectos de la misma.</p> <p>Para su diseño se han realizado entrevistas. Que, posteriormente han sido plasmadas en el diseño resultante en lenguaje UML. Por otra parte, la herramienta de implementación del proyecto es fundamentalmente el lenguaje Java EE y sus tecnologías. De esta forma se hace posible conseguir una aplicación reutilizable, escalable y robusta.</p> <p>Las mencionadas tecnologías de Java permiten realizar una separación de funcionalidades atendiendo a la estructura de las tres capas y al patrón MVC planteado. Todo ello permite un buen nivel de desacoplamiento entre los códigos de acceso a datos, negocio y presentación, permitiendo un buen grado de reutilización y modificación.</p> <p>Se hace necesario hacer notar que el análisis y el diseño de la aplicación han sido completados en casi su totalidad. En cambio, en la implementación sólo se ha podido crear la estructura básica de la aplicación y algunas muestras de casos de uso. Además, una muestra de servicio REST.</p>	

Sumario

1. Motivación del TFG.....	2
2. Objetivos.....	3
3. Planificación y programación.....	4
4. Análisis.....	5
4.1. Descripción del problema.....	5
4.2. Enumeración de requisitos de producto.....	8
4.2.1 Requisitos funcionales.....	8
Gestión del sistema.....	8
Gestión de aves.....	8
Gestión de cría.....	9
Gestión de certámenes.....	9
Gestión de la información.....	10
4.2.2 Requisitos no funcionales.....	12
5. Especificación.....	13
5.1 Índice de casos de uso.....	15
5.2 Sumario de actores.....	15
5.3 Casos de uso.....	16
6. Datos.....	29
6.1 Diagrama de clases.....	29
6.2 Diseño relacional de la base de datos.....	30
7. Arquitectura.....	31
7.1. Diagramas de componentes de “System”.....	31
7.2. Diagramas de componentes de “Aviary”.....	33
7.3. Diagramas de componentes de “Breeding”.....	35
7.4. Diagramas de componentes de “Contest”.....	37
8. Bibliografía.....	39

1. Motivación del TFG

En nuestra sociedad existen actividades que, en comparación con otras, tienen la consideración de actividades minoritarias, aunque estas provoquen una actividad considerable. Entre ellas tenemos la conocida como Canaricultura, rama de la avicultura que se ocupa de la cría y métodos de crianza del canario doméstico. Aunque en realidad, esta crianza del canario doméstico se extiende a otras especies de aves de jaula de enorme variedad. Aunque, por practicidad esta actividad ha mantenido esta nomenclatura simplificadora.

Como es lógico, los aficionados a esta actividad han utilizado durante los siglos más recientes sus propias técnicas de control y gestión de la cría para poder materializar una rigurosa selección genética de los ejemplares con los que trabajan. Y, de esta forma, hacer evolucionar las distintas razas de aves hacia los estándares previamente concebidos, con objeto de que algunos ejemplares puedan llegar a ser campeones en los distintos concursos en los que son inscritos.

Dicho esto, la funcionalidad básica y objetivo de la aplicación planteada consistirá en el control genético de las distintas aves en cría. Así como diversas funcionalidades de carácter estadístico que pueden ser de gran utilidad para los criadores a la hora de obtener ciertas conclusiones. Por otra parte, una interesante evolución de este proyecto consistirá en la posibilidad de que los criadores pudieran interactuar entre ellos y/o con las distintas asociaciones, especialmente con aquellas que organizan concursos. Y, de esta manera, facilitar la gestión de estos eventos, tanto para los criadores como para los organizadores.

También, otra interesante evolución del proyecto será la de implementar una interfaz de usuario sobre un dispositivo móvil. El motivo de ello es evidente, la mejor manera de introducir información, consultarla y, en definitiva, gestionarla es "in situ" y que mejor que una herramienta portátil. En definitiva, el objetivo final de este proyecto consiste en que el usuario pueda acceder y gestionar fácilmente la información de su aviario.

2. Objetivos

Los objetivos generales de este trabajo son:

- Facilitar la labor del usuario
- Profundizar en los conocimientos de la tecnología Java EE y familiarizarse con la utilización de la misma.
- Utilización de herramientas de código abierto.
- Familiarización con:
 - Gestión de proyectos.
 - La Ingeniería de requisitos.
 - El análisis y diseño con patrones.
 - Diseño de bases de datos.
 - La implementación de aplicaciones en red.

Para alcanzar los objetivos relacionados se hace necesaria la utilización de las siguientes tecnologías:

- **Entorno Eclipse:** entorno de desarrollo integrado de código abierto multiplataforma que permitirá desarrollar “Aplicaciones de cliente enriquecido”.
- **Google Plugin for Eclipse (GPE):** Conjunto de herramientas que permite a los desarrolladores de Java diseñar, construir, optimizar y desarrollar aplicaciones de computación en nube..
- **Google Web Toolkit (GWT):** Framework de Google para ocultar algunos aspectos de la tecnología AJAX, compatible con distintos navegadores creando el código en Java.
- **Google App Engine (GAE):** Servicio de alojamiento web que presta Google de forma gratuita con limitaciones.
- **Patrón Modelo – Vista – Presentador (MVP):** Derivación del patrón arquitectónico modelo–vista–controlador (MVC) y se utiliza para construir interfaces de usuario.
- **Persistencia de datos:** Java EE ha desarrollado el API JPA para este menester.
- **MySQL y/o SQLite:** motores de base de datos.
- **Software adicional:** MagicDraw, LibreOffice, GanttProject.
 -

En cuanto a los objetivos personales, podemos decir que básicamente se trata de que el ponente sintetice todo lo aprendido durante el grado. Y, en este sentido hay que decir que ciertamente ha sido mucho lo aprendido. Pero, lo que ha quedado más claro es lo mucho que puede aprenderse. Es decir, no solamente lo que que por profundizar en el material impartido durante el grado, sino el camino a recorrer en todas las tecnologías “periféricas” (frameworks,...) que son imprescindibles para la materialización de cualquier proyecto hoy en día.

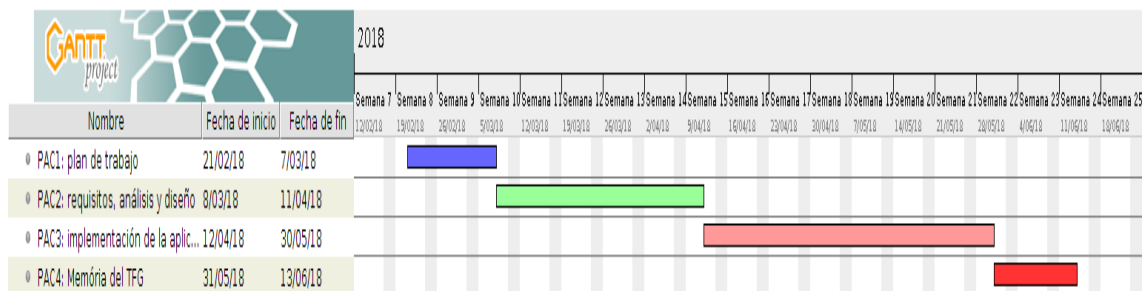
3. Planificación y programación

Para la planificación temporal del proyecto se hace uso de GanttProject que es un programa de código abierto con licencia GPL escrito en Java con la biblioteca Swing, cuyo objetivo es la administración de proyectos mediante la utilización de diagramas de Gantt.

Resumen de fechas clave del proyecto:

Entregas	Documento	Descripción
07/03/10	PAC1	Elaboración del plan de trabajo, estructura general del proyecto y su planificación.
11/04/10	PAC2	Estudio de la arquitectura, análisis y diseño del proyecto. Instalación y configuración del entorno de programación.
30/05/10	PAC3	Implementación y codificación del proyecto.
13/06/10	Memoria	Fecha de finalización del proyecto con la entrega de la memoria, presentación final y producto final.

Diagrama de Gantt con la planificación temporal de las tareas del proyecto:



4. Análisis

4.1. Descripción del problema

Ornithoculture pretende ser un sistema que ayude en el control de la cría de aves de jaula para los criadores aficionados de esta actividad. Así, los criadores dispondrán de una herramienta que les permita registrar e identificar detalladamente a sus animales de forma genérica. Y además, les permita gestionar información sobre la participación de estos animales en certámenes ornitológicos, los resultados obtenidos y los jueces que han proporcionado estos resultados. Por tanto, en **Ornithoculture** se hace necesario disponer de un control de la cría a los efectos de poder materializar la correspondiente selección de razas, la cual cosa está directamente relacionada con la participación y resultados de los mencionados certámenes. Finalmente, **Ornithoculture** llevará un registro de usuarios con la información de estos como criadores que previamente deberán serlo.

Con todo, diremos que **Ornithoculture** se divide en cinco grandes bloques:

- **Gestión del sistema:** en este apartado se manejará la información propia del sistema, tales como el registro de usuarios, comentado anteriormente. Y, los datos mínimos necesarios de estos usuarios serán: nombre de usuario, contraseña y correo electrónico. Eso sí, siendo que estos usuarios debe ser previamente criadores, necesitarán facilitar la información relativa a esta circunstancia. Este proceso se llevará a cabo en el registro de usuarios del sistema. Por otra parte, los usuarios que ya se hayan registrado podrán acceder al sistema y sus funcionalidades.
- **Gestión de aves:** en este bloque se maneja la información básica de las aves y sus respectivos criadores. Es decir, un aviario puede contener aves criadas por el propio criador y, también, aves procedentes de otros aviarios, con el objetivo de introducir características deseadas en dicho aviario. Por ello, se hace necesario el manejo de información concreta de aves y criadores.

Para empezar tenemos a los criadores con los datos habituales de una filiación más el número identificador de criador. Y, el usuario podrá realizar las funciones típicas de gestión de datos, tales como: añadir, editar y eliminar criadores.

En lo referente a las aves es importante poder controlar la existencia de las mismas en el aviario (stock) y su ausencia, origen, destino, fechas de entrada y salida del animal en el mismo, fecha de nacimiento (al menos el año de nacimiento), número de identificación (anilla) y el de su criador. Opcionalmente, se puede llevar un control de la jaula donde está ubicado el animal dentro del aviario. Y, en lo referente al sexo del animal se contemplarán cinco opciones: desconocido, hembra dudosa, macho dudoso, hembra y macho. E igual que sucedía con el registro de criadores, el usuario podrá: añadir, editar y eliminar sujetos del aviario. Eso sí, teniendo en cuenta que el criador del animal deberá estar registrado previamente a la creación del registro del animal.

Como se ha dicho, opcionalmente se puede ubicar un ave en una de la jaulas del aviario, disponiendo de un identificador alfanumérico de cada una de estas jaulas y una descripción de la misma. Por lo que el usuario gestionara: añadir, editar y eliminar jaulas, teniendo en cuenta que antes asignar la ubicación de cualquier animal a una de las jaulas será necesario haber generado el registro de la misma.

- **Gestión de cría:** en este bloque se gestiona la información de la cría y su proceso. Esto implica una dependencia sobre el bloque anterior. O sea, no tiene sentido gestionar la información de un proceso de cría sin la información previa del aviario. Como se ha dicho, este bloque es de gran importancia para el aficionado al objeto de utilizarlo como herramienta para la selección de especímenes. Además, se hace importante la información estadística del proceso de cría, como herramienta de detección de errores o posibles anomalías de distinta índole, dígame: problemas de fertilidad, errores de la técnica de cría del aficionado, etc.

Por tanto, en este apartado dispondremos de información sobre las parejas de aves que se han ido generando a largo del tiempo y cuyo número identificativo será permanente. Lógicamente, cada pareja estará compuesta por un ave de sexo masculino y otra de sexo femenino y podrá controlarse el estado actual de cada pareja en el sentido de si está criando en ese momento o no. Además, será necesaria la asignación de la jaula en que la pareja está ubicada en el momento de la cría y donde no será posible la ubicación de más de una pareja. También en esta ocasión el usuario podrá: añadir, editar y eliminar parejas, considerando que para el registro de una nueva pareja será necesario haber registrado previamente a los componentes de la misma.

De otra parte tendremos los resultados de cada una de las parejas puestas en cría. Es decir, cada pareja irá generando ciertas nidadas, de las cuales será conveniente controlar el número total de huevos, el número de huevos fértiles, número de huevos eclosionados, fecha de inicio de incubación, fecha de la eclosión. Y, todo ello estará relacionado con las aves que, tras el proceso de la cría, serán adultos totalmente identificados y vinculados con sus respectivos progenitores. Siguiendo la tónica de lo visto hasta ahora, el usuario podrá: añadir, editar y eliminar nidadas, teniendo presente que la creación de una nueva nidada está supeditada a que la pareja generadora exista previamente en el sistema.

De igual forma que hemos visto en la gestión de aves, cada pareja podrá ser ubicada en una jaula, previamente registra, del aviario y el usuario dispondrá de los mismos privilegios vistos anteriormente (añadir, editar y eliminar). Eso sí, con la salvedad de que al asignar una jaula a una pareja esta jaula no podrá ser asignada a ninguna otra pareja. Esto es porque la asignación de jaula servirá como condicionante del estado de dicha pareja, en el sentido de que se encuentre en cría o no.

- **Gestión de certámenes:** la información tratada por este bloque es la que implica todo lo relacionado con la competición de las aves. Por tanto, uniéndola a la del apartado anterior el criador dispone de argumentos útiles para la mencionada selección de especímenes.

De un lado tenemos las estructuras organizativas que son las asociaciones de criadores que, al agruparse entre ellas, conforman las federaciones de distintos ámbitos. Tanto unas como otras suelen ser las organizadoras de los referidos certámenes y (aunque para los que nos ocupa en estos momentos no es relevante) los dirigentes de las mismas son personas que no necesitan ser criadores obligatoriamente. Por tanto, sólo necesitamos el cargo que ostenta, además de los datos mínimos de su filiación.

Tanto de las organizaciones como de sus posibles representantes el usuario podrá: añadir, editar y eliminar registros, considerando que cada asociación necesitará al menos un cargo vinculado a esta (puede tratarse simplemente de un contacto). Y, de igual forma, los cargos generados en el sistema lo serán al menos de una organización (no tiene sentido en este sistema la generación de cargos que no estén vinculados a alguna organización). Además, hay que considerar que puede disponerse de la información que vincula a los criadores conocidos con su respectiva asociación o asociaciones. Pero, como esta información no siempre está disponible públicamente, parece recomendable que la relación entre criadores y asociaciones sea algo opcional y no vinculante.

De otro lado tenemos el concurso que será, como hemos dicho, organizado obligatoriamente por una asociación o federación. Y, necesitamos conocer el nombre del concurso, la fecha de inicio del mismo, su fecha de finalización y su ubicación. Además, es imprescindible que el concurso esté integrado por grupos de competición conocidos como secciones, al menos por una sección. En cuanto a las posibilidades del usuario en la gestión de esta información son: añadir, editar y eliminar concursos, conociendo que para generar un nuevo concurso es necesario que la entidad organizadora esté registrada previamente y el juez o jueces que van participar en el mismo también (imprescindible al menos un juez). Por lo que se refiere a las secciones

de participación, estas son algo tasado. Es decir, el sistema dispondrá de esta información previa.

Ahora, se hace necesaria la observación de los jueces del certamen, los cuales es obligatorio que sean criadores antes de llegar a ser jueces. Y, lo importante de estos personajes es que sin ellos no es posible la materialización del concurso. Por tanto, la información básica de estos jueces será su categoría (nacional o internacional) y la sección o secciones en que son competentes. Ello significa que los jueces asignados a un concurso deberán ser competentes en la sección o secciones del éste. Como ya suponemos, el usuario podrá; añadir, editar y eliminar jueces, sabiendo que para ser juez es necesario ser criador antes. Por ello, tendrá que existir como criador en el sistema antes de darle de alta como juez.

Finalmente, sólo nos quedan las inscripciones a los concursos. Es decir, cada animal podrá ser, o no, inscrito a uno o varios certámenes durante la temporada de competición. Para ello, es necesario un registro de estas inscripciones con la correspondiente vinculación entre el ave y concurso. Además, debemos conocer el juez que ha evaluado al animal (de entre los jueces participantes del certamen), el modo de competición (individual o equipos), el resultado obtenido (puntuación) y, opcionalmente, galardón obtenido y número de jaula. Es claro que, el usuario podrá: añadir, editar y eliminar inscripciones, considerando que, a la hora de generar una nueva inscripción, el certamen y el animal deben estar registrados previamente.

- **Gestión de datos:** finalmente, se hace necesaria la utilización de toda la información generada en forma de resultado y como herramienta de trabajo para el futuro. Por tanto, se necesitarán herramientas tales como:
 - Un informe del árbol genealógico de cada animal, donde se recoja toda la ascendencia del mismo.
 - Un informe con el que se puedan analizar los datos de la cría anual, con números como: total parejas en cría, total de nidadas, total de huevos, total de huevos fértiles, total de huevos eclosionados y total de aves criadas (por lo menos hasta ser identificadas con su correspondiente anilla).
 - Un informe de la competición anual con los datos de participación en eventos y los resultados obtenidos: número de eventos, número de inscripciones, puntuación media y número de galardones.
 - Una relación de las aves galardonadas con los galardones obtenidos.

4.2. Enumeración de requisitos de producto

4.2.1 Requisitos funcionales

Como sabemos, son aquellos aquellos que establecen el comportamiento del sistema y los datos que debe manejar. En nuestro caso, detallamos las funcionalidades descritas en el apartado “2. Descripción del problema”, que no es otra cosa que un resumen de la información recopilada de los diversos stakeholders.

- **Gestión del sistema**
 - Registro de usuarios:
 - Para ser usuario es necesario ser criador.
 - Datos: id, nombre, apellidos (al menos el primero), fecha de nacimiento (opcional), número identificativo de criador, teléfono (opcional, uno o varios), dirección (opcional, una o varias), asociación o asociaciones de las que es miembro (opcional), nombre de usuario, correo electrónico y contraseña.
 - Cada criador sólo podrá registrarse una vez (no se admitirán criadores duplicados).
 - Acceso de usuarios (Login):
 - El usuario deberá haberse registrado previamente en el sistema.
 - Datos: nombre de usuario y contraseña.
 - Acciones permitidas al usuario: acceder al sistema, actualizar datos de su cuenta y acceder todas las funcionalidades de gestión del aviario (cría y certámenes incluidos).
 - Deberán coincidir el nombre de usuario, y su contraseña, con los datos registrados para permitir el acceso.
 - Gestión de cuenta de usuario:
 - Datos: fecha de nacimiento (opcional), teléfono (opcional, uno o varios), dirección (opcional, una o varias), asociación o asociaciones de las que es miembro (opcional), correo electrónico y contraseña.
 - Acciones permitidas al usuario: actualizar los anteriores datos de su cuenta.
 - El usuario deberá estar identificado en el sistema (logged).
- **Gestión de aves**
 - Gestión de criadores:
 - Datos: id, nombre, apellidos (al menos el primero), fecha de nacimiento (opcional), número identificativo de criador, teléfono (opcional, uno o varios), dirección (opcional, una o varias) y asociación o asociaciones de las que es miembro (opcional).
 - Acciones permitidas al usuario: añadir, editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
 - Gestión de teléfonos:
 - Datos: id, tipo, número de teléfono y si se trata del predeterminado.
 - Los teléfonos no pueden existir sin estar vinculados a una filiación.
 - Acciones permitidas al usuario: añadir, editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
 - Gestión de direcciones:

- Datos: id, dirección, localidad, código postal, Comunidad autónoma (opcional), país y si se trata de la dirección predeterminada.
 - Las direcciones no pueden existir sin estar vinculadas a una filiación.
 - Acciones permitidas al usuario: añadir, editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
 - Gestión de aves:
 - Es necesario conocer en todo momento las aves en stock (puede ser un atributo derivado).
 - Debe poder registrarse aves criadas por el propio criador y/o aves procedentes de otros aviarios.
 - Datos: id, número de su criador, número de identificación (anilla), fecha de nacimiento (al menos el año de nacimiento), fecha del fallecimiento (si ha lugar), sexo del animal, opciones: desconocido, hembra dudosa, macho dudoso, hembra, macho, fecha de entrada (cuando se trata de ave de origen externo), origen, fecha de salida y jaula donde está ubicado (opcional y si la jaula no existe, se deberá permitir la creación de la misma).
 - Acciones permitidas al usuario: añadir (si el criador del animal no existe, deberá permitirse el registro del mismo), editar, eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
 - Gestión de jaulas:
 - Datos: identificador alfanumérico, descripción.
 - Acciones permitidas al usuario: añadir, editar, eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
- **Gestión de cría**
(este bloque tendrá dependencia total del bloque "Gestión de aviario")
 - Gestión de parejas:
 - Datos: número identificativo (permanente), componentes: un ave de sexo masculino y otra de sexo femenino, estado de la pareja: se hace necesario conocer en todo momento el estado de cada pareja (modo cría o no), para ello se utilizará la asignación de la correspondiente jaula de cría. Es decir, si la pareja tiene asignada una jaula de cría implicará que la pareja está en modo cría. Así, no será posible asignar más de una pareja a la misma jaula de cría.
 - Acciones permitidas al usuario: añadir (si alguno de los componentes de la pareja no existe, será necesario el sistema permita añadirlo), editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
 - Gestión de nidadas:
 - Datos: id, número total de huevos; número de huevos fértiles; número de huevos eclosionados; fecha de inicio de incubación; fecha de la eclosión: los huevos eclosionados que alcancen la edad suficiente como para ser identificados plenamente (anillado) generarán registros de aves ("Gestión de aves") que estarán vinculados a la correspondiente nidada. Y, lógicamente, a sus progenitores.
 - Acciones permitidas al usuario: añadir (si la pareja generadora no existe, el sistema permitirá la creación de dicha pareja), editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
- **Gestión de certámenes**

- Gestión de asociaciones:
 - Las asociaciones agrupadas conforman federaciones.
 - Tanto unas como otras son organizadoras de certámenes.
 - Datos: id, nombre oficial, nombre coloquial (opcional), fecha de creación (opcional), tipo (asociación/federación), teléfono (opcional, uno o varios), dirección (opcional, una o varias) y asociación o asociaciones de las que es miembro (opcional).
 - Acciones permitidas al usuario: añadir (necesitará al menos un cargo vinculado a la misma, pudiendo tratarse de un simple contacto), editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
- Gestión de cargos:
 - Los cargos no tienen porque ser criadores.
 - Datos: nombre, apellidos (al menos el primero), fecha de nacimiento (opcional), teléfono (opcional, uno o varios), dirección (opcional, una o varias), cargo que ostenta y la organización en la que lo ostenta (uno o varios cargos).
 - Acciones permitidas al usuario: añadir (los cargos generados en el sistema lo serán al menos de una organización), editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
- Gestión de concursos:
 - Datos: id, nombre del concurso, fecha de inicio, fecha de finalización, ubicación, secciones de competición (al menos por una) jueces participantes (al menos uno) y entidad organizadora.
 - Acciones permitidas al usuario: añadir (la entidad organizadora y el juez o jueces participantes deben existir en el sistema o deben poder crearse y los jueces participantes, estos han de ser competentes en las secciones de competición del evento), editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
- Gestión de jueces:
 - Es obligatorio que sean criadores antes de llegar a ser jueces (datos criador).
 - Datos: categoría (nacional o internacional), sección o secciones de su competencia.
 - Acciones permitidas al usuario: añadir (debe existir en el sistema como criador o debe poder generarse), editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
- Gestión de inscripciones a los concursos:
 - Cada animal podrá ser inscrito a uno o varios certámenes (nunca dos veces en el mismo).
 - Datos: id, identificación del animal, identificación del concurso, juez que lo a calificado (de entre los jueces participantes en el certamen), modo de competición (individual o equipos), resultado obtenido (puntuación), galardón obtenido (opcional), número de jaula (opcional) y concurso al que ha sido inscrito.
 - Acciones permitidas al usuario: añadir: (el certamen y el ave deben existir en el sistema o ser generados previamente), editar y eliminar.
 - El usuario deberá estar identificado en el sistema (logged).
- **Gestión de la información**

- Árbol genealógico: informe sobre la ascendencia de cualquier ave del aviario.
- Resultado de la cría: informe sobre la cría anual.
 - Datos:
 - Número total de parejas utilizadas.
 - Número total de nidadas.
 - Número total de huevos.
 - Número total de huevos fértiles.
 - Número total de huevos eclosionados.
 - Número de individuos criados e identificados.
 - Acciones permitidas al usuario: consultar.
 - El usuario deberá estar identificado en el sistema (logged).
- Resultado de la competición: informe anual de la competición.
 - Datos:
 - Número total de eventos en los que se ha participado.
 - Número total de inscripciones.
 - Media de las puntuaciones obtenidas.
 - Número total de galardones obtenidos.
 - Acciones permitidas al usuario: consultar.
 - El usuario deberá estar identificado en el sistema (logged).
- Relación de galardonados: informe nominal de los galardones de la temporada.
 - Acciones permitidas al usuario: consultar.
 - El usuario deberá estar identificado en el sistema (logged).

4.2.2 Requisitos no funcionales

Estos requisitos son aquellos que se refieren a las calidades esperadas por el proyecto. En definitiva, son restricciones a la solución del problema que, de no cumplirse, pueden invalidar la solución referida. Por ello, debemos hacer notar que lo referido a continuación forma parte del objetivo final de este proyecto, aún cuando la mayor parte de este objetivo no está reflejado todavía en la implementación del mismo.

- Requisitos operacionales y de entorno:

El usuario debe de poder acceder a la aplicación desde su teléfono móvil: este requisito es sumamente importante en este proyecto y nos referimos a él en primer lugar porque la idea del proyecto se basa precisamente en la importancia de este hecho. Es decir, el primer problema que tiene un criador a la hora de almacenar información sobre su criadero es el inconveniente que supone tener que tomar notas para luego trasladarlas a un ordenador o cualquier otro sistema de los utilizados hasta ahora. En cambio, la idea de que el criador pueda utilizar su teléfono móvil "in situ" es la clave principal para facilitar este trabajo de manejo de información. Además, no sólo la comodidad de introducir información de forma directa y al momento, sino el hecho de poder acceder a la misma desde su criadero, mientras observa la situación de su interés.

- Requisitos de usabilidad y humanidad:

El proceso de aprendizaje de los usuarios deberá ser sencillo y cómodo: se trata de un requisito muy importante. Pues, si el manejo de la aplicación se hace complicado o complejo, esto puede desanimar a la utilización de la misma por parte de los usuarios.

- Requisitos culturales y políticos:

Inicialmente, la aplicación será utilizable en los idiomas: castellano, catalán e inglés. Ni que decir tiene la importancia de que los usuarios puedan utilizar la aplicación en el idioma que más les pueda interesar.

- Requisito de seguridad y legales:

Los datos de usuario estarán protegidos y las comunicaciones entre nodos serán encriptadas: los usuarios no podrán ver datos de otros usuarios (LOPD).

- Requisitos de mantenimiento y soporte:

Si se da la situación de que el número de usuarios crece considerablemente, debe poder ampliarse la admisión de los mismos de forma transparente para estos.

- Requisitos de cumplimiento:

En el proceso de implementación se tendrá en consideración la rapidez y fiabilidad del sistema. Eso sí, para el registro y la identificación de los usuarios se requerirá conexión Internet. Aunque, una vez identificado el usuario esta conexión no será imprescindible.

5. Especificación

En este apartado desarrollaremos el trabajo de análisis realizado anteriormente. Así, identificaremos a los actores del sistema y los casos de uso vinculados a estos. En primer lugar, presentaremos el diagrama de estos actores, seguido de un diagrama global de Ornithoculture, con el objetivo de tener una visión amplia del objetivo. Seguidamente, aportamos un índice de casos de uso que facilitará la localización de los mismos, acompañado de un sumario de los actores principales de los casos de uso. Y, finalmente, una relación de todos los casos de uso: diagrama de caso de uso y ficha de caso de uso, básicamente.

Hay tener en cuenta que el nivel de granularidad de estos casos de uso no es del todo fino. Pero, con los ejemplos aportados en el bloque de “System management: activity diagrams” y “Birds management: add, update and delete bird” queda suficientemente explicada la dinámica del sistema en ambos sentidos.

Diagrama de “Actors”

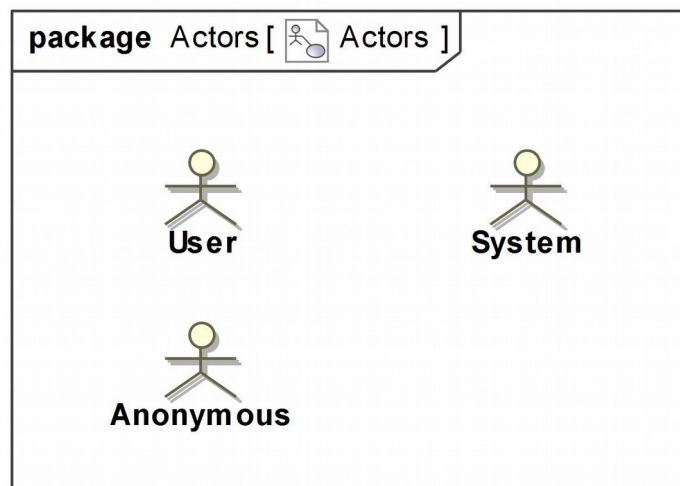
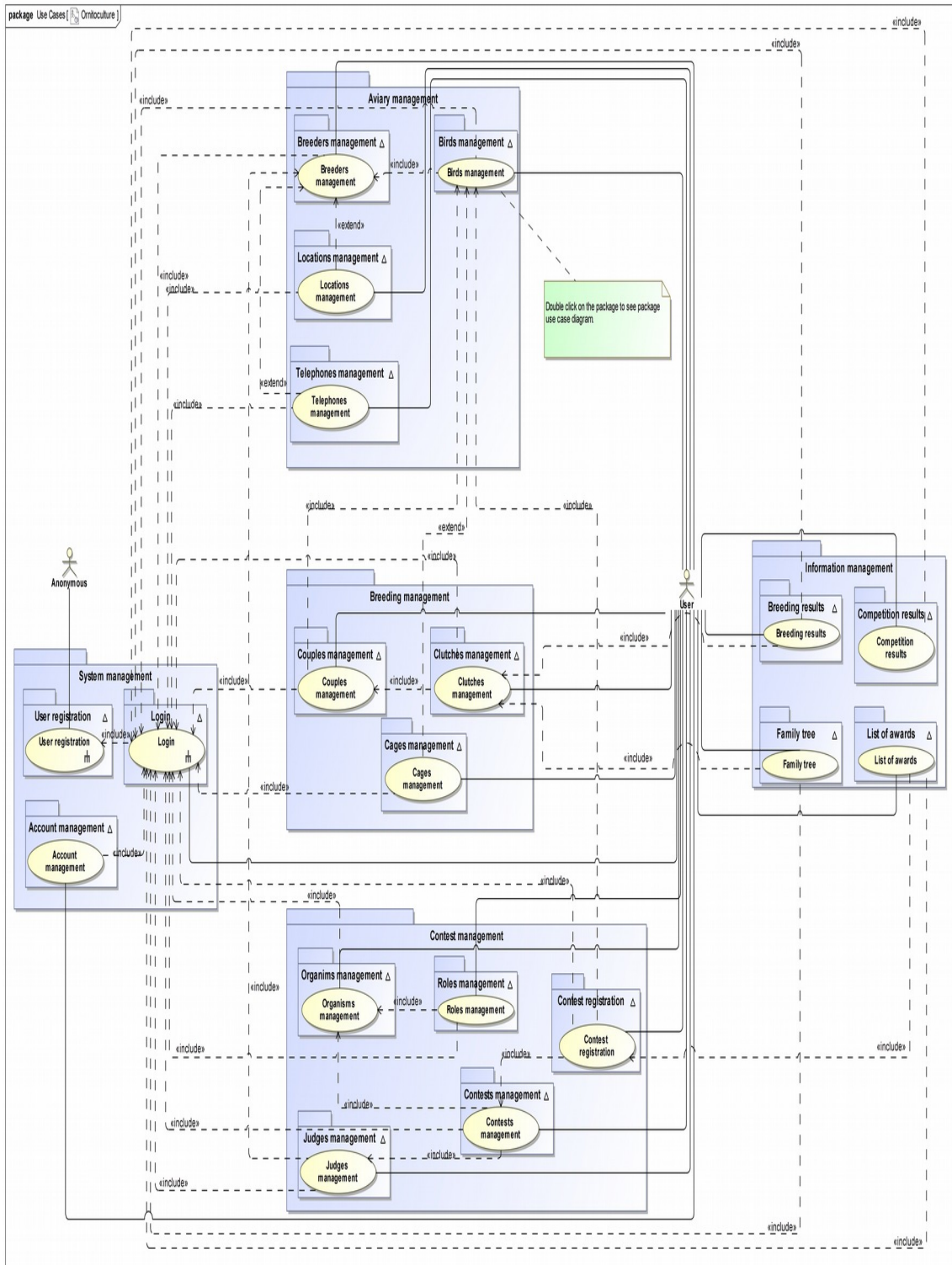


Diagrama de "Ornithoculture"



5.1 Índice de casos de uso

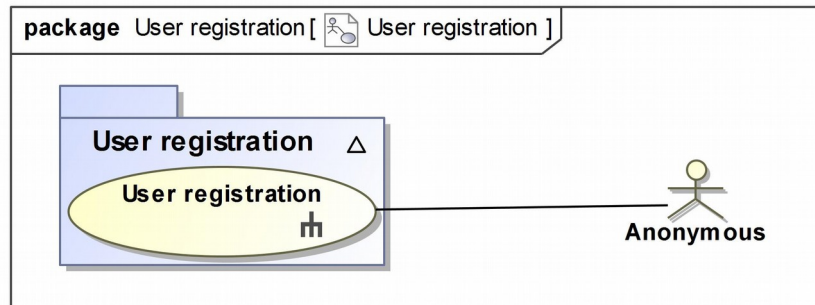
Caso de Uso	ID	Página
Account management	UC03	18
Birds management	UC07	21
Breeders management	UC04	19
Breeding results	UC16	27
Cages management	UC08	22
Clutches management	UC10	23
Competition results	UC19	28
Contest registration	UC15	26
Contests management	UC14	25
Couples management	UC09	22
Family tree	UC17	27
Judges management	UC13	24
List of awards	UC18	28
Locations management	UC05	20
Login	UC02	17
Organisms management	UC11	23
Roles management	UC12	24
Telephones management	UC06	20
User registration	UC01	16

5.2 Sumario de actores

Actor principal	Casos de Uso
Anonymous	<ul style="list-style-type: none"> User registration
User	<ul style="list-style-type: none"> Account management Birds management Breeders management Breeding results Cages management Clutches management Competition results Contest registration Contests management Couples management Family tree Judges management List of awards Locations management Login Organisms management Roles management Telephones management
System	<ul style="list-style-type: none">

5.3 Casos de uso

Diagrama de “User registration”



Caso de Uso: “User registration”

Nombre	User registration	ID	UC01
Descripción	This use case manages the process of user's registration, according to Ornithoculture system description in system management.		
Actores	<ul style="list-style-type: none"> • Anonymous • System 		
Precondición	User cant be registered.		
Postcondición	The user has been registered.		
Restricciones	<ul style="list-style-type: none"> • User must be a breeder • Just one breeder registration is allowed 		

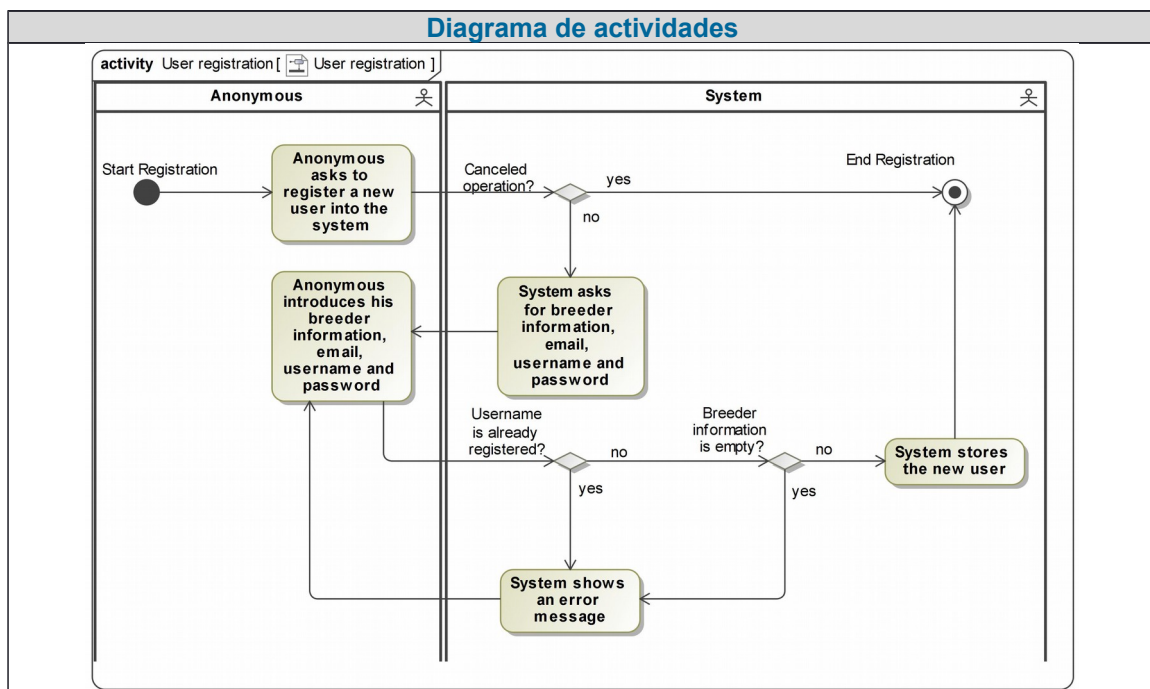
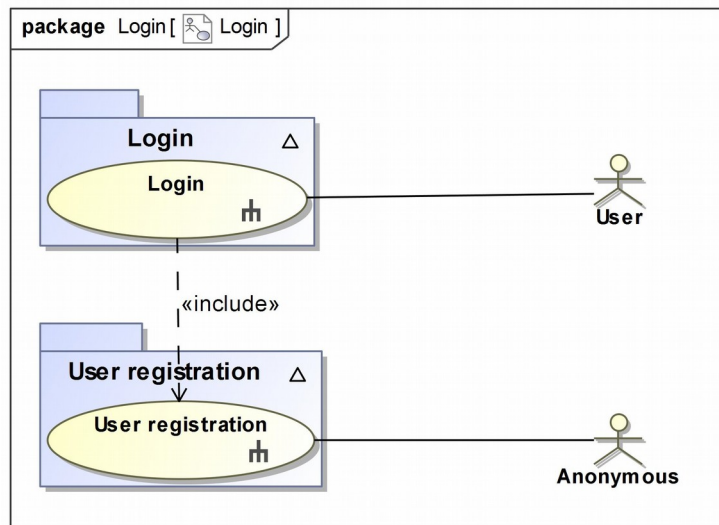


Diagrama de “Login”



Caso de Uso: “Login”

Nombre	Login	ID	UC02
Descripción	This use case manages the process of user's login, according to Ornithoculture system description in system management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be registered.		
Postcondición	The user has been logged.		

Diagrama de actividades

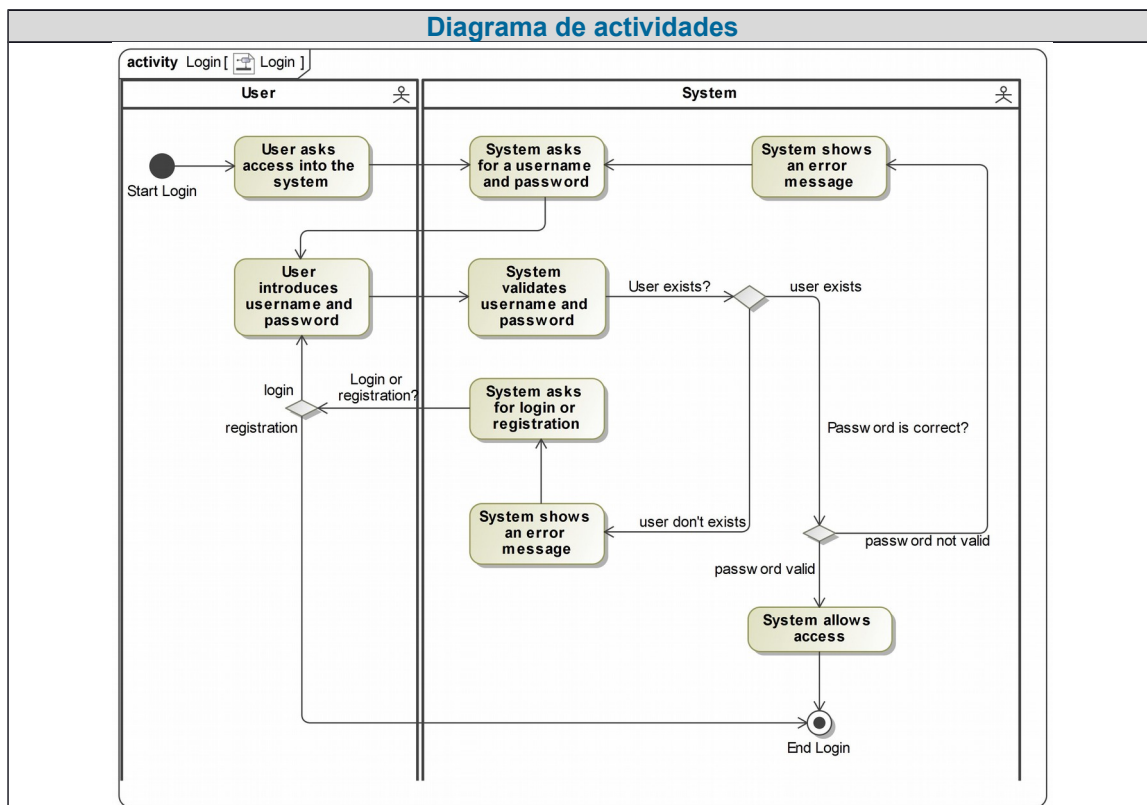
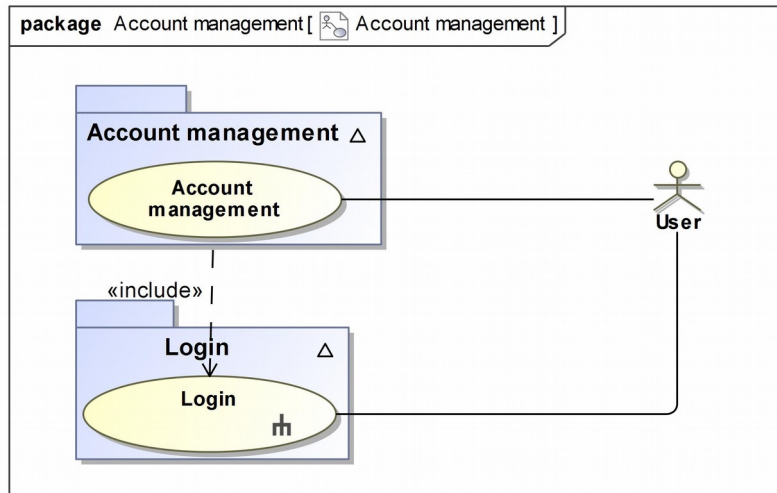


Diagrama de “Account management”



Caso de Uso: “Account management”

Nombre	Account management	ID	UC03
Descripción	This use case manages the user's account information, according to Ornithoculture system description in system management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged.		
Postcondición	The user's account has been modified.		

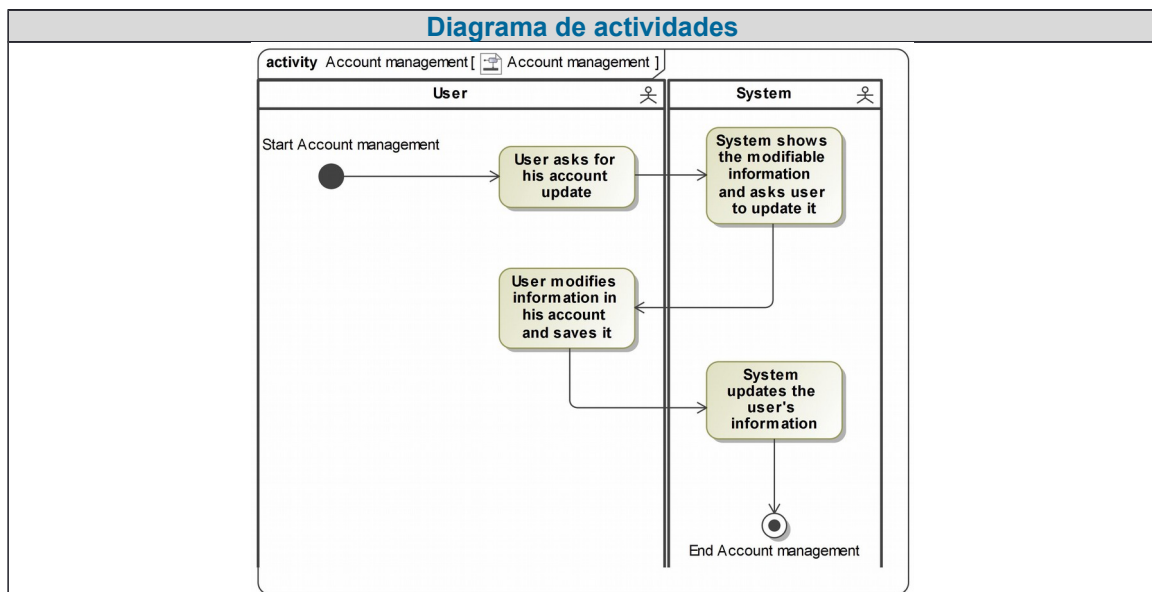
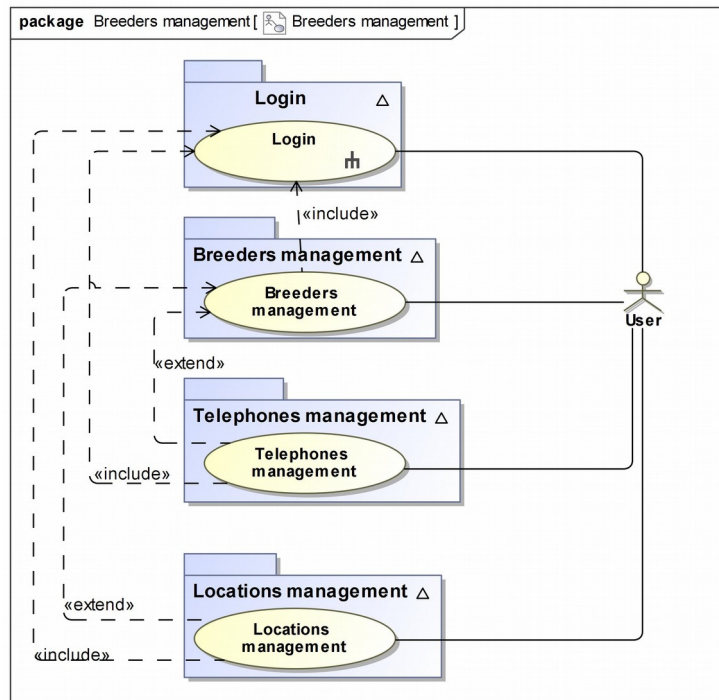


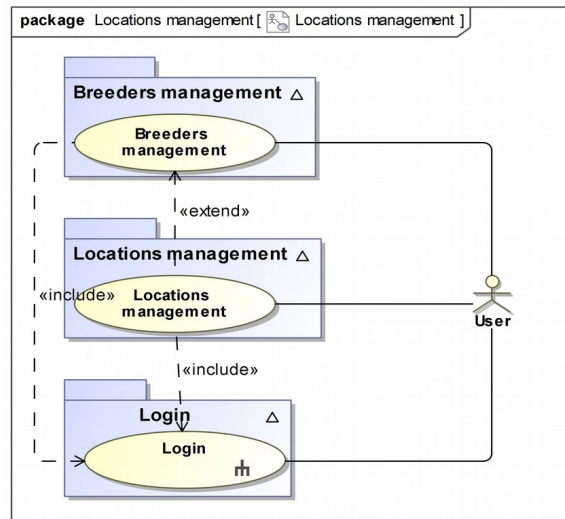
Diagrama de “Brederers management”



Caso de Uso: “Brederers management”

Nombre	Brederers management	ID	UC04
Descripción	This use case manages the breeder's information, according to Ornithoculture system description in Aviary management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged.		
Postcondición	Breeder has been registered, updated or deleted.		

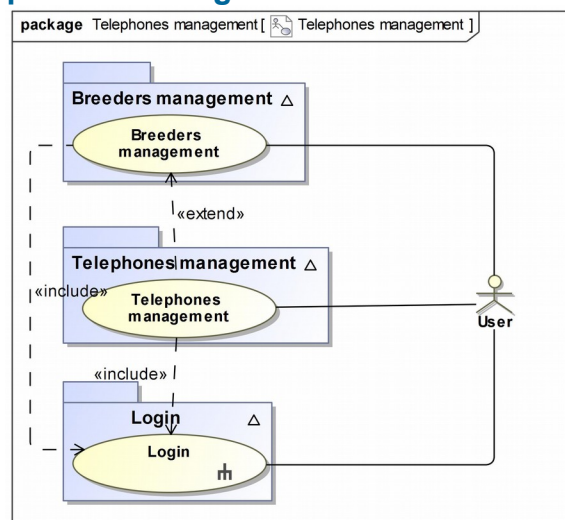
Diagrama de “Locations management”



Caso de Uso: “Locations management”

Nombre	Locations management	ID	UC05
Descripción	This use case manages the location's information, according to Ornithoculture system description in Aviary management.		
Actores	<ul style="list-style-type: none"> • User 		
Precondición	User must be logged; the breeder should be registered.		
Postcondición	Location has been registered, updated or deleted.		

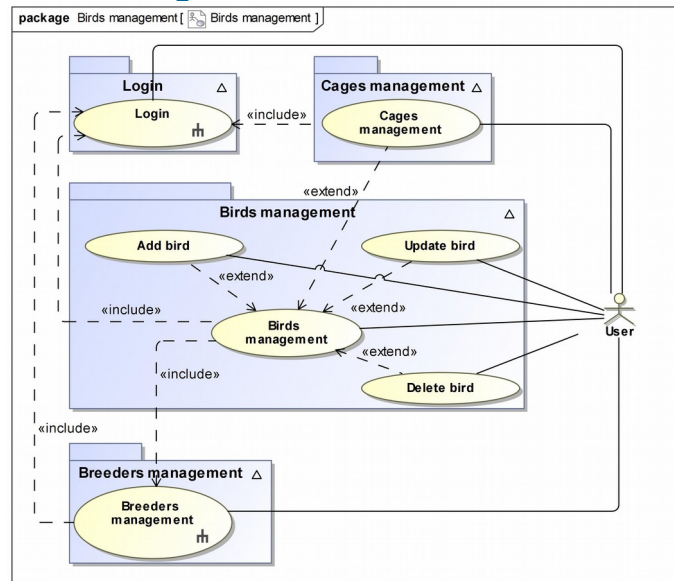
Diagrama de “Telephones management”



Caso de Uso: “Telephones management”

Nombre	Telephones management	ID	UC06
Descripción	This use case manages the telephone's information, according to Ornithoculture system description in Aviary management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; the breeder should be registered.		
Postcondición	Telephone has been registered, updated or deleted.		

Diagrama de “Birds management”



Caso de Uso: “Birds management”

Nombre	Birds management	ID	UC07
Descripción	This use case manages the bird's information, according to Ornithoculture system description in Aviary management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; the bird's breeder should be registered.		
Postcondición	Bird has been registered, updated or deleted.		

Caso de Uso: “Add bird”

Nombre	Add bird	ID	UC07a
Descripción	This use case manages the bird's information, according to Ornithoculture system description in Birds management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	The bird's breeder should be registered.		
Postcondición	Bird has been added.		

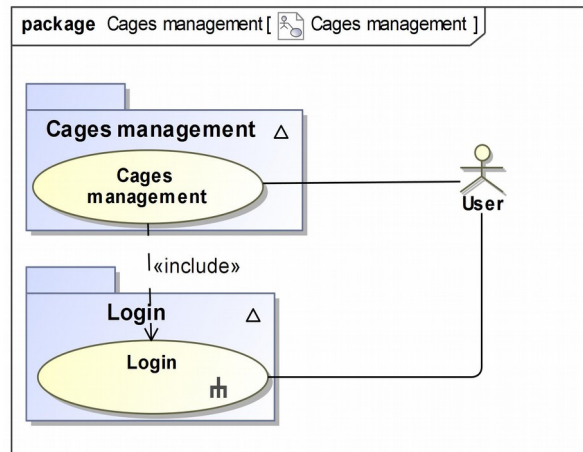
Caso de Uso: “Update bird”

Nombre	Update bird	ID	UC07b
Descripción	This use case manages the bird's information, according to Ornithoculture system description in Birds management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	The bird's breeder should be registered.		
Postcondición	Bird has been updated.		

Caso de Uso: “Delete bird”

Nombre	Delete bird	ID	UC07c
Descripción	This use case manages the bird's information, according to Ornithoculture system description in Birds management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	true		
Postcondición	Bird has been deleted.		

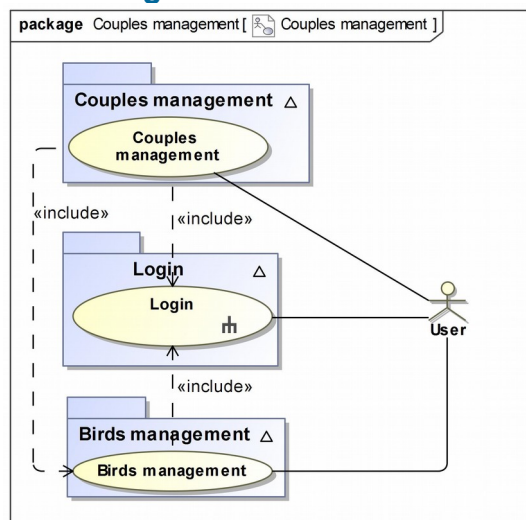
Diagrama de “Cages management”



Caso de Uso: “Cages management”

Nombre	Cages management	ID	UC08
Descripción	This use case manages the cage's information, according to Ornithoculture system description in Aviary management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged.		
Postcondición	Cage has been registered, updated or deleted.		

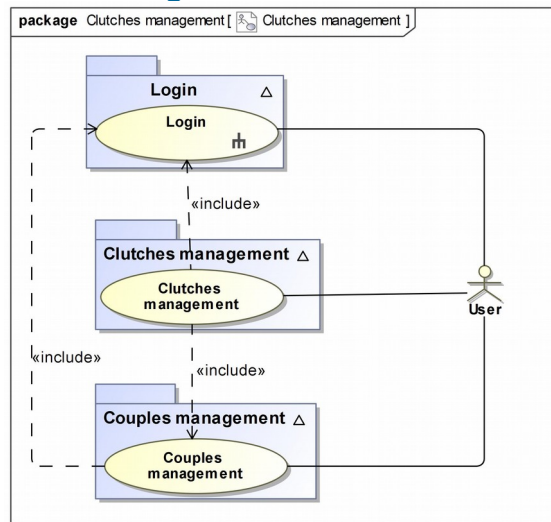
Diagrama de “Couples management”



Caso de Uso: “Couples management”

Nombre	Couples management	ID	UC09
Descripción	This use case manages the couple's information, according to Ornithoculture system description in Aviary management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; the couple components should be registered; the couple's cage must be registered.		
Postcondición	Couple has been registered, updated or deleted.		

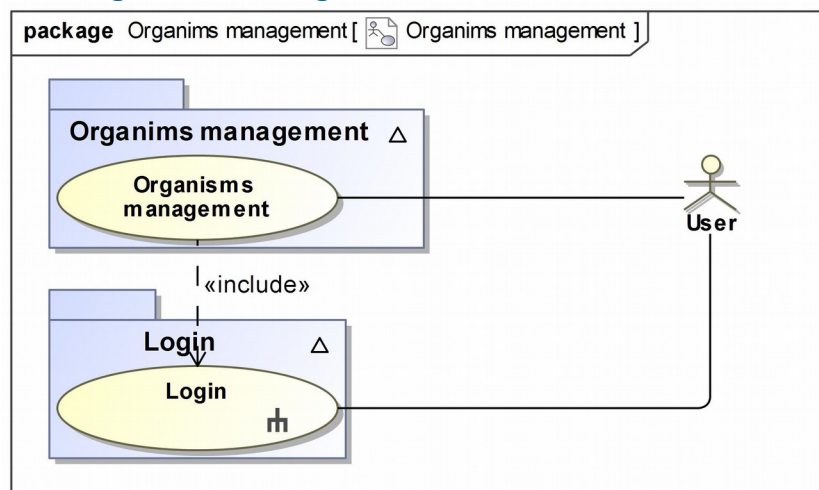
Diagrama de “Clutches management”



Caso de Uso: “Clutches management”

Nombre	Clutches management	ID	UC10
Descripción	This use case manages the clutch's information, according to Ornithoculture system description in Aviary management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; the couple who produces the clutch must be registered.		
Postcondición	Clutch has been registered, updated or deleted.		

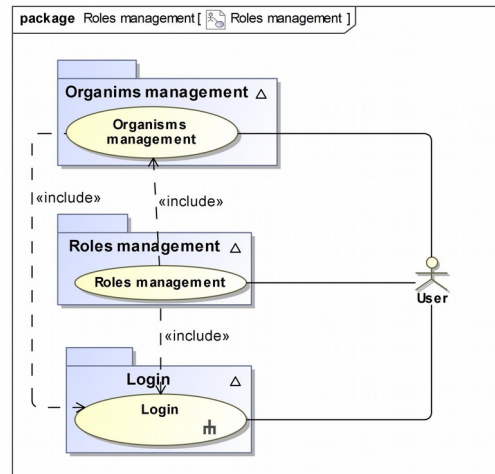
Diagrama de “Organisms management”



Caso de Uso: “Organisms management”

Nombre	Organisms management	ID	UC11
Descripción	This use case manages the organism's information, according to Ornithoculture system description in Contest management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged.		
Postcondición	Organism has been registered, updated or deleted.		

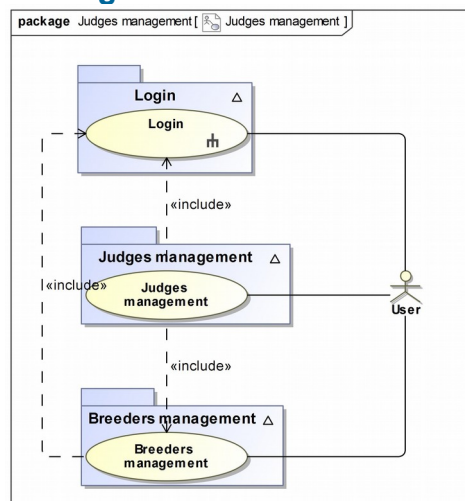
Diagrama de “Roles management”



Caso de Uso: “Roles management”

Nombre	Roles management	ID	UC12
Descripción	This use case manages the role's information, according to Ornithoculture system description in Contest management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; organism should be registered.		
Postcondición	Role has been registered, updated or deleted.		

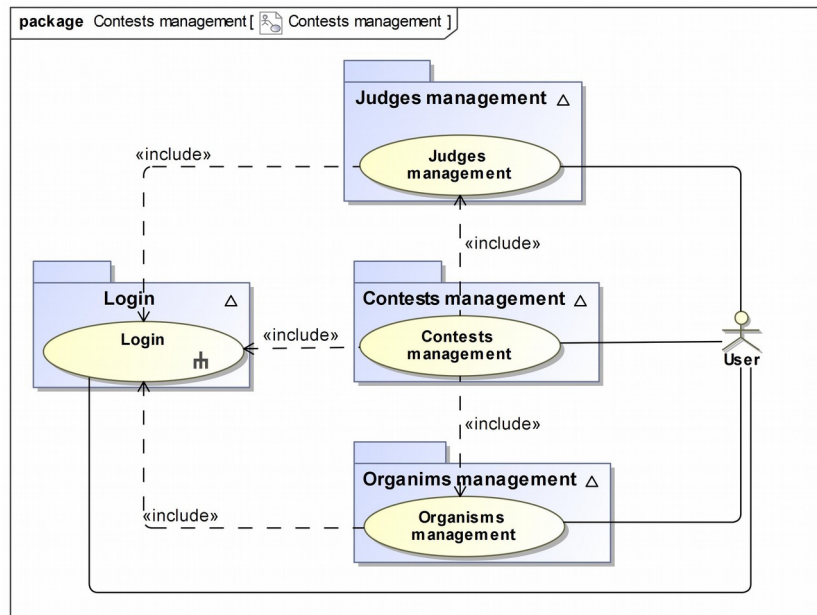
Diagrama de “Judges management”



Caso de Uso: “Judges management”

Nombre	Judges management	ID	UC13
Descripción	This use case manages the judge's information, according to Ornithoculture system description in Contest management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; the breeder should be registered (To become a judge should be breeder first).		
Postcondición	Judge has been registered, updated or deleted.		

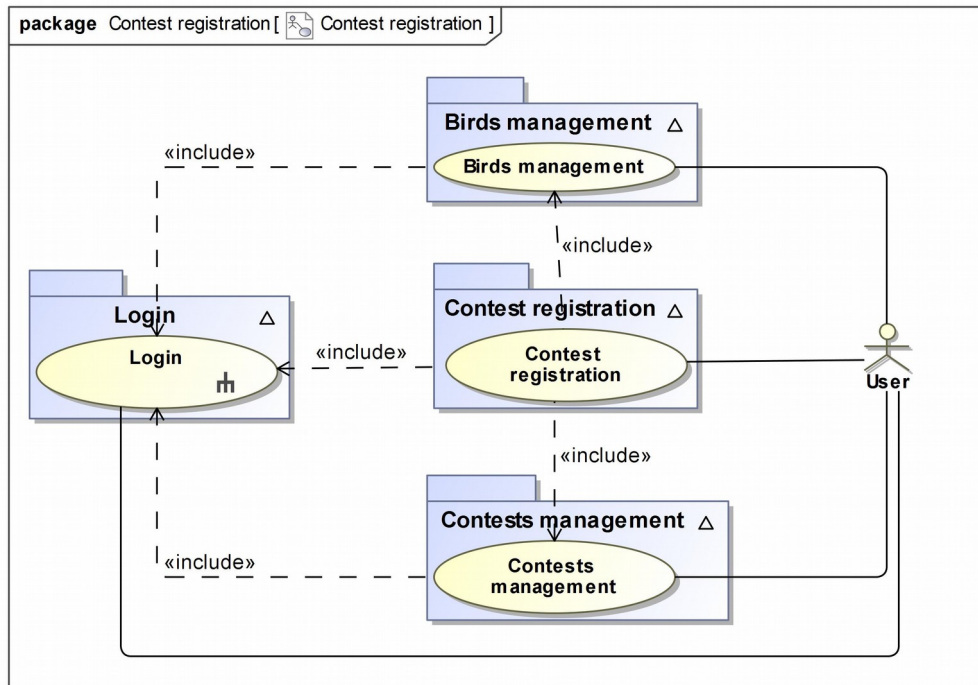
Diagrama de “Contests management”



Caso de Uso: “Contests management”

Nombre	Contests management	ID	UC14
Descripción	This use case manages the contest's information, according to Ornithoculture system description in Contest management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; the organism should be registered, At least a judge must be registered (no judge, no contest).		
Postcondición	Contest has been registered, updated or deleted.		
Restricciones	<ul style="list-style-type: none"> • Judges of the contest should be competent, at least, in one of the contest's sections 		

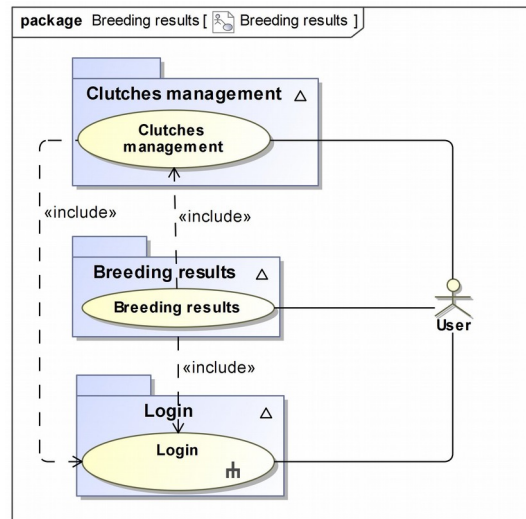
Diagrama de “Contest registration”



Caso de Uso: “Contest registration”

Nombre	Contest registration	ID	UC15
Descripción	This use case manages the contest registration's information, according to Ornithoculture system description in Contest management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; the contest should be registered; The bird should be registered.		
Postcondición	Contest registration has been registered, updated or deleted.		
Restricciones	<ul style="list-style-type: none"> • The evaluator judge must be one judge of the contest 		

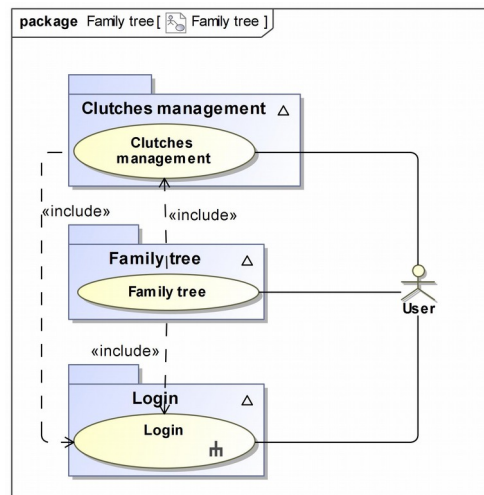
Diagrama de “Breeding results”



Caso de Uso: “Breeding results”

Nombre	Breeding results	ID	UC16
Descripción	This use case manages the breeding results report, according to Ornithoculture system description in Information management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; some clutches should be registered.		
Postcondición	Breeding results report has been shown.		

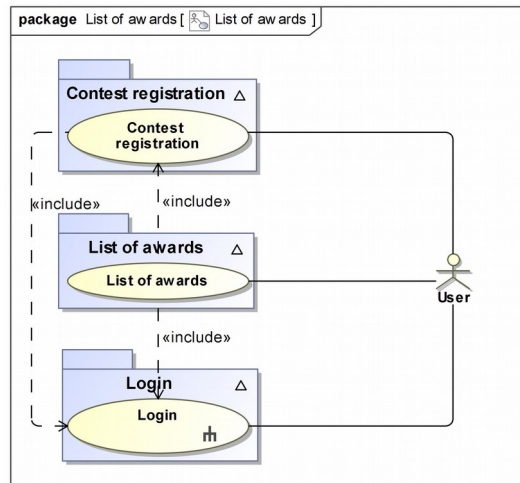
Diagrama de “Family tree”



Caso de Uso: “Family tree”

Nombre	Family tree	ID	UC17
Descripción	This use case manages the family tree report, according to Ornithoculture system description in Information management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; some bird's ancestors should be registered.		
Postcondición	Family tree report has been shown.		

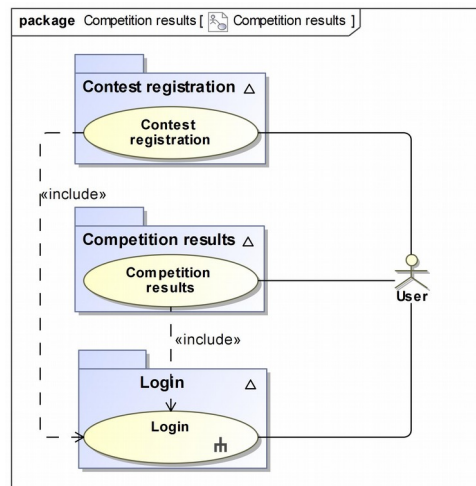
Diagrama de “List of awards”



Caso de Uso: “List of awards”

Nombre	List of awards	ID	UC18
Descripción	This use case manages the list of awards report, according to Ornithoculture system description in Information management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; some contest registrations whit awards should be registered.		
Postcondición	List of awards report has been shown.		

Diagrama de “Competition results”



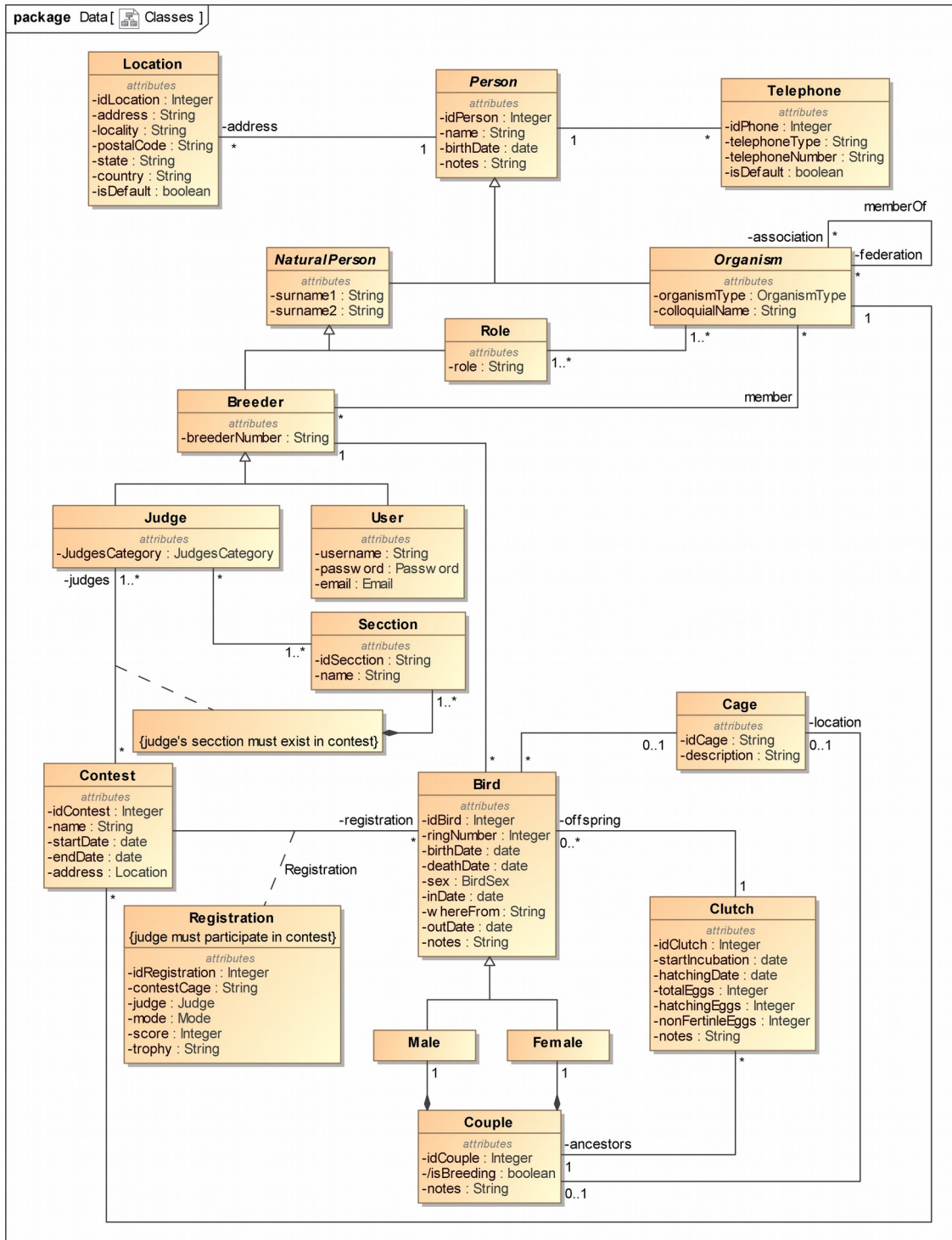
Caso de Uso: “Competition results”

Nombre	Competition results	ID	UC19
Descripción	This use case manages the competition results report, according to Ornithoculture system description in Information management.		
Actores	<ul style="list-style-type: none"> • User • System 		
Precondición	User must be logged; some contest registrations should be registered.		
Postcondición	Competition results report has been shown.		

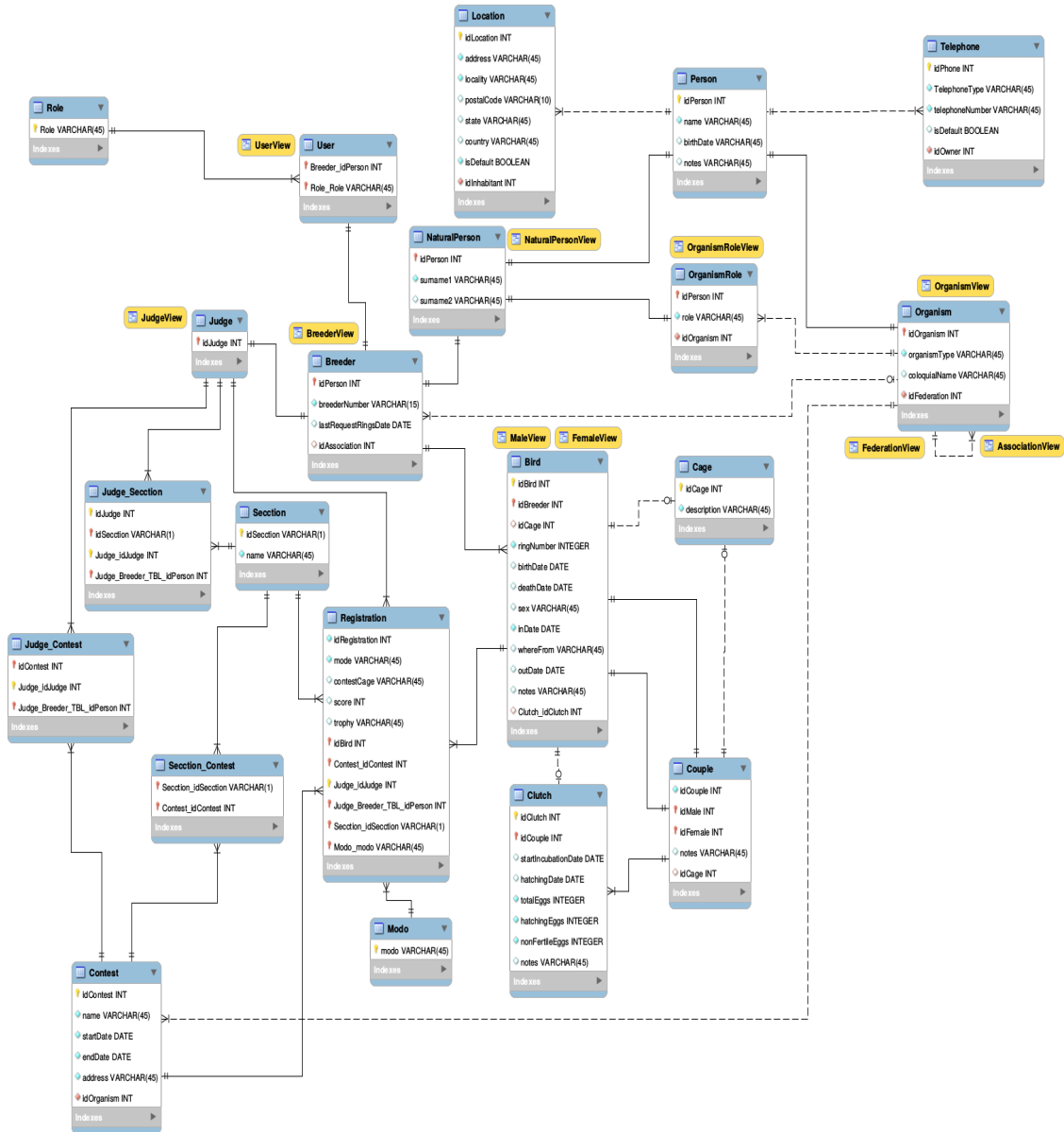
6. Datos

En este capítulo exponemos la especificación de la estructura estática del sistema y la correspondiente estructura de los datos persistentes en el mismo. Para ello, utilizaremos el diagrama de clases y su correspondiente modelo relacional.

6.1 Diagrama de clases



6.2 Diseño relacional de la base de datos

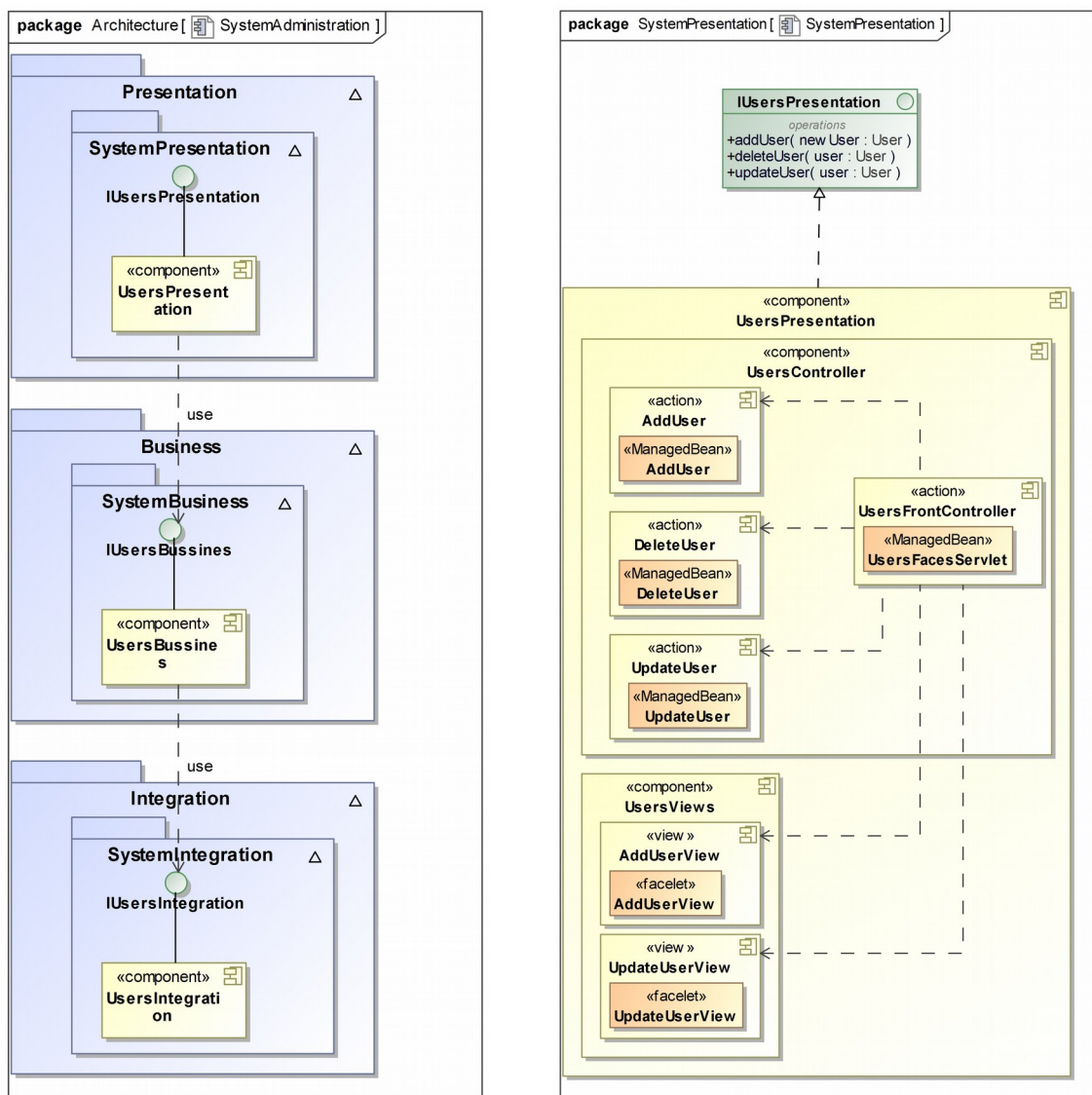


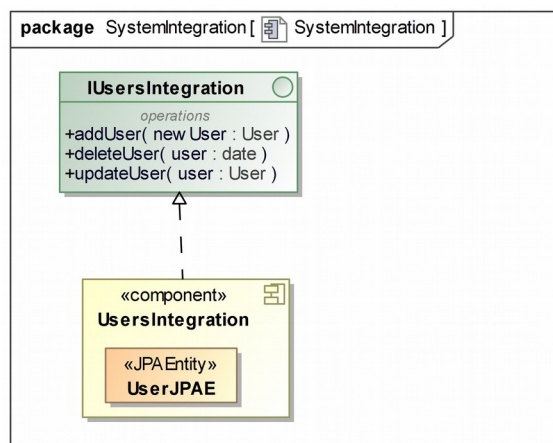
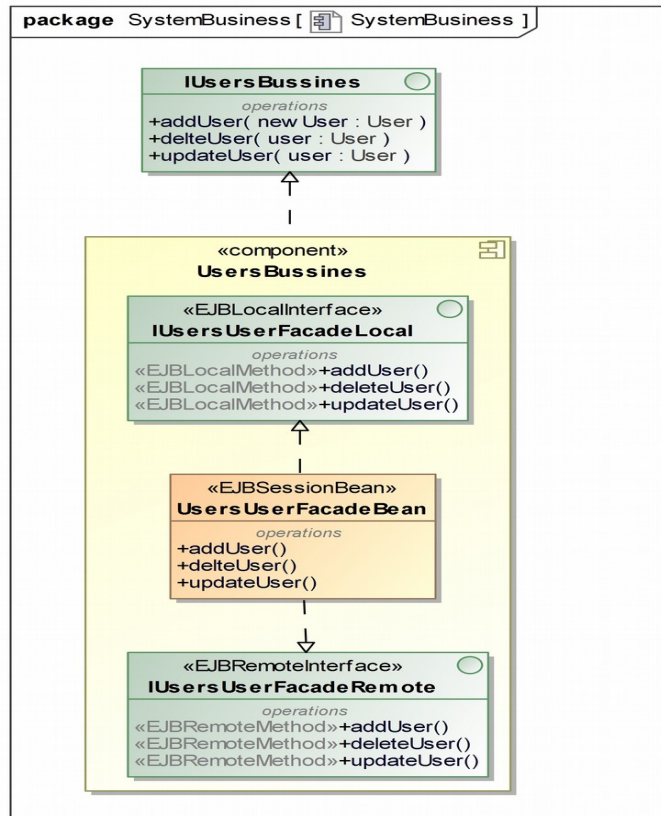
7. Arquitectura

Para la estructuración de nuestro proyecto utilizaremos el patrón MVC (Modelo – Vista – Controlador). Y, como veremos en los diagramas, tendremos un único controlador para todas las operaciones de un determinado componente (patrón FrontController). Así, las operaciones no se implementarán dentro del controlador. Para ello, se utilizará un esquema basado en el patrón Command. Por tanto, el controlador será el responsable de coordinar todo el proceso, pero la implementación estará desacoplada. Por otra parte, en cada acción que el usuario puede realizar por medio de la interfaz de usuario (GUI) se mapeará con una relación 1:1 con una vista y el controlador simplemente ejecuta estas acciones.

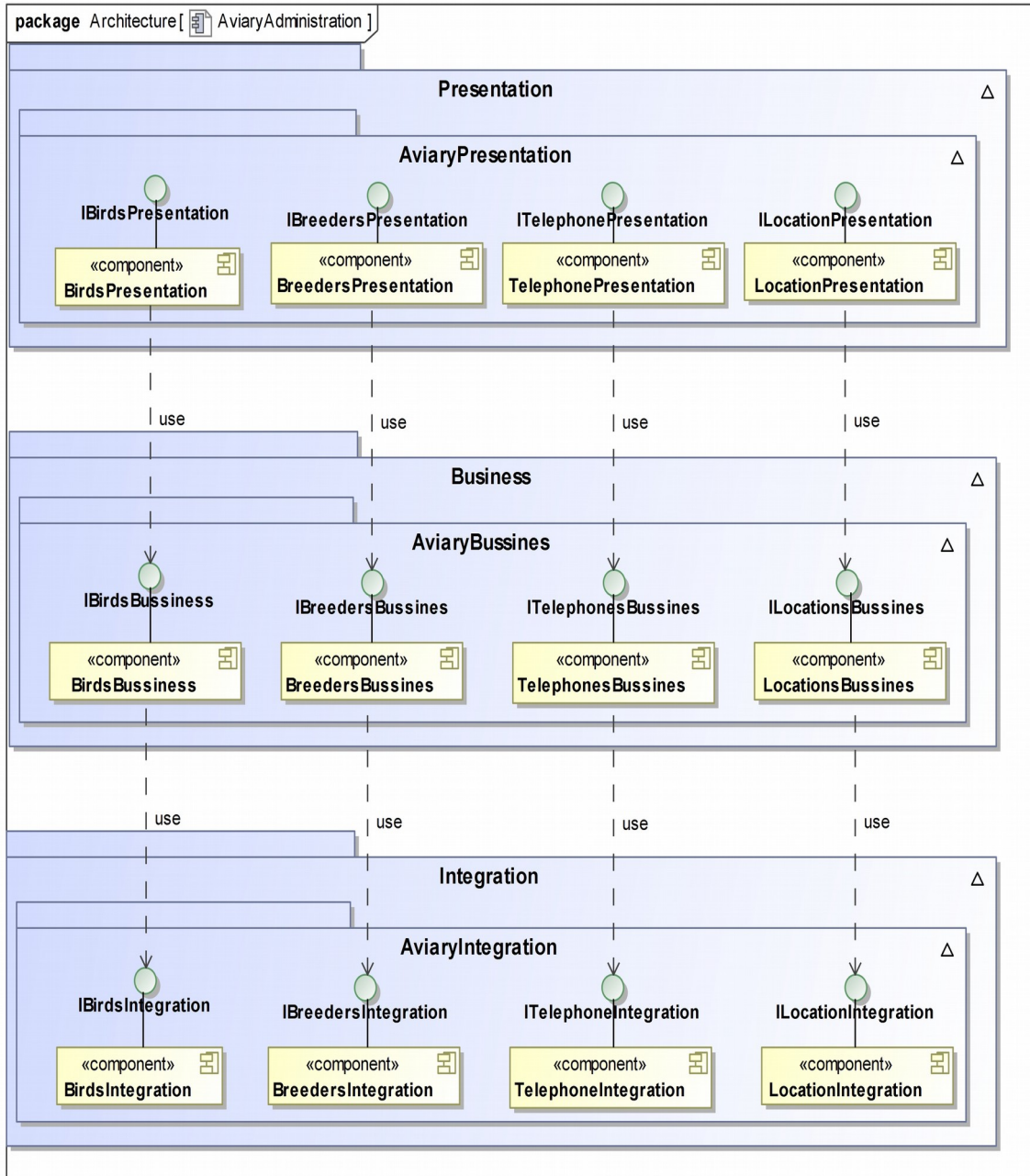
Siguiendo la tónica de estructuración de la información que hemos planteado desde el inicio y adaptándola al patrón MVC, presentaremos los diagramas de componentes agrupando los conocidos bloques: System, Aviary, Breeding y Contest. Primero los diagramas globales de cada uno de estos bloques, seguidos de los correspondientes niveles más detallados (particularmente aquellos casos que no son repetitivos).

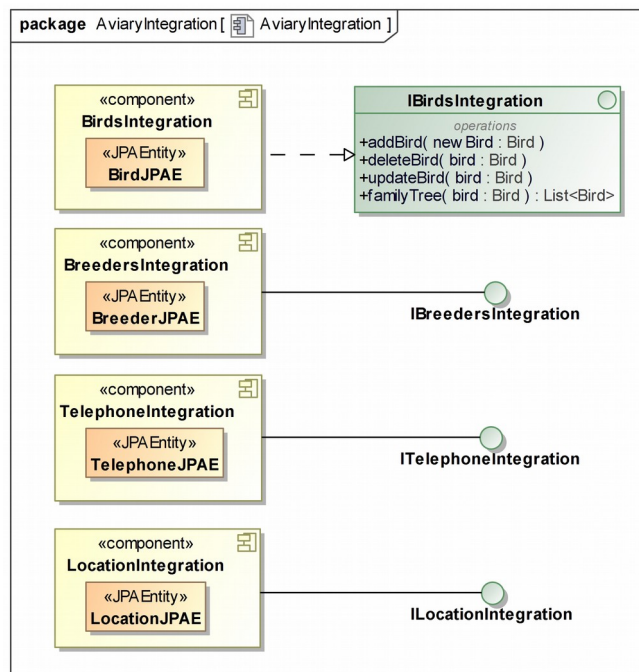
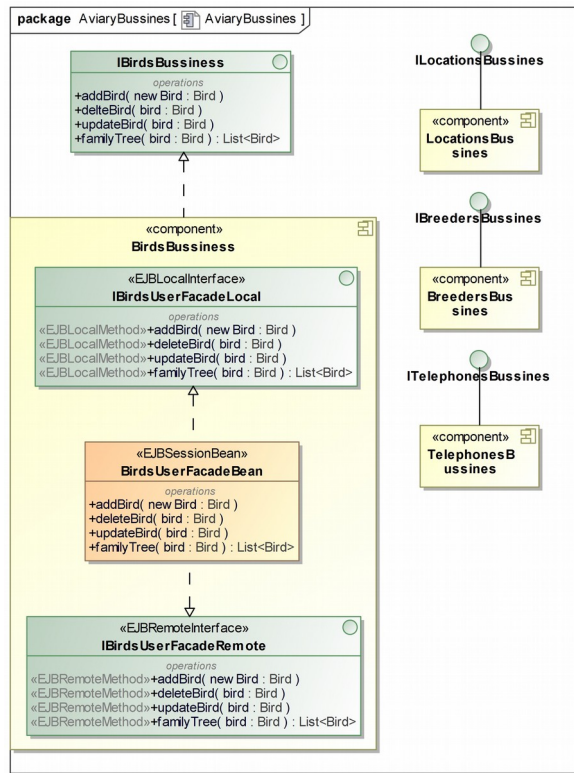
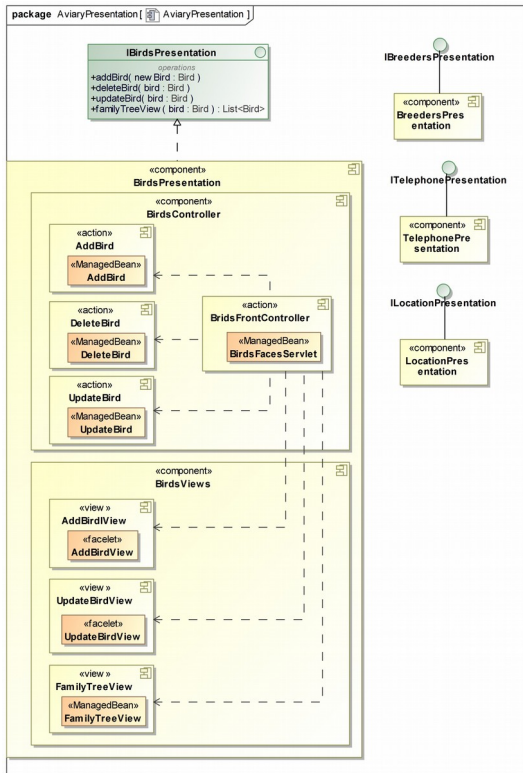
7.1. Diagramas de componentes de “System”



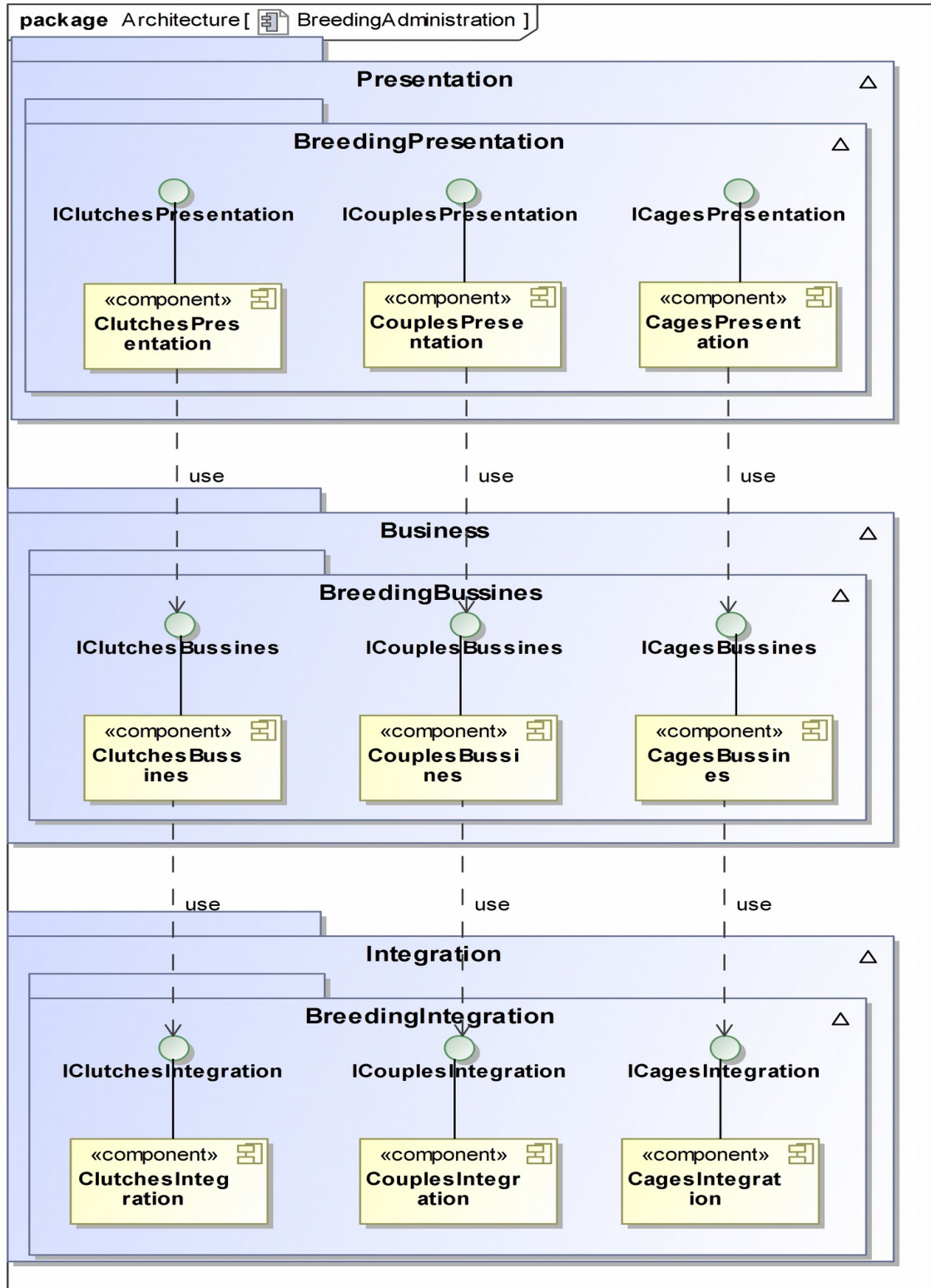


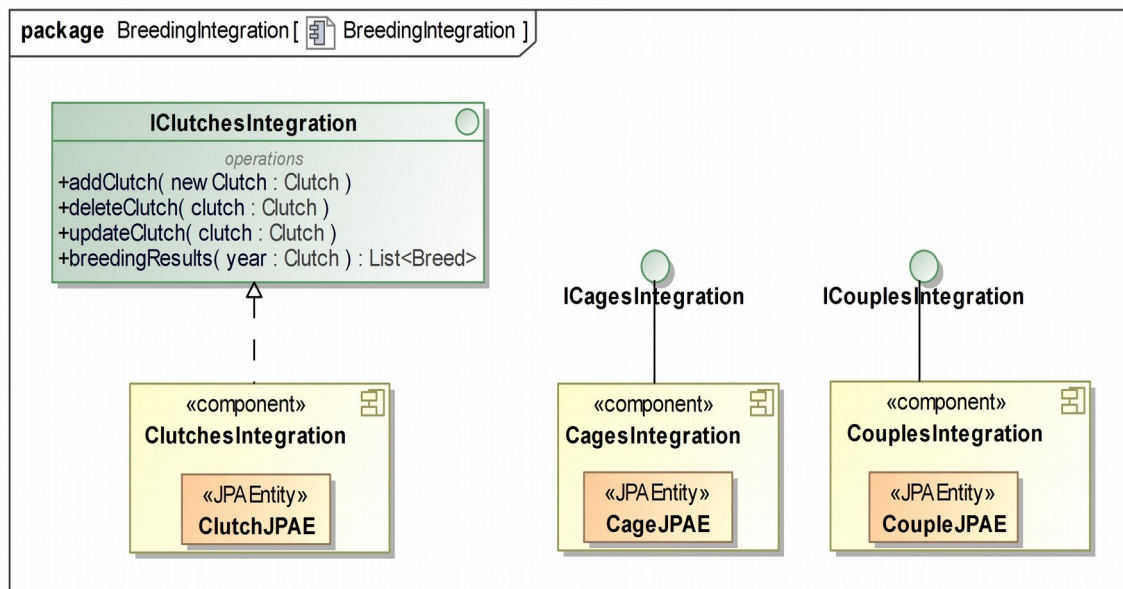
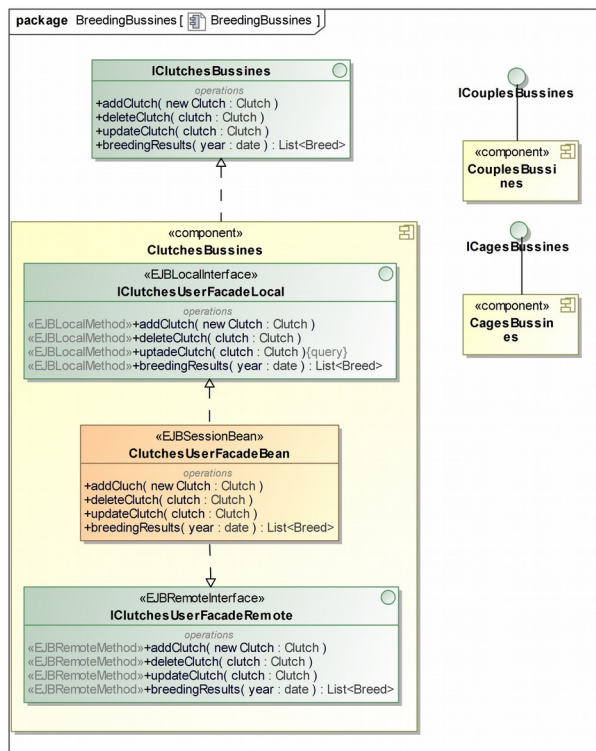
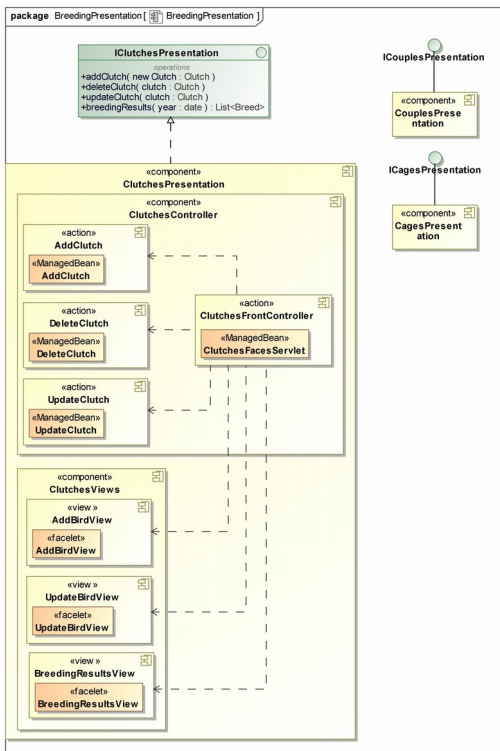
7.2. Diagramas de componentes de “Aviary”



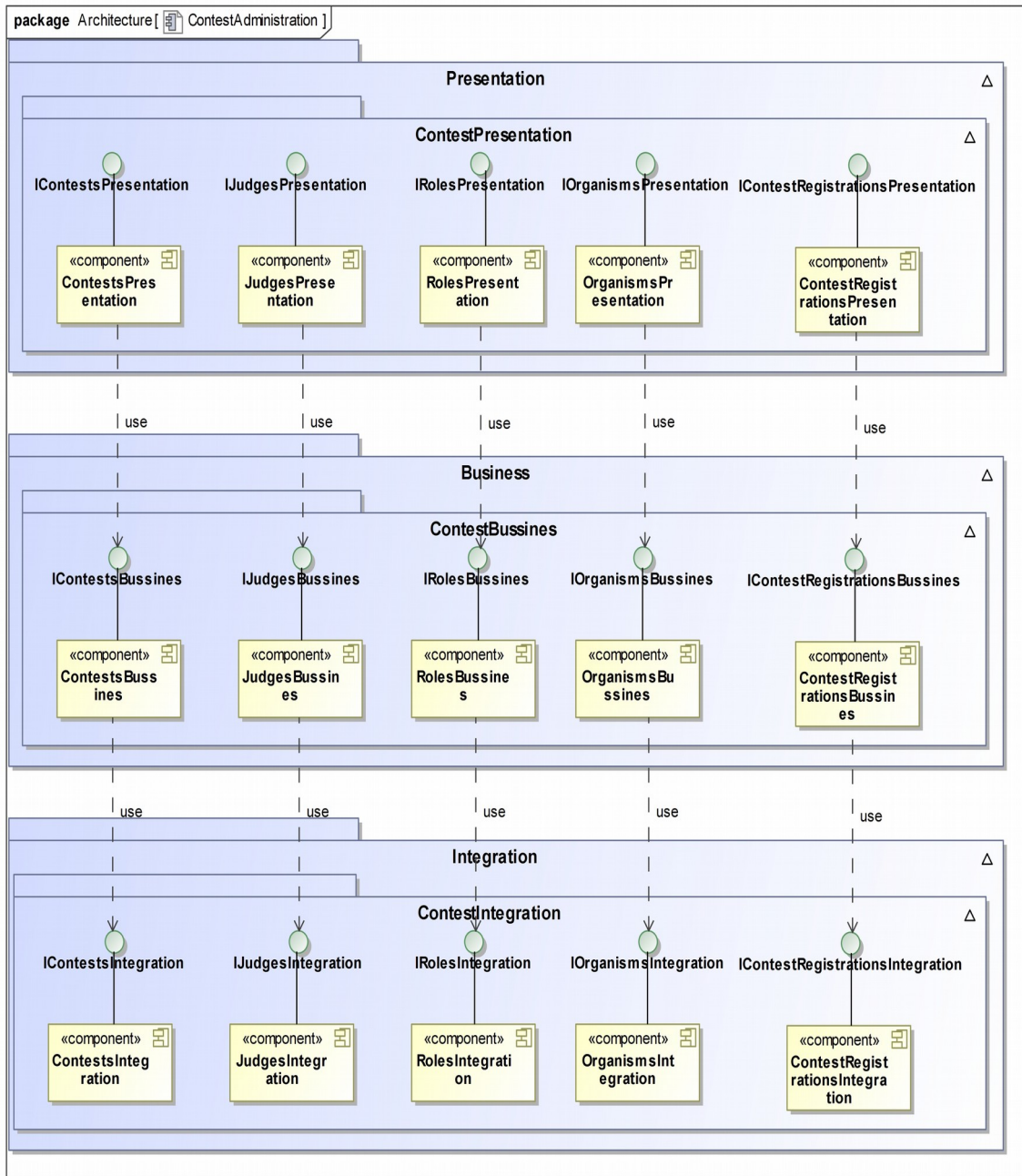


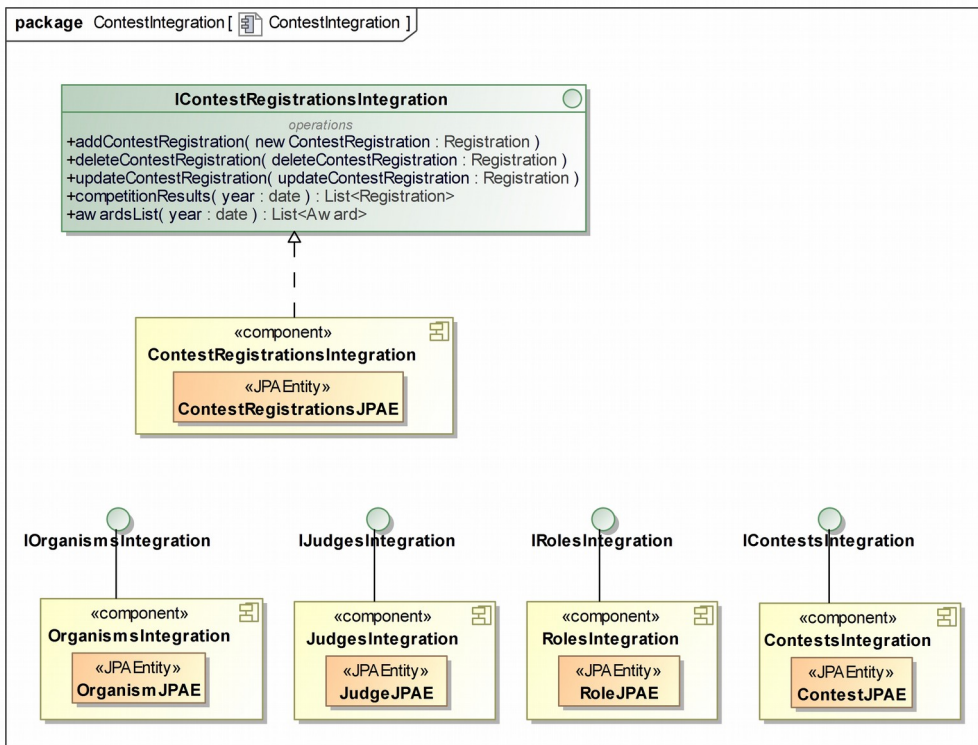
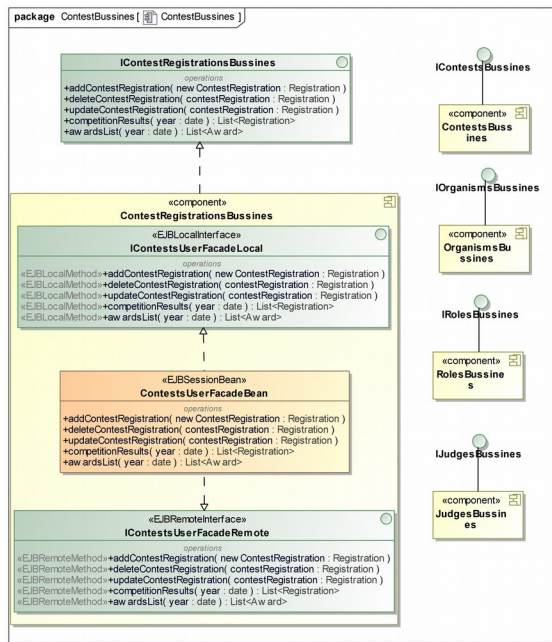
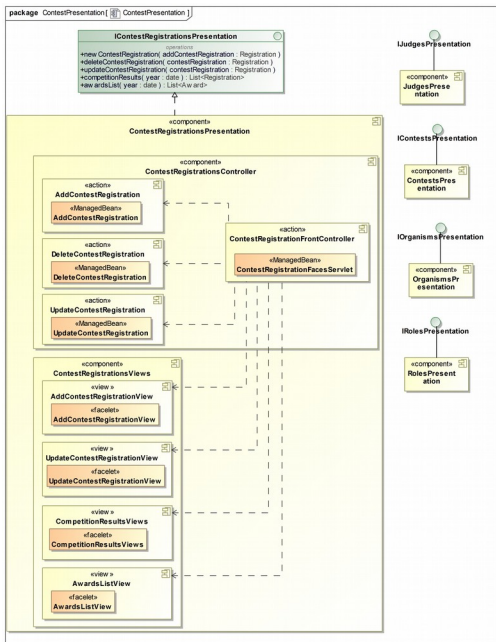
7.3. Diagramas de componentes de “Breeding”





7.4. Diagramas de componentes de “Contest”





8. Bibliografía

- [Ingeniería del software de componentes y sistemas distribuidos](#) (UOC).
- [Ingeniería de requisitos](#) (UOC).
- [Ingeniería del software](#) (UOC).
- [Gestión de proyectos](#) (UOC).
- [Redes y aplicaciones Internet](#) (UOC).
- [Diseño de bases de datos](#) (UOC).
- [The Java EE 6 Tutorial](#). (vídeos)
- [Eclipse](#): – ([Wikipedia](#)).
- Google:
 - [Plugin for Eclipse](#) (GPE).
 - [Web Toolkit](#) (GWT).
 - [App Engine](#) (GAE) ([Wikipedia](#)).
 - [Documentation](#).
- [JSF Tools Reference Guide](#).
- [EJB 3.0 Tutorial](#).
- [JPA Tutorial, JPA vs Hibernate](#).
- [Stack Overflow Business Solutions](#).
- Patrón:
 - [Modelo–Vista–Presentador](#) (MVP).
 - [Modelo–vista–controlador](#).
- [Persistencia de datos](#).
 - [SQLite](#).
 - [MySQL](#).
- [GanttProject](#).
 - [Download](#).
 - [Tutorial](#).
- [Canaricultura](#).
 - Los canarios (como seleccionarlos y cuidarlos), Cliff Newby, Editorial Hispano Europea, ISBN 84-255-0456-2.
 - The Norwich Canary (A guide to keeping, breeding an showing), Joe A. Bracegirdle, Blandford press, ISBN 0-7137-2019-0.
- [LibreOffice](#).
- [Grsync](#).
- [Vokoscreen](#).
- [Kdenlive](#).
- [Cambridge Dictionary](#).