

# **Software de gestión ganadera y análisis genético**

Trabajo de Fin de Grado  
**INGENIERÍA INFORMÁTICA**



## **VNiVERSiDAD D SALAMANCA**

---

CAMPUS DE EXCELENCIA INTERNACIONAL

Julio de 2017

Autor

**Pablo Jesús Espinosa Bermejo.**

Tutores

**Fernando De la Prieta Pintado.**

**Juan Ramos González.**

**Elena Solera Segura.**



## Certificado de los tutores.

D. Fernando De la Prieta Pintado profesor del Departamento de Informática y Automática de la Universidad de Salamanca, D. Juan Ramos miembro del grupo de investigación BISITE de la Universidad de Salamanca y Dña. Elena Solera Segura bióloga en Innovagenomics S.L.

CERTIFICAN:

Que el trabajo titulado “Software de gestión ganadera y análisis genético” ha sido realizado por D. Pablo Jesús Espinosa Bermejo, con DNI 70883640D y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado en Ingeniería Informática de esta Universidad.

Y para que así conste, a todos los efectos oportunos, firman el presente documento.

En Salamanca, a 7 de Julio de 2017.

D.Fernando De la Prieta Pintado  
Dpto. de Informática y Automática  
Universidad de Salamanca

D. Juan Ramos González  
Grupo de Investigación BISITE  
Universidad de Salamanca

Dña. Elena Solera Segura  
Innovagenomics S.L.



## Resumen

El sector primario es uno de los principales sectores de la economía de Castilla y León. Más concretamente, el sector ganadero tiene gran importancia en nuestra comunidad ya sea bien por el ganado porcino o bovino, con denominaciones de origen y razas con gran reconocimiento tanto a nivel nacional como internacional. Además, este sector está cada vez más interesado en la innovación y durante los últimos años colabora e impulsa proyectos de investigación junto a la Universidad de Salamanca.

Para el aumento de la productividad del ganado, este sector tiene una alta demanda de análisis genéticos y por lo tanto existe una gran cantidad de laboratorios dedicados a ello.

Sin embargo, es importante señalar que en un mundo en el que la mayoría de sectores han dado el salto a la automatización de sus tareas de gestión y análisis de datos mediante recursos y aplicaciones informáticas nos encontramos con que el sector ganadero aun no hace uso de ellos, o al menos no de la mejor forma. En consecuencia, la comunicación y la transferencia de datos a los laboratorios se produce de forma rudimentaria. Por lo tanto, el salto hacia este tipo de recursos y aplicaciones se debería de realizar de forma ágil ya que se traduciría en una mejora de la productividad.

Con el desarrollo de este proyecto se ha creado una aplicación que sea capaz de cubrir las necesidades de este sector. Se han utilizado las tecnologías y estándares más actuales. Esto ha involucrado el uso del framework de PHP Laravel y también los lenguajes HTML, CSS y JavaScript. Todo esto siguiendo el patrón Modelo Vista Controlador adaptado a Laravel.

La aplicación está dividida en dos vertientes bien definidas. La gestión de una asociación ganadera y todo lo que ello conlleva y el análisis genético del ganado gracias a la implementación de algoritmos genéticos. Estas dos vertientes se pueden relacionar con las necesidades de una asociación ganadera en el primer caso y con las de un laboratorio de análisis genético en el segundo. Sin embargo, todo esto se encuentra perfectamente ensamblado, de forma que la comunicación y el acceso a datos entre las dos partes se produzca de manera fácil y rápida. Para esto el sistema implementa un control de roles y responsabilidades que cubren todos los posibles actores que interactúan con el sistema.

Para el desarrollo de la aplicación se han utilizado las directrices del Proceso Unificado, y mediante la consecución de varias iteraciones en las que se han ido alcanzando determinados hitos se ha obtenido el producto de software que se describe en esta memoria.

## Palabras clave

Web, Análisis, Algoritmos, Genética, Gestión, Ganado, Laravel.



## Summary

The primary sector is one of the main sectors of the economy of Castilla y León. More specifically, the livestock sector has great importance in our community, whether by pigs or cattle, it has some certificates of origin and breeds with great recognition both nationally and internationally. In addition, this sector is increasingly interested in innovation and during the recent years collaborates and promotes research projects along with the University of Salamanca.

To increase the productivity of livestock, this sector has a high demand for genetic analysis and therefore there is a large number of laboratories dedicated to it.

However, it is important to note that in a world in which most sectors have made the leap to automate their tasks of data management and analysis through technological resources and computer applications we find that the livestock sector still does not use them, or at least not in the best way. Consequently, the communication and transfer of data to the laboratories occurs in a rudimentary way. Therefore, the leap towards this type of resources and applications should be made agile as it would translate into improved productivity.

The development of this project has created an application that is able to meet the needs of this sector. The most up-to-date technologies and standards have been used. This has involved the use of the Laravel PHP framework as well as the HTML, CSS and JavaScript languages. All of this following the Model View Controller pattern adapted to Laravel.

The application is divided into two well-defined slopes. The management of a cattle ranch association and all that it entails and the genetic analysis of livestock thanks to the implementation of genetic algorithms. These two aspects can be related to the needs of a cattle ranch association in the first case and those of a genetic analysis laboratory in the second. However, all this is perfectly assembled, so that communication and access to data between the two parts take place easily and quickly. For this purpose, the system implements a role and responsibility control that cover all the possible actors that may interact with the system.

For the development of the application, the guidelines of the Unified Process have been used, and through the achievement of several iterations in which certain milestones have been reached, the software product described in this report has been obtained.

## Keywords

Web, Analysis, Algoritmos, Genetics, Management, Livestock, Laravel.



## Tabla de contenido

1.	Introducción .....	13
2.	Objetivos.....	17
2.1.	Objetivos funcionales.....	17
2.2.	Objetivos no funcionales.....	17
2.3.	Objetivos personales.....	18
3.	Estado del arte.....	19
3.1.	Conceptos teóricos.....	19
3.1.1.	<i>ADN</i> .....	19
3.1.2.	<i>Gen</i> .....	20
3.1.3.	<i>Filiación Genética</i> .....	21
3.2.	Herramientas.....	22
3.2.1.	<i>Software de Gestión Ganadera</i> .....	22
3.2.2.	<i>Análisis Genético</i> .....	26
3.2.3.	<i>Intercambio de datos</i> .....	27
4.	Técnicas y Herramientas .....	29
4.1.	HTML, CSS y JavaScript.....	29
4.1.1.	<i>HTML</i> .....	29
4.1.2.	<i>CSS</i> .....	30
4.1.3.	<i>JavaScript</i> .....	31
4.1.4.	<i>Twitter Bootstrap</i> .....	32
4.1.5.	<i>jQuery</i> .....	32
4.2.	PHP .....	33
4.3.	Patrón MVC.....	33
4.4.	Laravel Framework.....	34
4.4.1.	<i>Estructura de los directorios en la aplicación</i> .....	35
4.5.	MySQL .....	36
4.6.	phpMyAdmin.....	36
4.7.	Apache.....	37
4.8.	PhpStorm.....	38
4.9.	DreamWeaver .....	38
4.10.	Sketch .....	39
4.11.	Microsoft Excel.....	39
4.12.	Herramientas CASE .....	40
4.12.1.	<i>UML (Lenguaje Unificado de Modelado)</i> .....	40
4.12.2.	<i>REM</i> .....	41
4.12.3.	<i>Microsoft Project</i> .....	41
4.12.4.	<i>EZEstimate</i> .....	41
4.12.5.	<i>phpDocumentor</i> .....	42
4.12.6.	<i>Git (VCS)</i> .....	42
5.	Aspectos relevantes del desarrollo .....	45
5.1.	Metodología .....	45
5.1.1.	<i>Primera Iteración</i> .....	46
5.1.2.	<i>Segunda Iteración</i> .....	48

## Memoria

5.1.3.	<i>Tercera Iteración</i> .....	53
5.1.4.	<i>Cuarta Iteración</i> .....	58
5.1.5.	<i>Quinta Iteración</i> .....	59
5.2.	MVC en Laravel .....	60
5.2.1.	<i>Rutas</i> .....	60
5.2.2.	<i>Modelo</i> .....	60
5.2.3.	<i>Vista</i> .....	62
5.2.4.	<i>Controlador</i> .....	64
5.3.	Implementación del algoritmo de análisis genético.....	65
6.	Conclusión .....	71
7.	Líneas de trabajo futuro .....	73
8.	Referencias .....	75

## Listado de figuras.

Ilustración 1	El ADN y su estructura [9].....	19
Ilustración 2	Par de cromosomas y sus alelos .....	21
Ilustración 3	Aplicación utilizada por la Asociación de la Morucha. ....	22
Ilustración 4	Formulario de registro de un animal en la plataforma y base de datos FILOVI proporcionada por el Ministerio de Agricultura y Pesca, Alimentación y Medio Ambiente. 23	
Ilustración 5	Aplicación Bovi Control.....	24
Ilustración 6	Aplicación Vaquitec. ....	25
Ilustración 7	Logo de HTML, CSS y JS .....	29
Ilustración 8	Logo de PHP.....	33
Ilustración 9	MVC adaptado a Laravel.....	34
Ilustración 10	Logo de Laravel.....	34
Ilustración 11	Logo de MySQL.....	36
Ilustración 12	Logo de phpMyAdmin .....	36
Ilustración 13	Logo Apache Server .....	37
Ilustración 14	Logo de PhpStorm .....	38
Ilustración 15	Logo de DreamWeaver .....	38
Ilustración 16	Logo de Sketch.....	39
Ilustración 17	Logo de Microsoft Excel.....	39
Ilustración 18	Logo de UML.....	40
Ilustración 19	Logo de Visual Paradigm .....	40
Ilustración 20	Logo de Microsoft Project 2016 .....	41
Ilustración 21	Logo de phpDocumentor.....	42
Ilustración 22	Logo git .....	42
Ilustración 23	Logo de GitHub.....	43
Ilustración 24	Elicitación de requisitos.....	46
Ilustración 25	Especificación de Requisitos con la herramienta REM .....	47
Ilustración 26	Captura de la analítica de Google Trends.....	48
Ilustración 27	Modelo de Dominio inicial.....	49
Ilustración 28	Vista de Arquitectura del Modelo de Análisis .....	50
Ilustración 29	Diagrama de Secuencia del caso de uso “Iniciar Sesión” .....	51
Ilustración 30	Inicio del diseño de la interfaz de la web con DreamWeaver .....	52
Ilustración 31	Actualización del Modelo de Dominio.....	53
Ilustración 32	Actualización de la vista de Arquitectura del Sistema.....	54
Ilustración 33	Nueva realización de Casos de Uso añadidos al sistema.....	54
Ilustración 34	Capa controlador .....	55
Ilustración 35	Vista arquitectónica del diseño .....	56
Ilustración 36	Diagrama de Secuencia. UC-0015. Baja Ganado .....	57
Ilustración 37	Diseño de los iconos de la aplicación con Sketch. ....	58
Ilustración 38	Capa de Rutas de la aplicación .....	60
Ilustración 39	Capa Modelo del sistema desarrollado .....	61

## Memoria

Ilustración 40	Funcionamiento de Eloquent con almacenamiento persistente.....	62
Ilustración 41	Estructura de las vistas en Laravel Blade.....	63
Ilustración 42	Contenido de la Capa Vista.....	64
Ilustración 43	Ejemplo de una filiación realizada por Innovagenomics .....	65
Ilustración 44	Tabla de fórmulas utilizadas para la filiación parental .....	66
Ilustración 45	Primeros resultados efectivos de la aplicación del algoritmo.....	67
Ilustración 46	Interfaz para la solicitud de una filiación.....	68
Ilustración 47	Resultados de una filiación sin parente conocido, resultados según los presenta Innovagenomics y resultados de una filiación con parente conocido.....	69

## 1. Introducción.

El presente documento recoge la memoria del Trabajo de Fin de Grado titulado “Software de gestión ganadera para la interpretación, almacenamiento, análisis y gestión de datos identificativos, genéticos, productivos, sanitarios y económicos” en la titulación de Grado en Ingeniería Informática. Ha sido realizado por el alumno Pablo Jesús Espinosa Bermejo durante los meses de febrero a julio de 2017.

El proyecto ha sido desarrollado en colaboración con la empresa Innovagenomics S.L. (Sector Biológico/Genético) y ha sido dirigido por D. Fernando de la Prieta Pintado, D. Juan Ramos y D.ª Elena Solera Segura.

El sector primario es uno de los principales sectores de nuestra región [20]. Más concretamente el sector ganadero tiene gran importancia en ésta ya sea bien por el ganado porcino o bovino, con denominaciones de origen y razas con gran reconocimiento tanto a nivel nacional como internacional [33]. Además, este sector está cada vez más interesado en la innovación y durante los últimos años colabora e impulsa proyectos de investigación junto a la Universidad de Salamanca [5].

Sin embargo, es importante señalar que en un mundo en el que la mayoría de sectores han dado el salto a la automatización de sus tareas de gestión y análisis de datos mediante recursos y aplicaciones informáticas nos encontramos con que el sector ganadero aun no hace uso de ellos, o al menos no de la mejor forma. El salto hacia este tipo de sistemas se debería de realizar de forma ágil ya que se podría traducir en una mejora de la productividad.

El objetivo principal del proyecto es el desarrollo de una herramienta que permita la gestión de una asociación ganadera, pero gracias a la cantidad de datos almacenados de tipo genético, también ofrece la posibilidad de utilizar algoritmos de filiación parental genética. Para esto nos pusimos en contacto con la asociación de la Morucha además de con Innovagenomics y vimos como trabajaban actualmente. Como pudimos comprobar manejan una gran cantidad de datos, sobre todo de ganado, pero también de las ganaderías, las explotaciones, las muestras biológicas de los animales para intercambiarlas con los laboratorios...

Este trabajo se realiza actualmente de la siguiente forma: por parte de la asociación consiste básicamente en un software poco flexible y cuyas bases de datos se alojan solamente de forma local. Por lo tanto, el acceso de forma remota y el intercambio de datos se complican. Por otro lado, las empresas o laboratorios de análisis genéticos, en este caso concreto Innovagenomics, utilizan hojas de cálculo para tratar todo el conjunto de datos genéticos de los animales, que además es de gran tamaño tanto en número de animales como en características a almacenar sobre ellos y sus datos genéticos. Además, la trasferencia de datos entre la asociación y la empresa que realiza los análisis se realiza mediante e-mails con hojas de cálculo adjuntas.

## Memoria

Para resumir, el sistema que se usa actualmente está lejos de ser el idóneo para tratar una gran cantidad de datos y además la comunicación entre ambas partes es bastante rudimentaria.

Por lo tanto, se decide desarrollar una aplicación web toda la información y datos estén almacenados del lado del servidor en vez de forma local. Esto se traduce en una forma mucho más ágil y accesible además de mejorar la comunicación y trasferencia de datos entre ambas partes. Además, gracias a las reuniones que han tenido lugar con los implicados en el proyecto durante el desarrollo, se ha añadido funcionalidad que se ha considerado útil tanto por parte de las asociaciones ganaderas como por Innovagenomics, algunos ejemplos sobre esto son la ayuda a la migración de los datos desde y hacia los sistemas utilizados actualmente o funciones y algoritmos para realizar filiaciones de una forma mucho más fácil e intuitiva que anteriormente. También es importante señalar que, aunque el desarrollo se ha hecho de forma conjunta con Innovagenomics, se pone énfasis en que la aplicación sea lo más adaptable posible a otras empresas con mismas o parecidas necesidades.

Esta memoria pretende describir el proyecto realizado, está compuesta por las siguientes secciones:

- Introducción: Sección en la que se está presente, y que trata de poner al lector en situación.
- Objetivos: Sección donde se describen los objetivos hacia los que apunta el proyecto.
- Estado del arte: Describe los conceptos teóricos necesarios para la comprensión del proyecto y las herramientas actualmente utilizadas para la realización de trabajos similares.
- Técnicas y herramientas: En esta sección se describe el software utilizado para la realización del proyecto.
- Aspectos relevantes del desarrollo: Sección en la que se comenta los puntos mas importantes del desarrollo de este proyecto.
- Conclusiones: Breve exposición de lo aprendido en el proyecto y los objetivos que se han alcanzado.
- Líneas de trabajo futuro: Describe las posibles ampliaciones del proyecto en el futuro.
- Referencias: Citas utilizadas para el desarrollo de esta memoria.

Además de la memoria, el documento presente, se entrega la documentación técnica que se compone de 5 anexos y el código fuente:

- Anexo I: Plan de proyecto software: En este anexo se realiza la planificación temporal del proyecto y un análisis del esfuerzo.
- Anexo II: Especificación de requisitos: se realiza el análisis de requisitos funcionales y no funcionales, los modelos de casos de uso y la especificación detallada de casos de uso.

- Anexo III: Especificación de análisis: Recoge el análisis del sistema, su división en paquetes de análisis y la realización de casos de uso de la etapa de análisis.
- Anexo IV: Especificación de diseño: Incluye los documentos de diseño de la aplicación, así como la arquitectura empleada para la construcción de ésta. Se recoge la realización de casos de uso de la etapa de diseño.
- Anexo V: Documentación técnica y despliegue: Contiene las instrucciones para la instalación del software en el servidor y la documentación necesaria para el programador.
- Anexo VI: Manual de usuario: Recoge las directrices de uso de la herramienta para el usuario final.



## 2. Objetivos.

El objetivo principal del trabajo es la realización de una aplicación web que permita el almacenamiento, gestión y consulta de datos relacionados con una asociación ganadera, sus ganaderías, explotaciones, ganado y además la posibilidad por parte del laboratorio de realizar filiaciones parentales gracias a estos datos. A continuación, se van a exponer los objetivos funcionales y no funcionales de una forma más concreta.

### 2.1. Objetivos funcionales.

La aplicación deberá permitir gestionar una asociación ganadera y todas las secciones de esta, por lo tanto se deberá poder consultar, tratar y modificar información de la propia asociación, sus ganaderías y el ganado y explotaciones de éstas.

Deberá proporcionar un **sistema de gestión y control de usuarios**, posibilitando un registro libre de los mismos para después gestionar los datos y acceder a la información almacenada. También se deberá poder asignar **roles** y diferentes **permisos** según el tipo de usuario que se registra por parte del administrador.

Se deben proporcionar **herramientas para importar y exportar datos** desde y hacia otras aplicaciones o servicios para conseguir una convergencia hacia nuestro sistema. Para esto se debe de permitir la interacción mediante “drag & drop” para facilitar su manejo, ya que además este método no es compatible con todos los navegadores se debe de ofrecer una alternativa (mediante el uso de un formulario) que lo supla.

Para facilitar el manejo y tratamiento de la información se deben de proporcionar **herramientas y opciones de filtrado y búsqueda**.

Se deberá de implementar un **sistema de filiación genética** de forma que aproveche la gran cantidad de datos almacenados para obtener mejores y sobre todo más rápidos resultados que de forma tradicional.

### 2.2. Objetivos no funcionales.

La aplicación deberá permitir el almacenamiento y **acceso ubicuo a información**, de modo que se pueda consultar, tratar y modificar dicha información desde el navegador de cualquier dispositivo sin necesidad de instalar ningún tipo de software.

La aplicación debe de ser utilizable con el mayor número de configuraciones posibles y que además sea **compatible con los principales navegadores y sistemas operativos**. Para ello, utilizamos componentes multiplataforma y evitamos la presencia de dependencias de alguna plataforma en concreto.

La aplicación debe de tener una **interfaz sencilla y orientada al usuario final**. Debe de garantizar que la transición de un software a otro no sea complicada para el usuario.

Utilización de **tecnologías lo más actuales, eficientes, seguras y respaldadas** (HTML5, CSS3, PHP y JavaScript) en la medida de lo posible, evitando tecnologías alternativas que se encuentran desfasadas y problemáticas a todos los niveles (como por ejemplo Adobe Flash), para que de esta forma el **desarrollo, instalación y mantenimiento** de la aplicación sea más **fácil**.

Se intenta que la aplicación sea **modular y adaptable** de forma que se pueda amoldar a las diferentes necesidades de cada asociación o cliente.

Implementación de los servicios usando una **arquitectura RESTful** que permita en un futuro utilizar o mudar estos servicios desde otro tipo de software ya sea web, móvil o escritorio.

Utilización de **tecnologías asíncronas** como por ejemplo AJAX (Asynchronous JavaScript and XML) para la realización de peticiones al servidor para de ésta forma facilitar la usabilidad de la web al usuario evitando tiempos de espera y congelaciones durante las operaciones.

La aplicación debe de salvaguardar la **seguridad y confidencialidad** tanto de los datos como de la propia funcionalidad ante elementos externos al sistema.

## 2.3. Objetivos personales.

Los objetivos personales que se persiguen mediante la realización del presente trabajo son:

Entrar de lleno en el conocimiento del **diseño y desarrollo de aplicaciones Web**, un conocimiento que es imprescindible y altamente valorado en la actualidad y que sin embargo se echa en falta en el plan de estudios realizado.

Afianzar y aplicar los conocimientos adquiridos durante los años de carrera en la realización de este trabajo, desde la captura de los requisitos, pasando por el modelo de diseño y terminando con su implementación y despliegue.

Aprendizaje de nuevos lenguajes y tecnologías, como son **PHP** integrado en el **framework Laravel** (Desarrollo Orientado a Tests, Migraciones, uso de ORM Eloquent, sistema de ruteo...) y **JavaScript/JQuery**.

**Aprendizaje transversal**, el aprendizaje del funcionamiento tanto de las empresas del sector como del uso de los algoritmos de genética que se utilizan para el desarrollo del trabajo presente.

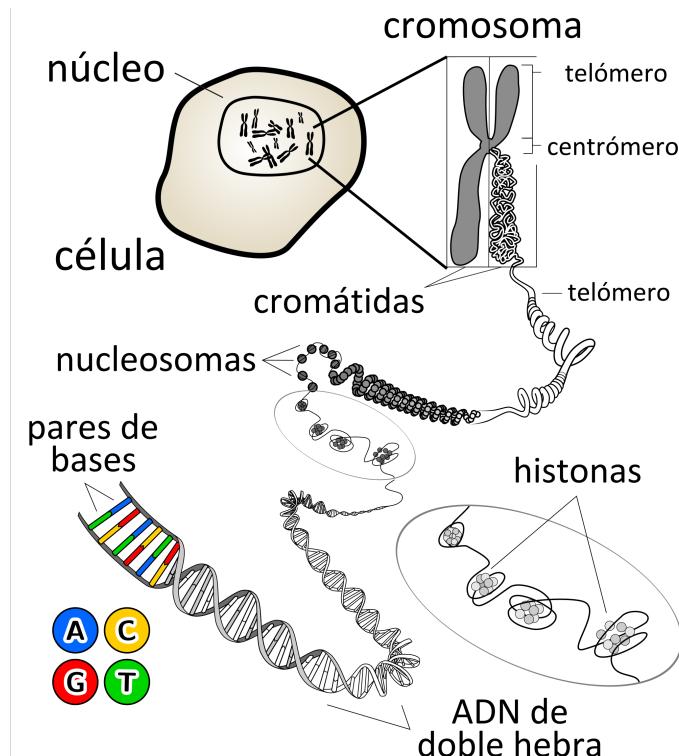
### 3. Estado del arte

#### 3.1. Conceptos teóricos

##### 3.1.1. ADN

El ácido desoxirribonucleico o ADN es junto con el ARN o ácido ribonucleico la principal biomolécula que provee a las células de la información necesaria para su desarrollo y funcionamiento, es decir dependiendo de las características biológicas de la célula, la información presente en el ADN permite el desarrollo de la misma hasta la formación de un organismo adulto, que tendrá ciertas características específicas dada por esta biomolécula, permitiendo asimismo el funcionamiento celular y del organismo y la transferencia de parte de esas características a un nuevo individuo. El ADN es una molécula de gran tamaño y está formada por unidades denominadas nucleótidos, unidos entre sí por enlaces fosfodiéster, formando largas cadenas. Los ácidos nucleicos pueden ser de distintos tipos, cumpliendo a su vez diferentes funciones dentro de la célula [4].

###### 3.1.1.1. Estructura del ADN



**Ilustración 1 El ADN y su estructura [9]**

Cada molécula de ADN está constituida por dos cadenas o bandas formadas por un elevado número de compuestos químicos llamados

nucleótidos. Estas cadenas forman una doble hélice. Cada nucleótido está formado por tres unidades: una molécula de azúcar llamada desoxirribosa, un grupo fosfato y uno de cuatro posibles compuestos nitrogenados llamados bases: adenina (abreviada como A), guanina (G), timina (T) y citosina (C).

La molécula de desoxirribosa ocupa el centro del nucleótido y está flanqueada por un grupo fosfato a un lado y una base al otro. El grupo fosfato está a su vez unido a la desoxirribosa del nucleótido adyacente de la cadena. Estas subunidades enlazadas desoxirribosa-fosfato forman los lados de la escalera; las bases están enfrentadas por parejas, mirando hacia el interior, y forman los travesaños.

Los nucleótidos de cada una de las dos cadenas que forman el ADN establecen una asociación específica con los correspondientes de la otra cadena. Debido a la afinidad química entre las bases, los nucleótidos que contienen adenina se acoplan siempre con los que contienen timina, y los que contienen citosina con los que contienen guanina. Las bases complementarias se unen entre sí por enlaces químicos débiles llamados enlaces de hidrógeno.

En 1953, el bioquímico estadounidense James Watson y el biofísico británico Francis Crick publicaron la primera descripción de la estructura del ADN. Su modelo adquirió tal importancia para comprender la síntesis proteica, la replicación del ADN y las mutaciones, que los científicos obtuvieron en 1962 el Premio Nobel de Medicina por su trabajo.

### 3.1.2. Gen

Un gen es una unidad de información en un locus (posición fija en un cromosoma donde se encuentra un gen o un marcador) de ácido desoxirribonucleico (ADN) que codifica un producto funcional, proteínas por ejemplo. Es la unidad molecular de la herencia. También se conoce como una secuencia de nucleótidos contiguos en la molécula de ADN (o de ARN en el caso de algunos virus) que contiene la información necesaria para la síntesis de una macromolécula con función celular específica; habitualmente proteínas pero también ARN mensajero (ARNm), Ácido ribonucleico ribosómico (ARNr) y ARN de transferencia (ARNr), entre otros.

Ésto último está vinculado con el desarrollo de funciones fisiológicas. El gen es considerado la unidad de almacenamiento de información genética y unidad de la herencia genética, pues transmite esa información a la descendencia. Los genes se disponen, pues, a lo largo de ambas cromátidas de los cromosomas y ocupan, en el cromosoma, una posición determinada llamada locus. El conjunto de genes de una especie se denomina genoma. Los genes están localizados en los cromosomas en el núcleo celular [1].

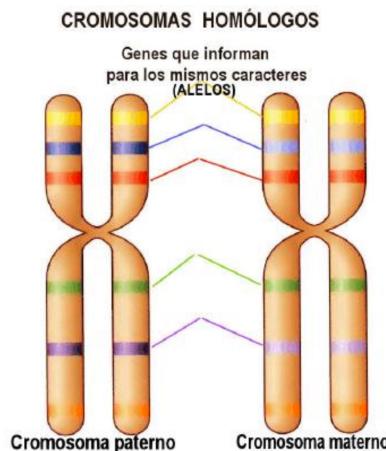
### 3.1.3. Filiación Genética

Cuando hablamos de filiación genética nos referimos a la prueba de paternidad o de maternidad basada en el ácido desoxirribonucleico. Es la técnica médica, biológica y científica que permite establecer la identidad genética (huella genética única que permite conocer la verdad biológica) y la relación filial legítima respecto de quien engendró o procreó.

#### 3.1.3.1. Marcador

Un marcador genético es un segmento de ADN cuya herencia se puede rastrear. Puede ser un gen o un segmento sin función conocida. Dado que las secuencias de ADN que se encuentran contiguas en un cromosoma tienden a heredarse juntas, los marcadores se usan como herramientas para rastrear el patrón hereditario de genes que aún no han sido identificados, pero cuyas ubicaciones aproximadas se conocen [15].

#### 3.1.3.2. Alelo



**Ilustración 2 Par de cromosomas y sus alelos**

Un alelo es cada una de las dos o más versiones de un gen. Un individuo hereda dos alelos para cada gen, uno del padre y el otro de la madre. Los alelos se encuentran en la misma posición dentro de los cromosomas homólogos. Si los dos alelos son idénticos, el individuo es homocigoto para este gen. En cambio, si los alelos son diferentes, el individuo es heterocigoto para este gen. Aunque el término alelo fue usado originariamente para describir variaciones entre los genes, ahora también se refiere a las variaciones en secuencias de ADN no codificante (es decir, que no se expresan) [12].

Los alelos están fuertemente relacionados con la herencia genética. Por ejemplo en la herencia mendeliana, en el caso de un humano se tiene un gen concreto que define el color de los ojos, pueden existir varios alelos para este gen pero para simplificarlo se puede hablar de “ojos claros” y “ojos oscuros”. El alelo “ojos oscuros” es dominante, es decir si se presenta en alguno de los dos alelos

que hereda el hijo, éste se va a manifestar. El alelo “ojos claros” por el contrario es recesivo y solo se manifestará si ambos alelos lo son. Por lo tanto se nos presentan las siguientes combinaciones (padre y madre) y sus productos (hijo):

Madre	Padre	Hijo
Ojos Claros	Ojos Oscuros	Ojos Oscuros
Ojos Oscuros	Ojos Oscuros	Ojos Oscuros
Ojos Claros	Ojos Claros	Ojos Claros

## 3.2. Herramientas

Actualmente existen pocas herramientas para realizar el cometido que se propone. Suele ser software bastante antiguo, difícilmente accesible a las empresas y poco flexible ya que normalmente son aplicaciones para Windows, esto impide por ejemplo su uso en tablets o móviles. Además, no consigue una conectar los laboratorios con las asociaciones ganaderas.

### 3.2.1. Software de Gestión Ganadera

Se trata de una herramienta diseñada específicamente para la Asociación de la Morucha y basada en sus necesidades. Este software se pudo ver y utilizar durante una de las reuniones. Consiste en una aplicación solo compatible con Windows que utiliza una base de datos local. Este software les permite realizar gestiones del ganado sobre el que tienen control como por ejemplo, registrar nuevo ganado, organizarlo en ganaderías, añadir muestras...

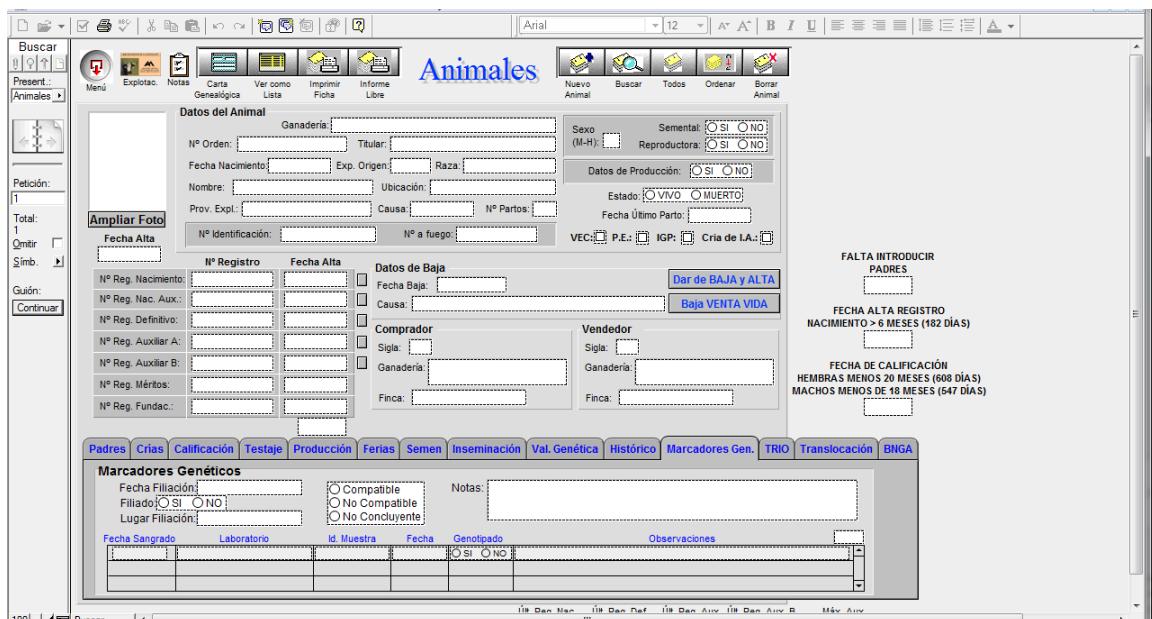


Ilustración 3 Aplicación utilizada por la Asociación de la Morucha.

Además también se utiliza una plataforma web que proporciona el Ministerio de Agricultura y Ganadería que funciona a modo de base de datos de ganado y sus datos genéticos llamada **FILOVI [21]**.

**Enlaces de interés:**  
Alta masiva por fichero

**DATOS DE ALTA DE UN NUEVO ANIMAL**

En el siguiente formulario complete los datos necesarios para dar de alta al animal.

**Datos relativos al animal**

Identificador del animal \*

Id. temporal (crotal)

Raza \*

macho  hembra

Animal procedente de explotación / centro de selección (chequeado en caso de Bovinos)

Código \*

Fecha de nacimiento \*

Tipo de parto \*

\* Campo obligatorio

Página principal | ENESA | FROM | FEGA | AAO | Información y atención al ciudadano | Tienda virtual de publicaciones

**Ilustración 4 Formulario de registro de un animal en la plataforma y base de datos FILOVI proporcionada por el Ministerio de Agricultura y Pesca, Alimentación y Medio Ambiente.**

Aparte de las herramientas que se utilizan en la Asociación de la Morucha, podemos encontrar algunas otras en internet que se expondrán a continuación.

## Memoria

### 3.2.1.1. Bovi Control

Tipo de Consulta: 921 57 42 02 -- 669662900 ganaderosonline@hotmail.com ganaderosonline.com Análisis individual de Reproductoras

Procedencia:	Propia	626	2287	Situación:	Alta	-	Crotal-DIB		STOP	Ejecución de Reproducción	Historico de Hembras
Llegada:	23/06/11	Nacimiento:	30/10/02	Partos Totales:	8		Nº Asociación			Modificación	Tratamientos Ficha REV
Localización:			12 #####	Destetes Totales:	8		Tag			Nolas Cualidades Sucesos	
Último Estado:	Gestacion	20/01/15	Días 34	Per Año y días:	2,48 146,9 9		Depar o jaula			Progenitores Hijas Pedigrí	
PFProducción:	06/07/11	17/02/15	Banda: 60	Vivos Año:	27,04		Hembra Seleccionada:	124			
Crotal elect:				Destetados Año:	25,2						
es de Lactacion:				Padre:							
Días Primera cubricion:	3171			Madre:							
Sin Avisos											

**2288**

Red sión ciclo	Ver Salto	Int DC	DC sem ana	Fecha Cubricion	Ind uci do	Res	Cumple	Diagnosticos	Indu cido	dias entre partos	Fecha parto	Gestacion seman ana	Dia	Muertos	Mori cid ad	Adop tado	Baja	Cola	Nodri zas	Dest etad os	Fecha destete	Dias Lact	Ceb Bag	%
1 1	1	1	3	06/07/11	64		28/10/11	Positivo			25/10/11	111 2	14			5		9		24/11/11	30			
2 1	1	1	39	39 1	59		25/04/12	Positivo			183 25/04/12	114 3	11 1	8				11		24/05/12	29			
3 1	1	6	6	30/05/12	59		21/09/12	Repetida	40															
3 2	2	6	46 1	09/07/12	61		31/10/11	Positivo			187 29/10/12	112 1	12							26/11/12	28			
4 1	2	5	5	09/11/11	59		03/03/13	Positivo			125 03/03/13	114 7	13 2	13	-1	3		9		04/04/13	32			
5 1	2	4	4 1	08/04/13	60		31/07/13	Positivo			150 31/07/13	114 3	14 1	7	-1			13		29/08/13	29			
6 1	2	4	4 1	02/09/13	60		25/12/13	Positivo			145 23/12/13	112 1	11 7	39		3		8		23/01/14	31			
7 1	1	5	5	28/01/14	60		22/05/14	Repetida	29															
7 2	2	5	34 3	26/02/14	61		20/06/14	Repetida	14															
7 3	1	5	48 3	12/03/14	62		04/07/14	Positivo			194 05/07/14	115 6	3			8		11		31/07/14	26			
8 1	1	5	5 2	05/08/14	62		27/11/14	Positivo			145 27/11/14	114 4	9 4	31	2	1		10		26/12/14	29			
9 1	1	25	25 2	20/01/15	63		14/05/15	Positivo										0						

**Sumas Totales**

Nacidos	102	Vivos	87	Muer.	15	Mora.	0	% Perd.	24	Total.	36,27	Destetados	71
---------	-----	-------	----	-------	----	-------	---	---------	----	--------	-------	------------	----

**Obr**

Partos	9	Vivos	#####	Muertos	0	Pro Coles	0	Pro bejas	4	Pro Dest.	8,88	Pro Nodri	0,00	Pro Repe.	6,92	Destetados	9
--------	---	-------	-------	---------	---	-----------	---	-----------	---	-----------	------	-----------	------	-----------	------	------------	---

**Carracos**

Ilustración 5 Aplicación Bovi Control

Es una aplicación para Windows desarrollada por JADA Informática para la Ganadería [16]. Nos permite la gestión de la ganadería al completo, así como la realización de cálculos de los costes de producción. Los datos de la aplicación se guardan de forma local.

En su página web incluyen un formulario para pedir presupuesto, pero no tienen un precio disponible al público.

### 3.2.1.2. Vaquitec

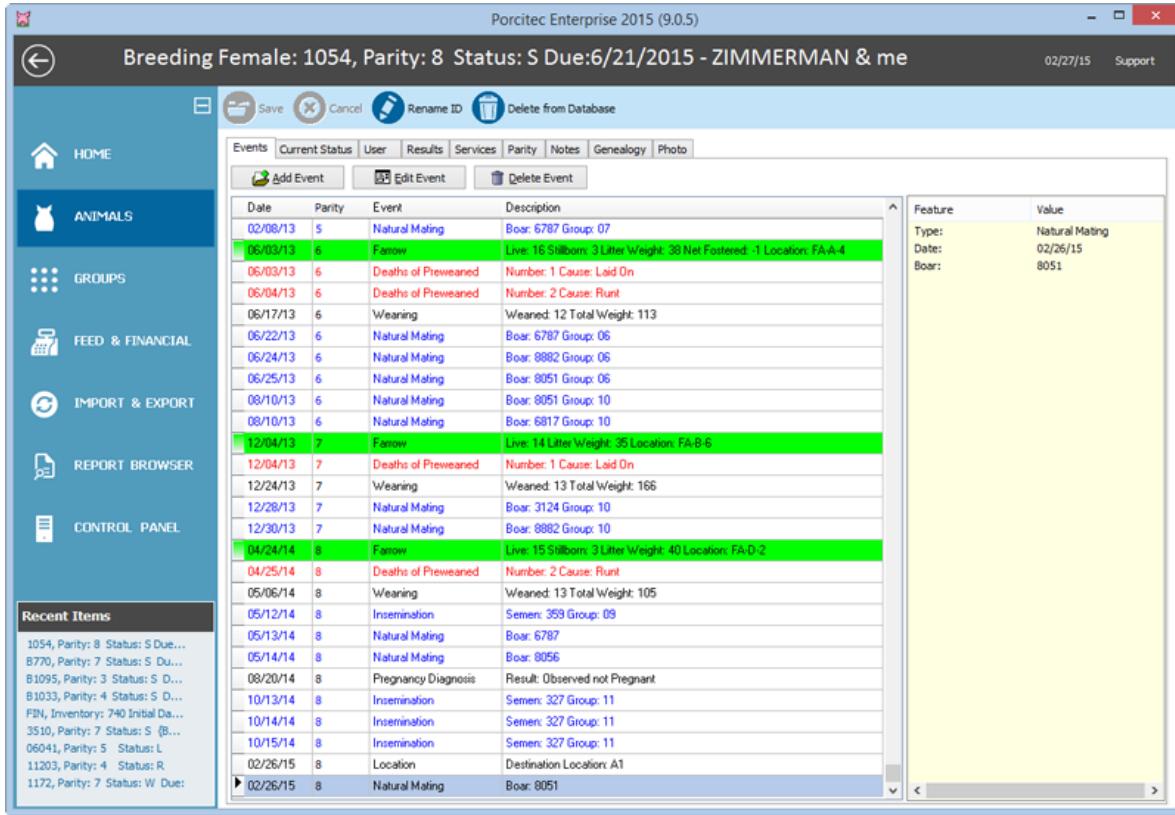


Ilustración 6 Aplicación Vaquitec.

Consiste en una aplicación Windows desarrollada por Agritec [1]. Nos permite gestionar ganaderías por completo. Ademas de funcionar con una base de datos en la nube tiene también otra aplicación web donde se pueden realizar las mismas funciones. La aplicación tiene un sistema de roles y responsabilidades por lo que el acceso a lectura o modificación de los datos dependerá de ello.

Su versión más popular requiere del pago anual de 61€ por año e incluye la mayoría de funcionalidades, pero no las funcionalidades de base de datos ubicua ni el acceso a la aplicación web. Para tener acceso a dichas funcionalidades se deberá de abonar 273€ al año.

### 3.2.1.3. Comparativa y conclusiones

	Aplicación A.M.	Bovi Control	Vaquitec
Aplicación Web	No	No	Posibilidad mediante suscripción
Almacenamiento de datos en servidor	No	Possible pero no por defecto	Sí
Almacenamiento de muestras recolectadas	Sí	No	Sí
Algoritmos de análisis genético	No	No	No
Gestión desde el laboratorio	No	No	No
Precios	Presupuesto según necesidades	No disponible / Presupuesto	273€/Año

Como podemos ver ninguna de ellas cuidan el apartado de gestión genética y relación con el laboratorio, además no suelen tener respaldado en bases de datos alojadas en servidores accesibles mediante internet y en la mayoría de los casos se requiere instalar una aplicación de escritorio para utilizarlas, es decir no tienen una versión web.

Este tipo de software suele crearse adaptándose estrechamente a las necesidades del cliente por lo que los presupuestos varían bastante.

### 3.2.2. Análisis Genético

En cuanto al análisis genético, se consultó a los expertos del Laboratorio de Innovagenomics que nos explicaron como lo realizaban actualmente filiaciones paternales. Gran parte del proceso es totalmente manual. Una vez analizadas las muestras mediante el uso de maquinaria especializada y obtenidos los valores de los alelos para los diferentes marcadores necesarios se comienza el cálculo de probabilidad paternal. Para esto se utilizan grandes hojas de Microsoft Excel donde se encuentran guardados dichos marcadores, también se utilizan para ayudarse a la hora de realizar los propios cálculos. Para estos cálculos se aplican las fórmulas de Evett y Weir [11] pero estas varían sus factores según las coincidencias de los valores de los alelos de los ganados analizados, dicha tarea la tiene que realizar la persona que lo analiza.

En definitiva, el proceso de análisis es muy poco automatizado y requiere de un largo periodo de tiempo la obtención de resultados.

### 3.2.3. Intercambio de datos

Para el intercambio de información entre la asociación ganadera y el laboratorio, en el caso de la Asociación de la Morucha e Innovagenomics se utiliza el envío de correos electrónicos en los que se adjuntan los datos para los análisis entre los que se incluyen los datos genéticos.



## 4. Técnicas y Herramientas

### 4.1. HTML, CSS y JavaScript



Ilustración 7 Logo de HTML, CSS y JS

#### 4.1.1. HTML

**HTML**, sigla en inglés de **HyperText Markup Language** (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C), organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, vipers etc.) No obstante, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión

del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Por lo que un navegador desactualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores todavía mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, todavía existen diferencias entre distintos navegadores y versiones al interpretar una misma página web [26].

Para el desarrollo del proyecto se ha utilizado HTML y HTML5 para por ejemplo utilidades como el “drag & drop”.

#### 4.1.2. CSS

Hojas de estilo en cascada (o CSS, siglas en inglés de **Cascading Stylesheets**) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. También permite aplicar estilos no visuales, como las hojas de estilo auditivas.

Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web, y GUIs (Graphic User Interface) para muchas aplicaciones móviles (como Firefox OS).

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML comparten un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

La separación del formato y el contenido hace posible presentar el mismo documento marcado en diferentes estilos para diferentes métodos de renderizado, como en pantalla, en impresión, en voz (mediante un navegador de voz o un lector de pantalla, y dispositivos táctiles basados en el sistema Braille. También se puede mostrar una página web de manera diferente dependiendo del tamaño de la pantalla o tipo de dispositivo. Los lectores pueden especificar una hoja de estilos

diferente, como una hoja de estilos CSS guardado en su computadora, para sobrescribir la hoja de estilos del diseñador.

La especificación CSS describe un esquema prioritario para determinar que reglas de estilo se aplican si más de una regla coincide para un elemento en particular. Estas reglas, aplicadas con un sistema llamado en cascadas, las prioridades son calculadas y asignadas a las reglas, así que los resultados son predecibles.

La especificación CSS es mantenida por el World Wide Web Consortium (W3C). El MIME type text/css está registrado para su uso por CSS descrito en el RFC 2318 (March 1998). El W3C proporciona una herramienta de validación de CSS gratuita para los documentos CSS [25].

Para el desarrollo del proyecto se han utilizado la versión CSS3.

#### 4.1.3. JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio del 2015.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como **AJAX**. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML [27].

#### 4.1.4. Twitter Bootstrap

Es importante señalar, que para el diseño de la página web o “front end” se ha utilizado el framework Bootstrap (también conocido como Twitter Bootstrap) que combina las tres tecnologías anteriormente mencionadas (HTML, CSS y JavaScript) y que nos ayuda a crear interfaces web de una forma más fácil y rápida.

Para el desarrollo del proyecto se utiliza la versión “3.3.7”.

#### 4.1.5. jQuery

**jQuery** es una biblioteca multiplataforma de **JavaScript**, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio [28].

Para el desarrollo del proyecto se ha utilizado la versión “3.2.1”.

##### 4.1.5.1. AJAX

**AJAX**, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor.

JavaScript es un lenguaje de programación (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM) [24].

## 4.2. PHP



**Ilustración 8 Logo de PHP**

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP [30].

Para el desarrollo del proyecto se utiliza la versión recientemente lanzada de PHP 7.

## 4.3. Patrón MVC

El patrón Modelo-Vista-Controlador es un patrón arquitectónico usado para el diseño de aplicaciones. Este patrón permite dividir la aplicación en 3 capas diferentes:

**Modelo:** contiene la lógica de negocio, es decir, la información que se almacenará en la base de datos de la aplicación. Es independiente al resto de las capas.

**Vista:** compuesta por los módulos que se encargan de mostrar la información al usuario además de permitir la interacción con éste. La vista obtiene la información que muestra del modelo. Es posible que existan diferentes vistas para el mismo modelo.

**Controlador:** se encarga de responder a los eventos que van ocurriendo en la vista, a las peticiones del modelo para que la vista se actualice. El usuario utiliza la vista para interactuar con los controladores.

Laravel, el framework PHP utilizado para el desarrollo, hace uso el patrón MVC de la siguiente forma:

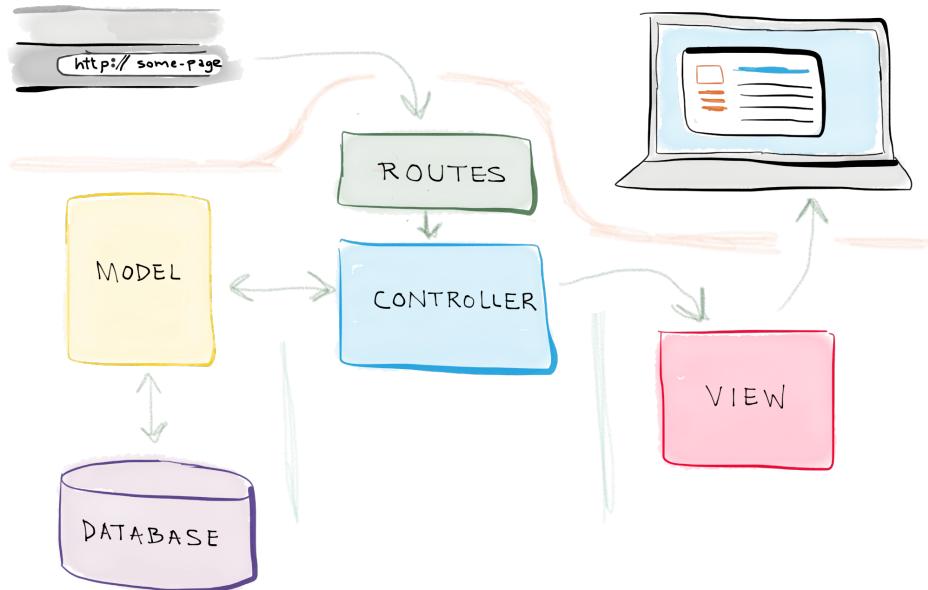


Ilustración 9 MVC adaptado a Laravel

#### 4.4. Laravel Framework



Ilustración 10 Logo de Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Gran parte de Laravel está formado por dependencias, especialmente de Symfony (se suele decir que está basado en Symfony), por lo tanto el desarrollo de Laravel depende en parte del desarrollo de sus dependencias [8].

Sus características principales son:

- Sistema de ruteo (RESTful)

- Blade, Motor de Plantillas
- Peticiones Fluent
- ORM Eloquent
- Basado en Composer
- Soporte para MVC
- Uso de componentes Symfony
- Unidad de pruebas unitarias (Desarrollo Orientado a Tests)

Para la realización de este trabajo se ha utilizado Laravel en su última versión LTS (Long Term Support, Soporte a Largo Plazo) que en el inicio del desarrollo del proyecto es la : "5.1".

#### 4.4.1. Estructura de los directorios en la aplicación

Los directorios mas importantes y necesarios para comprender el funcionamiento de una aplicación de Laravel 5.1 son:

- app/: por así decir el "hogar" para nuestro código (back-end). Aquí encontraremos el directorio "http" con nuestros "Controllers", "Middlewares", "Requests" y el archivo "routes.php" para nuestras rutas. En el directorio raíz de app, encontraremos los modelos, por defecto nos encontramos con el modelo "User.php".
- config/: donde encontraremos ficheros como "app.php" donde configurar características elementales de nuestra aplicación como "aliases", "providers"... Otros como por ejemplo database.php para configurar nuestro motor de bases de datos.
- database/: en este directorio se encontrarán los archivos relacionados con el control de las bases de datos como los mencionados anteriormente: "factories", "migrations", "seeds".
- public/: donde se alojarán diferentes recursos de la aplicación, como pueden ser los archivos CSS, JS, o imágenes. Se ha subdividido en carpetas con esos mismos nombres para que sea más accesible a la hora de trabajar con ello.
- tests/: donde se localizan los archivos de tests.
- .env: el cual tiene un respaldo .env.example donde se encuentra la configuración de ejecución de nuestra aplicación Laravel.

## 4.5. MySQL



Ilustración 11 Logo de MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

Está desarrollado en su mayor parte en ANSI C y C++. Tradicionalmente se considera uno de los cuatro componentes de la pila de desarrollo LAMP y WAMP. MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr, y YouTube [34].

Se ha utilizado para el almacenamiento de la base de datos que utiliza Laravel para sus migraciones, en su versión “5.7.18”.

## 4.6. phpMyAdmin



Ilustración 12 Logo de phpMyAdmin

phpMyAdmin es una herramienta de software libre escrita en PHP con la intención de manejar la administración de MySQL y MariaDB a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas. Se encuentra disponible bajo la licencia GPL Versión 2 [3].

Se ha utilizado esta herramienta para agilizar el uso de la base de datos MySQL mientras se trabajaba, en su versión “4.5.4”.

## 4.7. Apache



**Ilustración 13 Logo Apache Server**

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y energético pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la Apache Software Foundation dentro del proyecto HTTP Server (httpd).

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración [14].

El servidor apache se utiliza para el despliegue de la aplicación y también para otros servicios como por ejemplo phpMyAdmin.

Para el desarrollo del proyecto se ha utilizado Apache “2.4”.

## 4.8. PhpStorm



Ilustración 14 Logo de PhpStorm

JetBrains PhpStorm es un IDE multiplataforma creado por JetBrains para el desarrollo PHP. Forma parte de la plataforma IntelliJ IDEA a la cual nos proporciona acceso y licencias la Universidad de Salamanca.

Se ha utilizado este IDE para el desarrollo ya que tiene módulos adaptados a Laravel que facilitan bastante el desarrollo a la hora de aportar sugerencias, resaltado...

Para el desarrollo del proyecto se utiliza la versión de PHP Storm 2016.2.

## 4.9. DreamWeaver



Ilustración 15 Logo de DreamWeaver

DreamWeaver es un software gráfico para el diseño y creación de páginas web desarrollado por Adobe. Aun que tiene posibilidad de escribir código HTML, CSS y JavaScript como si de un editor de texto se tratase intenta utilizar la parte gráfica al máximo posible lo cual lo hace muy intuitivo. Fue utilizado inicialmente para el desarrollo front-end de la web como “aclimatamiento” a los tres lenguajes front-end y a Twitter Bootstrap (ya que lo soporta), aun que más adelante y al ir mejorando en la codificación pura se terminó desecharlo. En mi opinión me fue bastante útil para el aprendizaje inicial.

Para el desarrollo del proyecto se ha utilizado la versión Dreamweaver CC 2017.

## 4.10. Sketch



**Ilustración 16 Logo de Sketch**

Se ha utilizado Sketch, un programa de diseño gráfico, para la creación de algunos logos e iconos de forma fácil. Sketch es uno de los programas más usados en el sector del diseño gráfico para la creación, prototipado, UX y UI de páginas web y aplicaciones móviles.

Para el desarrollo del proyecto se utiliza la versión 41 de Sketch.

## 4.11. Microsoft Excel



**Ilustración 17 Logo de Microsoft Excel**

Microsoft Excel es un programa informático desarrollado y distribuido por Microsoft que forma parte de la suite de ofimática Microsoft Office. Se trata de un software que permite realizar tareas contables y financieras con fórmulas y gráficos desarrolladas específicamente para ayudar a crear y trabajar con hojas de cálculo [32].

Esta herramienta es imprescindible para el desarrollo del proyecto ya que se utiliza su formato a la hora de importar y exportar información hacia y desde nuestro sistema.

Para el desarrollo del proyecto se utiliza Microsoft Office 2016.

## 4.12. Herramientas CASE

CASE (Computer-Aided Software Engineering – Inginería de Software Asistida por Ordenador) es el nombre que se utiliza para denominar a las aplicaciones informáticas que semi-automatizan y mejoran la productividad de las tareas de ingeniería del software reduciendo el costo en términos de tiempo y dinero.

### 4.12.1. UML (Lenguaje Unificado de Modelado)



Ilustración 18 Logo de UML

El lenguaje unificado de modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es utilizado para visualizar, especificar, construir y documentar partes de un sistema de software desde distintos puntos de vista [22]:

- Puede usarse con cualquier proceso de desarrollo, a lo largo de todo el ciclo de vida y puede aplicarse a todos los dominios de aplicación y plataformas de implementación
- También puede usarse en otras áreas, como la ingeniería de negocio y modelado de procesos gracias a los mecanismos de adaptación/extensión mediante perfiles.

#### 4.12.1.1. Visual Paradigm



Ilustración 19 Logo de Visual Paradigm

Visual Paradigm es una herramienta CASE para UML que soporta UML 2. Además de soporte para el modelado, también nos aporta generación de reportes y opciones de ingeniería de código como puede ser la generación

de código. También puede realizar ingeniería inversa para crear diagramas a partir de código.

#### 4.12.2. REM

REM (REquirements Management) es una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software de acuerdo con la metodología definida en la Tesis Doctoral "Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información", presentada por Amador Durán en septiembre de 2000 [6].

#### 4.12.3. Microsoft Project



Ilustración 20 Logo de Microsoft Project 2016

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo [29].

Se ha utilizado a lo largo del proyecto para realizar la planificación temporal y los diagramas de Gantt, utilizando su última versión la lanzada en 2016.

#### 4.12.4. EZEstimate

Herramienta utilizada para realizar la estimación de esfuerzo del proyecto.

#### 4.12.5. phpDocumentor



**Ilustración 21 Logo de phpDocumentor**

PhpDocumentor es la herramienta de código abierto de documentación para el lenguaje PHP y que puede ser usado directamente desde la línea de comandos. Basado en el estándar PhpDoc (aun proceso de formalizarse), utiliza una sintaxis similar a JavaDoc. Puede ser usado para generar documentación de forma profesional directamente desde el código fuente de tu proyecto tanto para generar documentación en programación procedural como orientada a objetos [31].

Se ha utilizado este sistema de documentación sobre otras alternativas como ApiDoc por la facilidad de su instalación y uso mediante Composer.

#### 4.12.6. Git (VCS)



**Ilustración 22 Logo git**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux [13].

Se ha utilizado Git desde el principio del proyecto para realizar el control de versiones.

#### 4.12.6.1. GitHub



**Ilustración 23 Logo de GitHub**

Combinado con Git, se ha utilizado la plataforma GitHub. Es la plataforma más importante para el alojamiento de repositorios software Git normalmente de forma pública con licencias Open-Source, GPL o MIT. Sin embargo, la Universidad de Salamanca nos proporciona acceso de estudiante por lo que podemos crear repositorios privados también.

Es el sitio web donde se encuentran los repositorios de frameworks como por ejemplo Twitter Bootstrap, el propio Laravel...

Llama la atención que durante el plan de estudios y sobre todo en cursos iniciales no se dedique un mínimo de tiempo y explique de forma práctica el uso de estas dos herramientas combinadas ya que una vez las he aprendido a usar me ha resultado de gran utilidad y han facilitado mucho el aprendizaje.



## 5. Aspectos relevantes del desarrollo

A continuación se exponen los detalles y consideraciones que han sido más importantes en el proceso de desarrollo, lo que incluye:

Metodología.

- Implementación del algoritmo de análisis genético.

### 5.1. Metodología

Para la realización del proyecto se han seguido y utilizado las directrices marcadas por el **Proceso Unificado** [23]. El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (UML) para el desarrollo de la mayoría de las fases del proyecto. El Proceso Unificado de Modelado es extensible, de modo que se puede aplicar a una gran variedad de proyectos, incluyendo el diseño y la construcción de aplicaciones Web.

El proceso unificado está caracterizado por su desarrollo a través de iteraciones, cada iteración finalizará con la consecución de hitos, incorporando nuevos incrementos al proyecto. De esta forma podemos controlar mejor el progreso del proyecto, realizar estimaciones temporales y de recursos, y ayudar en la toma de decisiones.

Para la elaboración del proyecto se describen a continuación las diferentes iteraciones que se han ejecutado durante el transcurso del desarrollo del proyecto:

Iteración	Fase	Descripción	Fechas
Primera	Inicio	Se define el alcance del proyecto con los tutores. Se realiza también la <b>especificación</b> de la mayoría de los <b>requisitos</b> del proyecto.	20/12/2016-1/1/2017
Segunda	Elaboración	Se comprueban los requisitos. A partir de éstos se realiza el <b>análisis del sistema de sistema software</b> .	13/2/2017-16/3/2017
Tercera	Elaboración	Partiendo del análisis del sistema, se realiza el <b>diseño del sistema software</b> .	17/3/2017-4/4/2017

Cuarta	Construcción	Basándonos en las tres iteraciones anteriores se realiza la construcción del proyecto.	4/4/2017-12/6/2017
Quinta	Transición	<b>Se revisa el estado del sistema</b> , los posibles errores de funcionamiento o de diseño, y se trata de corregirlos.	12/6/2017-20/6/2017

Para más información sobre el proceso, se recomienda consultar el ANEXO I: Plan de Proyecto Software.

### 5.1.1. Primera Iteración

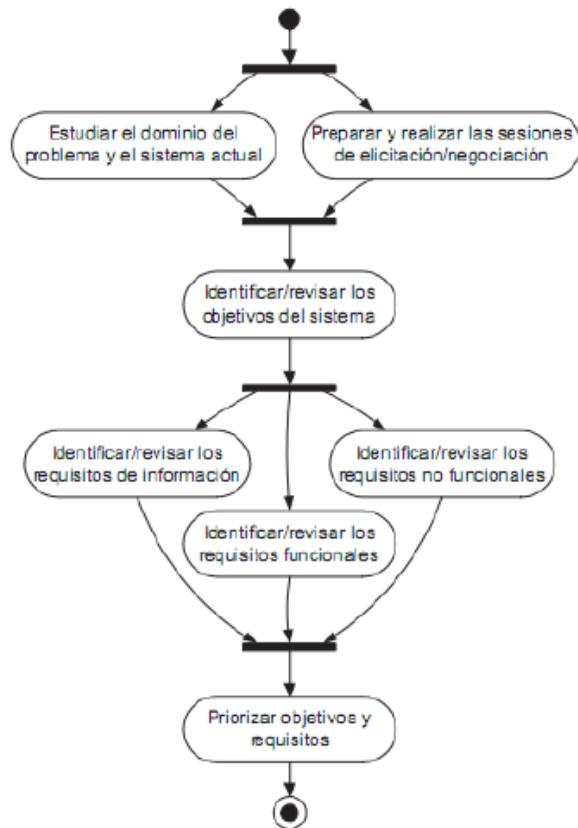


Ilustración 24 Elicitación de requisitos

Se establece una reunión inicial en la que se estudia y discuten los aspectos relevantes a desarrollar en el proyecto en cada uno de los planos que se va a desarrollar.

El punto de vista del cliente es aportado por la tutora Elena Solera. Fernando de la Prieta contribuye tanto en la recomendación y consejo en el uso de tecnologías como en el desarrollo de la propia Ingeniería del Software colaborando en todas las fases de ésta. El tutor Juan Ramos actúa como nexo entre las cuestiones técnicas de tema biológico e informático.

Tras la primera reunión obtuvimos un pequeño boceto de lo queríamos que la aplicación fuera. Gracias a esto empezamos a elaborar el catálogo de requisitos mediante la Metodología de Durán y Bernárdez [7], obteniendo las tablas de requisitos con la herramienta CASE proporcionada por la Universidad de Sevilla REM (REqueriments Management).

UC-0001	Registrar Ganadería	
Versión	1.0 ( 14/03/2017 )	
Autores	<ul style="list-style-type: none"> <li>• <a href="#">Pablo Espinosa Bermejo</a></li> </ul>	
Fuentes	<ul style="list-style-type: none"> <li>• <a href="#">Elena Solera</a></li> </ul>	
Descripción	<i>Permite dar de alta una Ganadería en el sistema.</i>	
Secuencia normal	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Administrador Asociación (ACT-0003)</a> hace click en "Nueva Ganadería".
	2	El sistema muestra los campos necesarios a llenar para la nueva ganadería.
	3	El actor <a href="#">Administrador Asociación (ACT-0003)</a> Introduce los datos necesarios sobre la nueva ganadería y hace click en "Registrar".
	4	Se realiza el caso de uso <a href="#">Realizar búsqueda (UC-0003)</a>
	5	El sistema guarda los datos de la nueva Ganadería en la Base de Datos.
Postcondición	Se crea una ganadería.	
Excepciones	<b>Paso</b>	<b>Acción</b>
	3	Si se ha introducido un tipo de dato erróneo o no se ha rellenado un campo obligatorio, se realiza el caso de uso <a href="#">Mostrar Error (UC-0002)</a> , a continuación este caso de uso continúa
	4	Si se encuentra en el sistema una Ganadería igual que la que acabamos de introducir, se realiza el caso de uso <a href="#">Mostrar Error (UC-0002)</a> , a continuación este caso de uso continúa

**Ilustración 25 Especificación de Requisitos con la herramienta REM**

Además, se comienza a investigar y decidir el tipo de herramientas que se van a utilizar para el desarrollo del proyecto. En este caso se desarrolla como una plataforma o aplicación web ya que nos aporta exactamente lo que estamos buscando, tanto la flexibilidad de uso en cualquier dispositivo y desde cualquier lugar como un desarrollo medianamente rápido.

A la hora de elegir la tecnología web sobre la cual desarrollar, Fernando de la Prieta sugiere varias opciones, entre las que se encuentran Yii2, Symfony, Laravel o Django.

Tras una fase de investigación y documentación sobre los diferentes tecnologías web se elige como framework Laravel. Los motivos principales son:

- Curva de aprendizaje rápida
- Orientado a objetos
- Basado en MVC
- Gran apoyo de la comunidad en internet y soporte
- Lenguaje PHP

Son estos dos últimos factores los que nos ayudan a decantarnos por esta opción ya que el alumno ya tiene unas nociones básicas de PHP y el soporte de Laravel en internet es el mayor de todos y con gran crecimiento durante los últimos años:



**Ilustración 26 Captura de la analítica de Google Trends**

Al final de nuestra primera iteración llegamos a la consecución de nuestro primer hito:

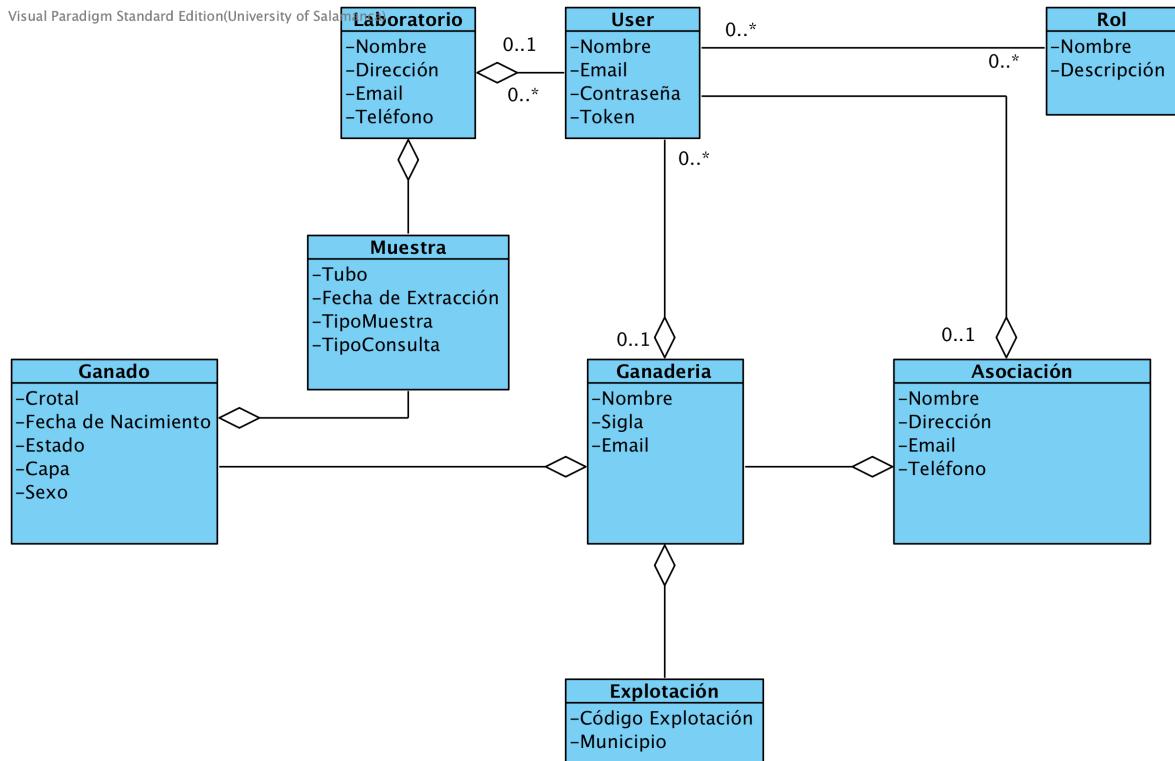
**Hito 1:** Se completa la captura y especificación de los requisitos.

### 5.1.2. Segunda Iteración

Tras la primera iteración, se organiza una reunión con todos los integrantes del proyecto para verificar el catálogo de requisitos elaborado con REM. Se realizan algunos cambios sobre estos. Fernando enfoca las etapas de Análisis y Diseño del sistema sobre las que se va a empezar a trabajar.

Después de esta reunión el alumno empieza a trabajar sobre el Análisis del Sistema, para el que se utiliza la herramienta Visual Paradigm.

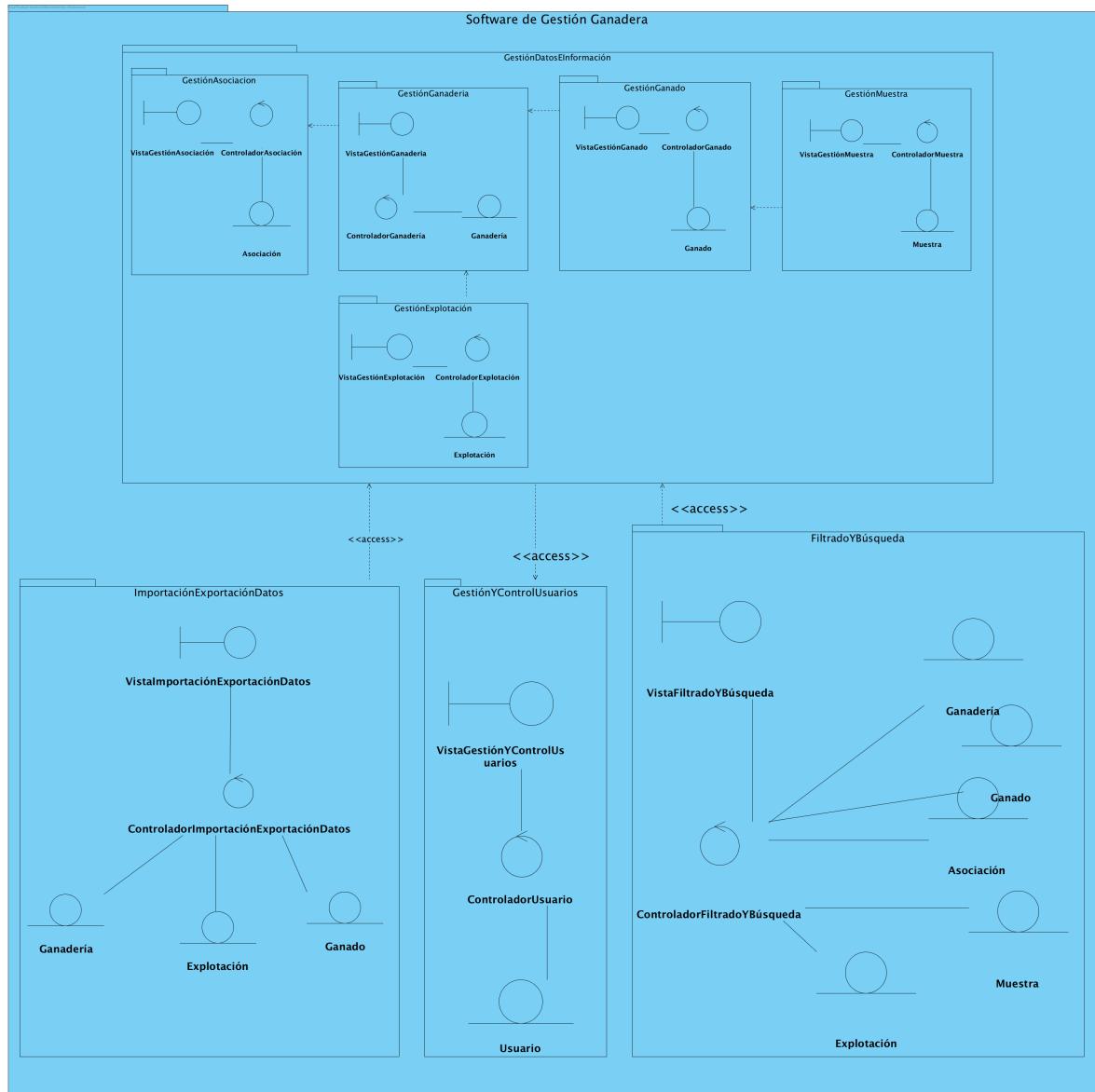
Inicialmente se crea el Modelo de dominio donde recogemos los requisitos de almacenamiento de información en nuestra Base de Datos y sus relaciones:



**Ilustración 27** **Modelo de Dominio inicial**

Más tarde se realizan los diagramas de paquetes en los que el sistema va a estar dividido. Para crear una visión general del sistema y sintetizar todos estos paquetes, se crea la vista de arquitectura:

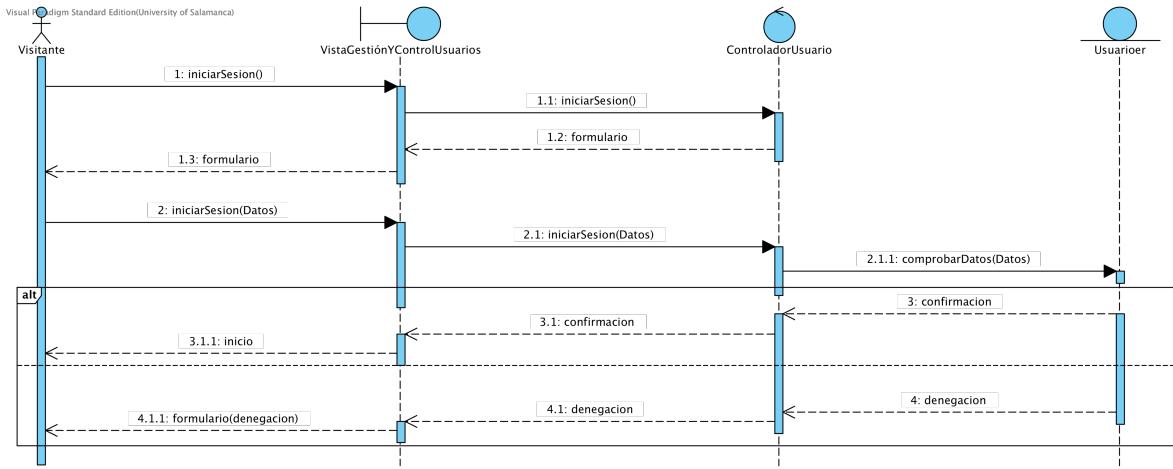
## Memoria



**Ilustración 28 Vista de Arquitectura del Modelo de Análisis**

Nos encontramos en una etapa inicial del proceso por lo que es bastante probable que la mayoría de los componentes cambien además de ni siquiera haber aplicado ningún patrón aún. Todo esto irá refinándose y mejorando según se vaya avanzando en el proceso.

También se trabaja en la realización de los casos de uso creando diagramas de secuencia como el que podemos ver a continuación:



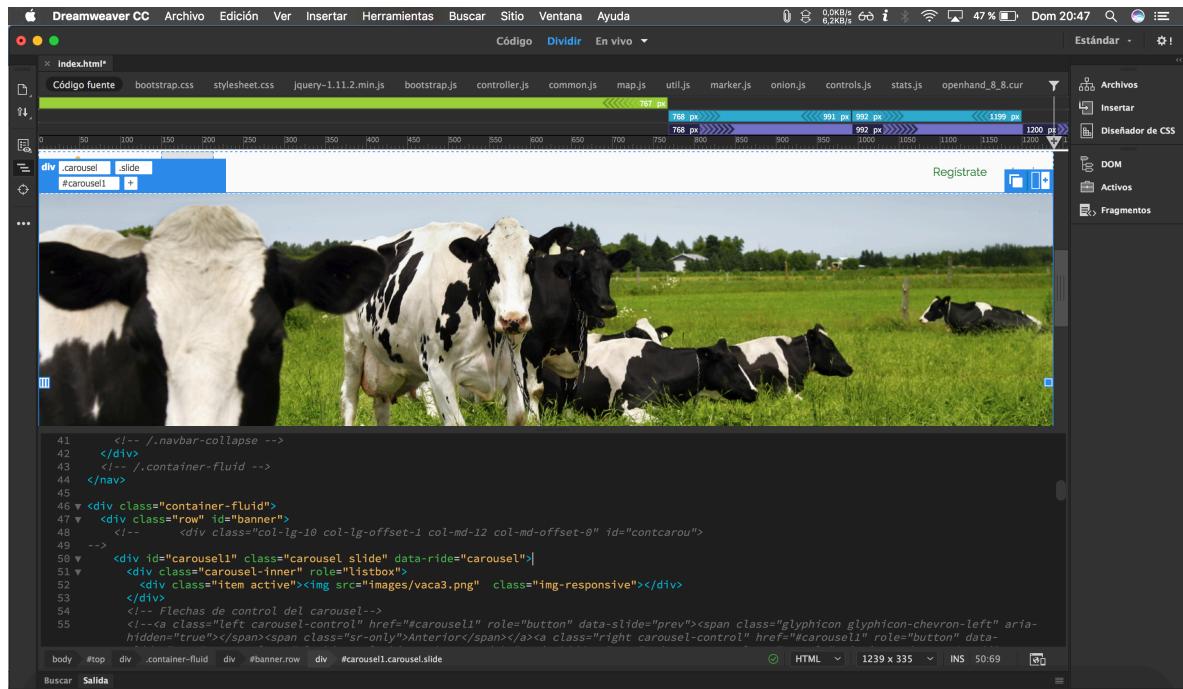
**Ilustración 29 Diagrama de Secuencia del caso de uso “Iniciar Sesión”**

Si se quiere consultar este proceso de forma más detallada se remite al lector al ANEXO III.

Mientras se realizaba el análisis del sistema y después de haber elegido Laravel como tecnología/framework para la implementación del proyecto el alumno empieza a aprender a trabajar y a familiarizarse con dicha tecnología. Es importante que cuando se realice el Diseño del sistema ya se tenga una visión global de cómo funciona el framework de forma que el diseño esté orientado en el mismo sentido que éste (Laravel).

A su vez, se comienza a crear una vista principal sobre la que en un futuro irán incrustadas las que aporten funcionalidad al sistema. Para esto se utiliza DreamWeaver ya que nos permite trabajar de forma muy fácil con HTML y CSS para el diseño front-end ya que el alumno tampoco está muy familiarizado con ello.

## Memoria



### Ilustración 30 Inicio del diseño de la interfaz de la web con DreamWeaver

Además, se empieza a investigar sobre otras plataformas similares que puedan existir actualmente para saber cómo enfocar la experiencia del usuario a la hora de interactuar con la aplicación. Debido a que en Internet no se encuentra demasiada información sobre la materia se consigue concertar una reunión con la Asociación de la Morucha de Salamanca.

Durante la reunión, los trabajadores de esta asociación describieron el flujo de trabajo que siguen para la gestión de las ganaderías que tienen asociadas. Enseñaron el software que utilizan actualmente, como se maneja, los datos que guardan en él, las funcionalidades más importantes...

Una vez visto esto, se les plantean varias cuestiones como por ejemplo el tipo de funcionalidades que ven necesarias y que les gustarían tener, cosas que no les gusta cómo funcionan y posibles errores o fallos técnicos que puedan tener.

Esto por ejemplo nos sirvió para verificar requisitos y casos de uso que ya contemplábamos en los requisitos como por ejemplo un módulo de importación y exportación de los datos que a Elena Solera le pareció interesante tener para manejar los datos del propio laboratorio, la asociación también lo ve bastante interesante a la hora de migrar en un futuro de su sistema actual al sistema que nosotros vamos a desarrollar por lo que se empiezan a plantear pequeñas adiciones en los casos de uso del sistema.

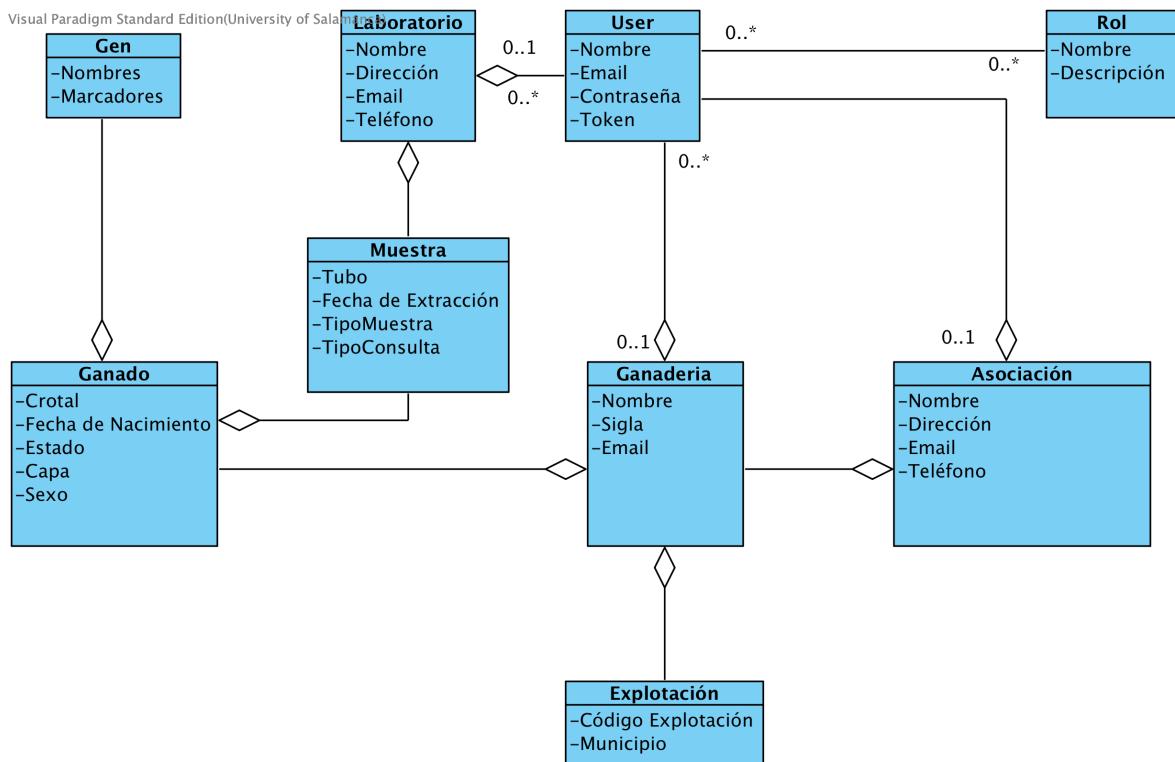
Después de esta reunión se da por concluida la iteración y por lo tanto alcanzamos un nuevo hito en nuestro proyecto:

**Hito 2:** Se completa el análisis del sistema.

### 5.1.3. Tercera Iteración

Después de la reunión los miembros del proyecto se reúnen para debatir sobre ello. Se decide que hay que darle importancia a que la migración desde otro software al que se está desarrollando sea lo más fácil e inapreciable posible por lo que nos debemos adaptar un poco al usuario y a sistemas como los que hemos visto en la Asociación de la Morucha. Además, surgen nuevos casos de uso ya que Elena Solera piensa que sería interesante poder realizar cálculos o análisis sobre los datos genéticos. Se decide abordar esta nueva funcionalidad enfocándonos en algoritmos y cálculos de filiación parental. Para el desarrollo de estos cálculos y análisis es importante que el sistema de gestión sea funcional, ya que éstos dependen fuertemente del conjunto de datos que se posee.

El problema que surge es que en el modelo actual que contemplamos no se guarda ningún tipo de dato genético por lo que se decide añadirlo a nuestro Modelo de Dominio. Estos datos genéticos estarán almacenados dentro de la entidad “Gen” y que le pertenecerá a cada “Ganado”. Gen almacenará tanto el nombre del marcador como los dos posibles valores de sus alelos. Además, tanto los requisitos de información como el Modelo de Dominio quedan mucho mejor definidos tras esta reunión:

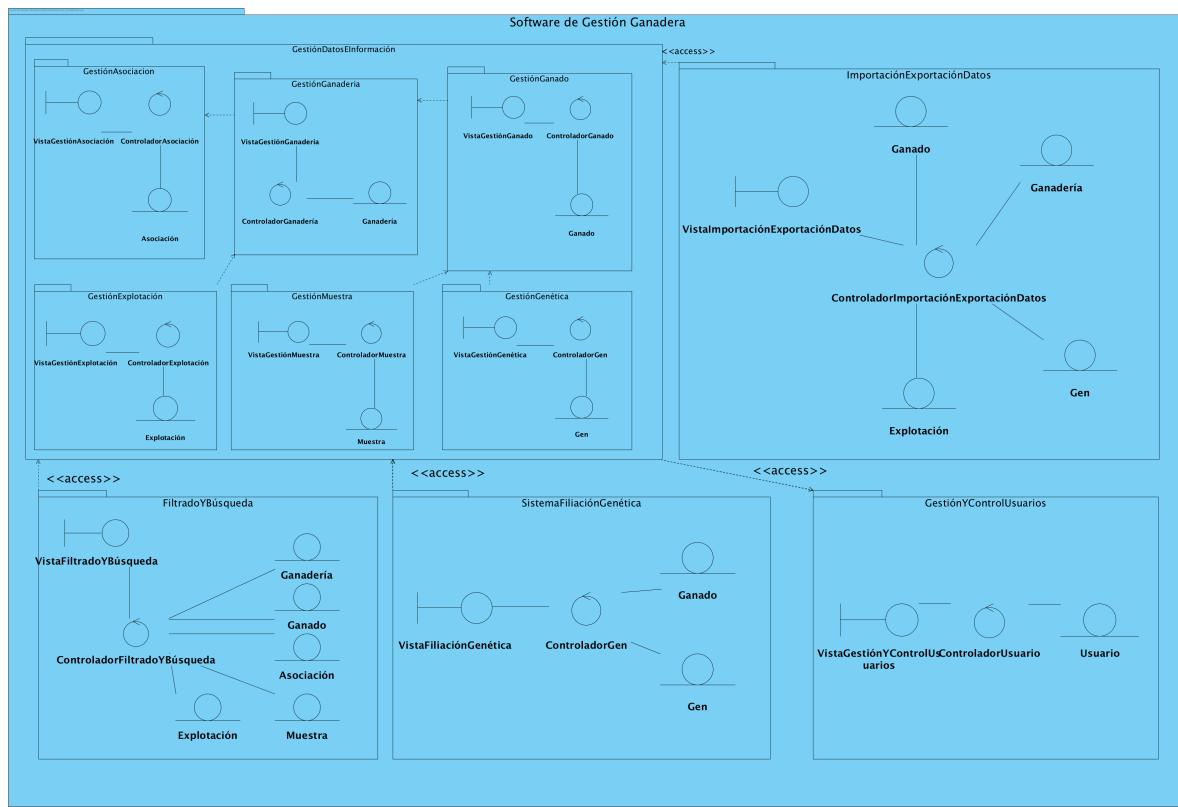


**Ilustración 31 Actualización del Modelo de Dominio**

Esto también repercute en el análisis de nuestro sistema y añade nuevos casos de uso, por lo tanto, debemos de cambiar la arquitectura de nuestro sistema. Se tendrá que añadir un módulo para la gestión de los datos genéticos que a su vez y al igual

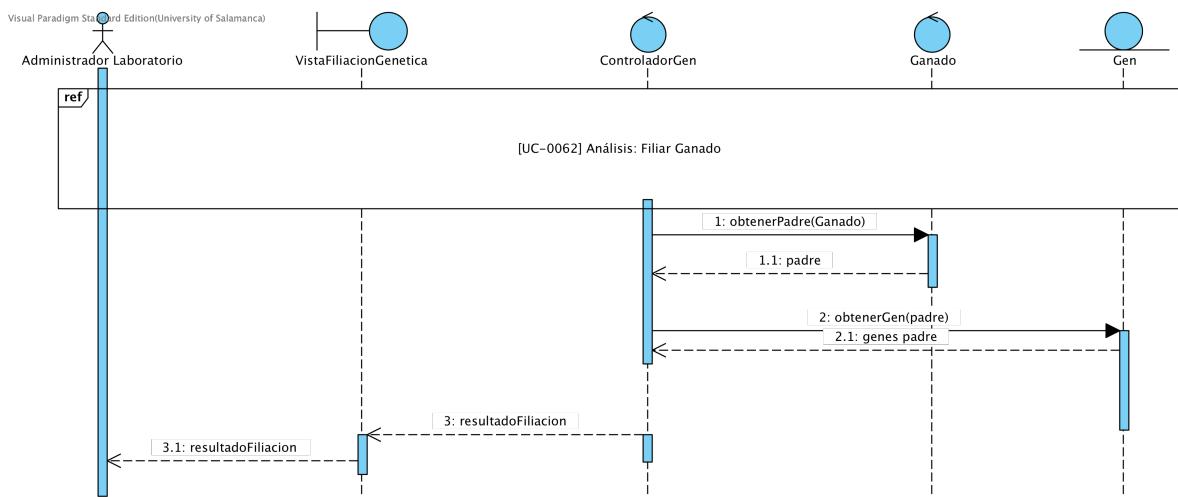
## Memoria

que en el modelo este fuertemente relacionado con el ganado. También se tendrá que crear otro módulo mediante el cual se puedan realizar los cálculos sobre los datos:



## Ilustración 32 Actualización de la vista de Arquitectura del Sistema

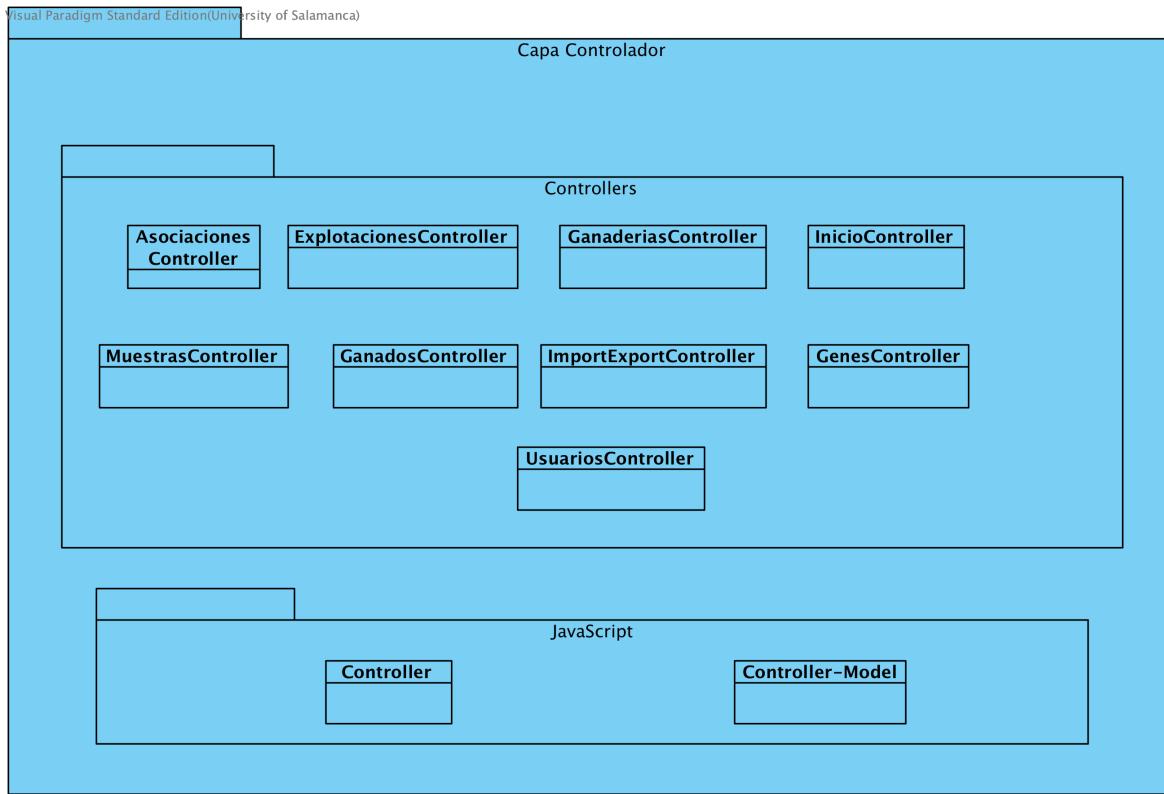
También se añaden tanto los nuevos casos de uso al catálogo de requisitos como las realizaciones de éstos al análisis de nuestro sistema:



### **Ilustración 33 Nueva realización de Casos de Uso añadidos al sistema**

Seguidamente se empieza a desarrollar el Diseño del Sistema partiendo desde lo obtenido gracias al Análisis del Sistema.

Lo primero que se realiza es el modelo de diseño, que estará basado en el patrón arquitectónico Modelo-Vista-Controlador:



**Ilustración 34 Capa controlador**

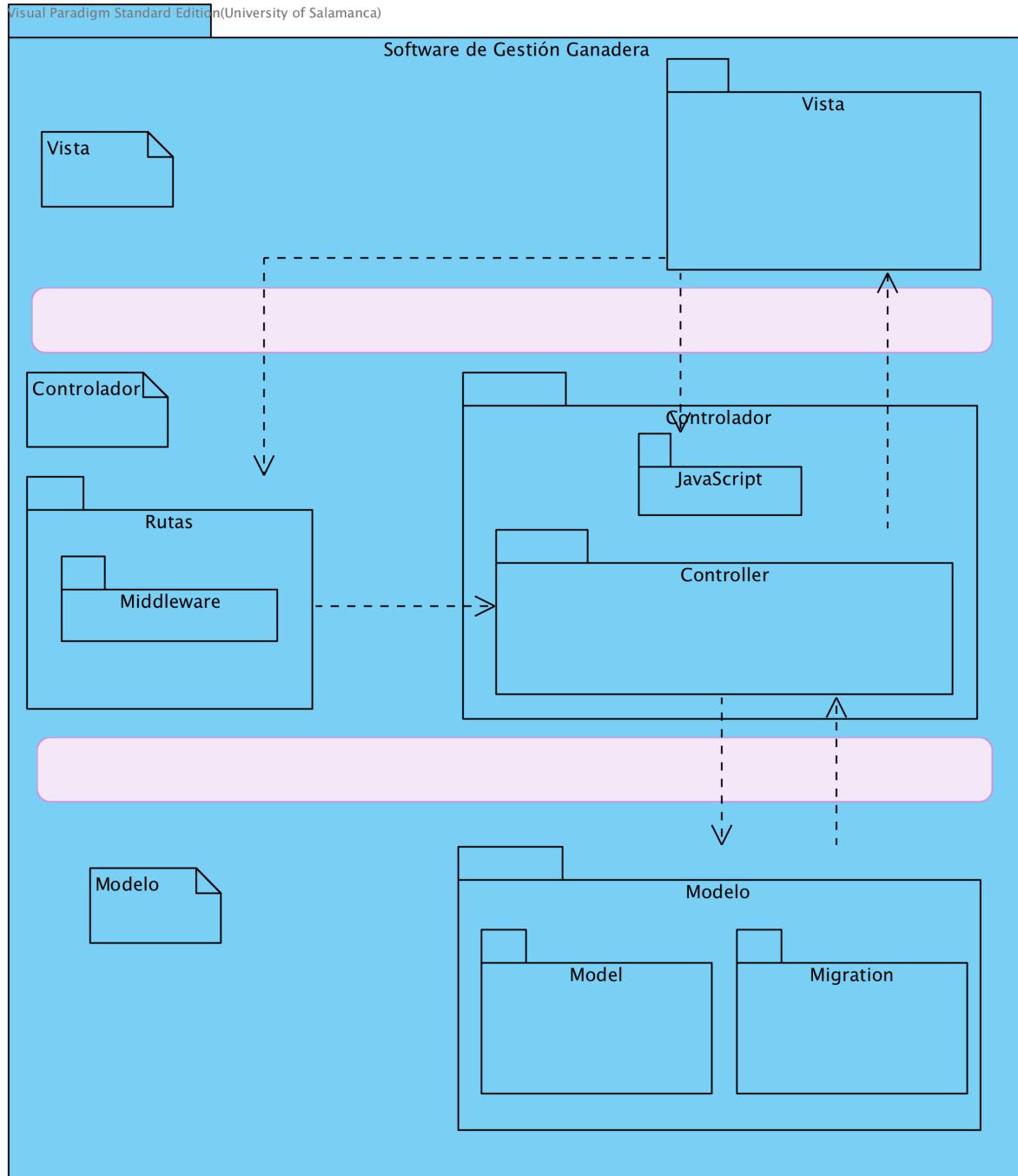
Después, se realiza un desglose de las capas, por ejemplo la capa controlador, se debe de dividir en dos: una capa que se ejecuta en el cliente en lenguaje JavaScript y otra capa que se ejecuta en el servidor programada en PHP.

La capa en el cliente está organizada en clases o funciones ya que JavaScript considera las funciones como clases y viceversa. JavaScript actúa como una interfaz entre los controladores del servidor y la interfaz que ve el usuario, enviando las peticiones y también actualizando la vista dependiendo de ciertos eventos.

En el servidor, tenemos un controlador por cada una de las clases de nuestro modelo (Asociación, Ganadería, Explotación...) a lo que añadimos también un controlador para las importaciones y exportaciones, y otro para gestionar las vistas de inicio y “landing”. También existen controladores para la autenticación y la recuperación de contraseñas. Todos estos controladores heredan de la clase de Laravel “Controller”.

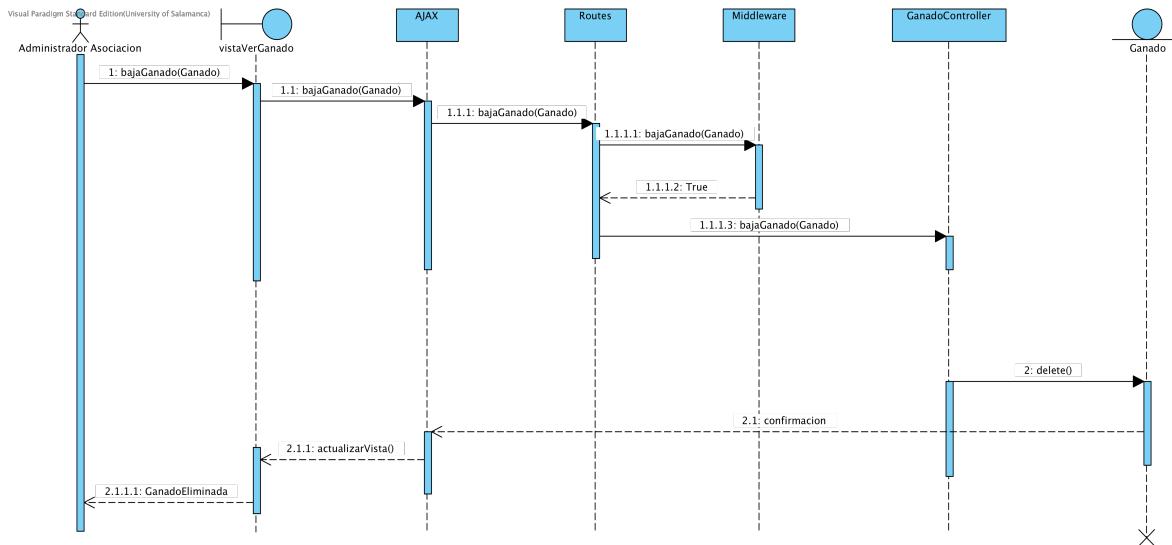
El middleware es gestionado mediante la declaración de rutas.

## Memoria



**Ilustración 35 Vista arquitectónica del diseño**

Además, conociendo ya la tecnología que se va a utilizar y el diseño de su arquitectura se refina la realización de los casos.

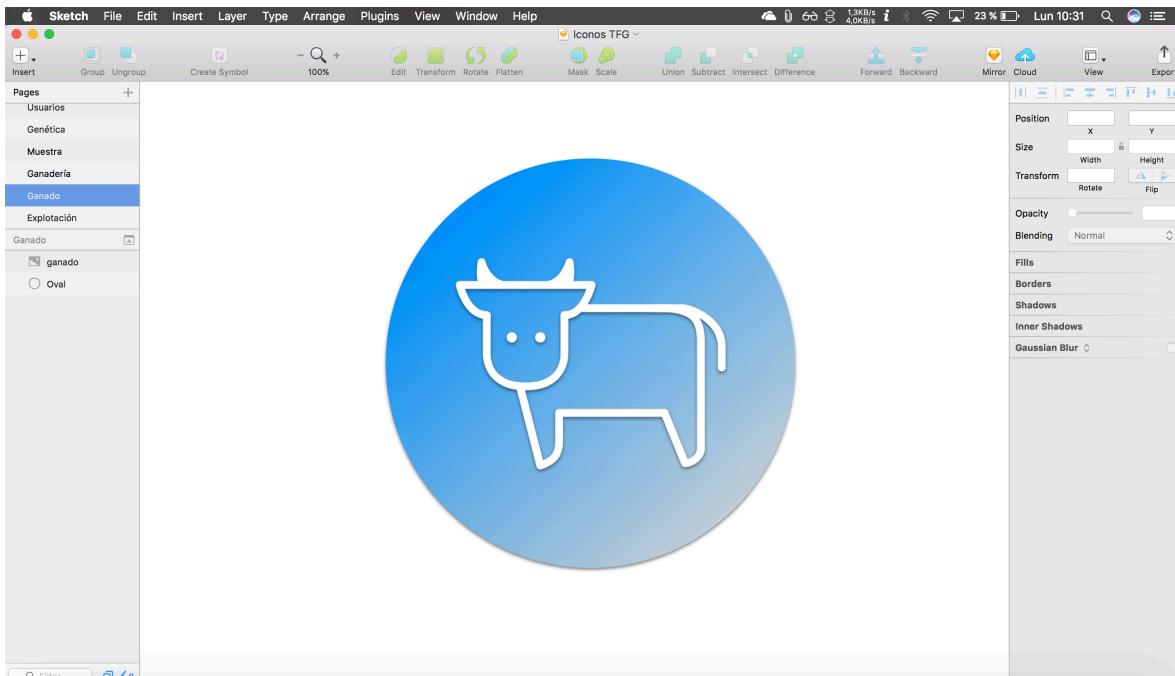


**Ilustración 36 Diagrama de Secuencia. UC-0015. Baja Ganado**

Si se quiere consultar este proceso de forma más detallada se remite al lector al ANEXO IV.

Mientras tanto, el alumno se sigue familiarizando con el entorno de desarrollo, investigando sobre los aspectos que pueden causar problemas en un futuro como son las relaciones en el modelo o la implementación de un sistema de roles.

A su vez se empiezan a diseñar algunos iconos necesarios para el panel principal de la aplicación web. Para este cometido se utiliza Sketch, una aplicación de diseño gráfico que facilita bastante este tipo de tareas:



### Ilustración 37 Diseño de los iconos de la aplicación con Sketch.

Se establecen nuevas reuniones para afrontar el nuevo módulo de filiación. Gracias a los consejos y explicaciones de Elena Solera y Juan Ramos se consigue enfocar la manera de implementar en algoritmo las fórmulas descritas por Evett y Weir en nuestra aplicación web. Para esto se hace un poco de investigación sobre la materia además de realizar manualmente algunas filiaciones para entender el funcionamiento del algoritmo.

**Hito 3:** Se completa el Diseño del Sistema Software.

#### 5.1.4. Cuarta Iteración

Se establece una reunión para revisar tanto el modelo de análisis como el dominio. Tras la confirmación de ambos modelos se establece como prioridad desarrollar la funcionalidad de gestión de la información.

El alumno que ya ha adquirido unos conocimientos básicos en Laravel comienza a desarrollar la aplicación web. Lo primero que crea es la capa del modelo, es decir, las clases llamadas *Modelo* en Laravel en las que se encuentra la lógica de negocio que será la información que se debe guardar en nuestra base de datos.

Además de los atributos de cada uno de los *Modelos* establecimos las relaciones entre ellos según lo demanda el ORM Eloquent para de esta forma en un futuro acceder fácilmente en el futuro desde uno de los *Modelos* a otros con los que tiene algún tipo de relación.

Como queremos que la base de datos tenga un almacenamiento persistente se establece la conexión de Laravel con una base de datos MySQL. Además, para que el ORM funcione de forma correcta con el *Modelo* se crean las *Migraciones*.

Mientras tanto y partiendo de la interfaz que se empezó a diseñar en las iteraciones anteriores se empiezan a añadir sobre esta las diferentes interfaces que necesitamos. Inicialmente se crean para cada *Modelo* una de registro y otra de listado, de esta forma se va viendo si el funcionamiento del sistema es correcto o no.

Para esto también se programa la capa controladora de cada uno de los *Modelos*.

Una vez se llega a un estado mínimo de funcionalidad, se establece una reunión para que los tutores puedan verlo y aportar su opinión. Se anotan algunas posibles mejoras en la parte que ya está implementada como por ejemplo el aspecto de la interfaz, pero también se comenta qué módulos empezar a desarrollar como la visualización de detalles y la edición de los elementos del modelo.

El desarrollo continúa centrándose en las partes citadas anteriormente y además el alumno añade la funcionalidad de borrado. También se desarrolla el módulo de importación y exportación que se requería inicialmente.

Posteriormente nos encontramos con un sistema de gestión de los datos lo que cumple la mayoría de requisitos comentados. Por lo tanto, se decide avanzar hacia el módulo genético y de filiación. Como se ha comentado anteriormente en esta misma memoria se comienza investigando sobre qué es una filiación y cómo se realiza a mano dicho proceso. Más tarde se añade el *Modelo Gen* en el que se almacenará la información de éstos y luego se comienza a implementar el algoritmo en sí.

Una vez terminado alcanzamos nuestro cuarto hito,

**Hito 4:** Se obtiene una versión preliminar del software.

### 5.1.5. Quinta Iteración

Se concierta una nueva reunión para mostrar el software a los tutores. Éstos encuentran algunos fallos de funcionamiento y posibles mejoras en la interfaz del sistema sobre todo en el módulo genético (el último diseñado).

El alumno se centra en corregir los errores primero y una vez hecho esto rediseña la interfaz del módulo genético.

Tras esto comienza a realizar el despliegue en un servidor público proporcionado por la Universidad, creando a la vez el manual de despliegue. También genera la documentación técnica del código programado y se crea un manual para el usuario final.

**Hito 5:** Se obtiene una versión entregable del producto de software.

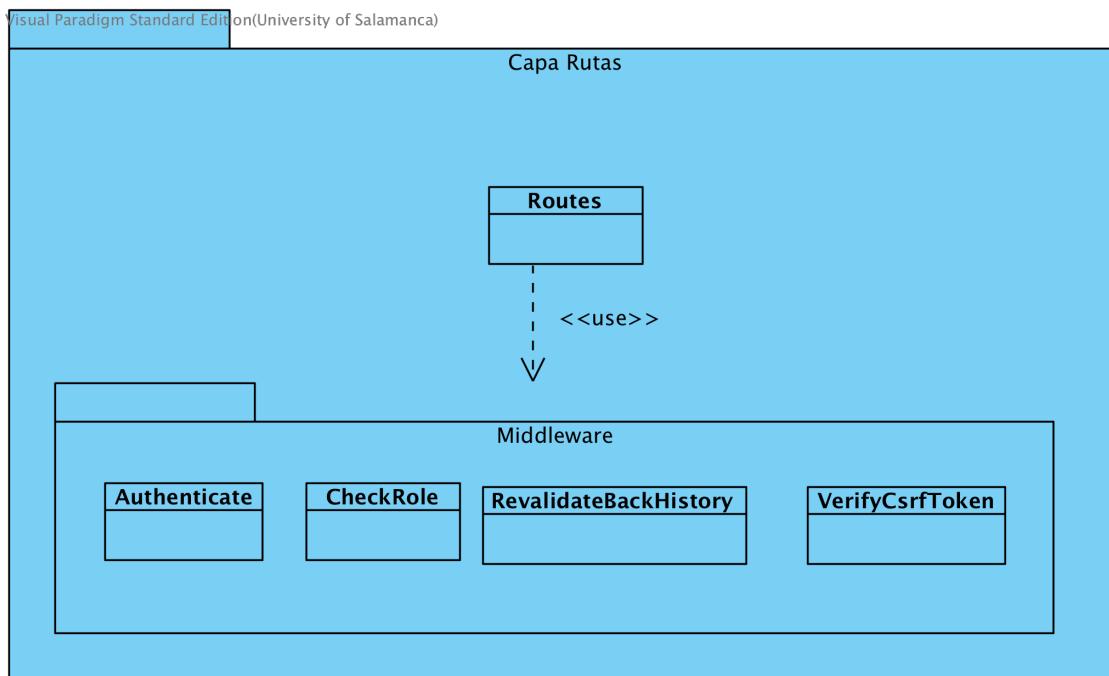
Se puede acceder a la aplicación en la URL <http://prodiasv04.fis.usal.es>. Se puede utilizar el registro, sin embargo, para la asignación de los roles y las responsabilidades es necesario el uso del SuperAdministrador y sin esto el sistema no tiene mucha funcionalidad. Por lo tanto, se facilitan en la siguiente tabla varios usuarios para poder hacer uso de la aplicación y comprobar su funcionamiento:

Usuario/E-mail	Contraseña
ganadero@tfg.com	ganadero
laboratorio@tfg.com	laboratorio
asociacion@tfg.com	asociacion
superadmin@tfg.com	superadmin

## 5.2. MVC en Laravel

### 5.2.1. Rutas

Como podemos ver en la Ilustración 9 , el MVC de Laravel, además de aplicar el típico patrón MVC, lo adapta añadiendo Rutas con el objetivo de hacer el código aún más claro. Éstas rutas serán las que se accederán desde el navegador para hacer invocaciones a los controladores.



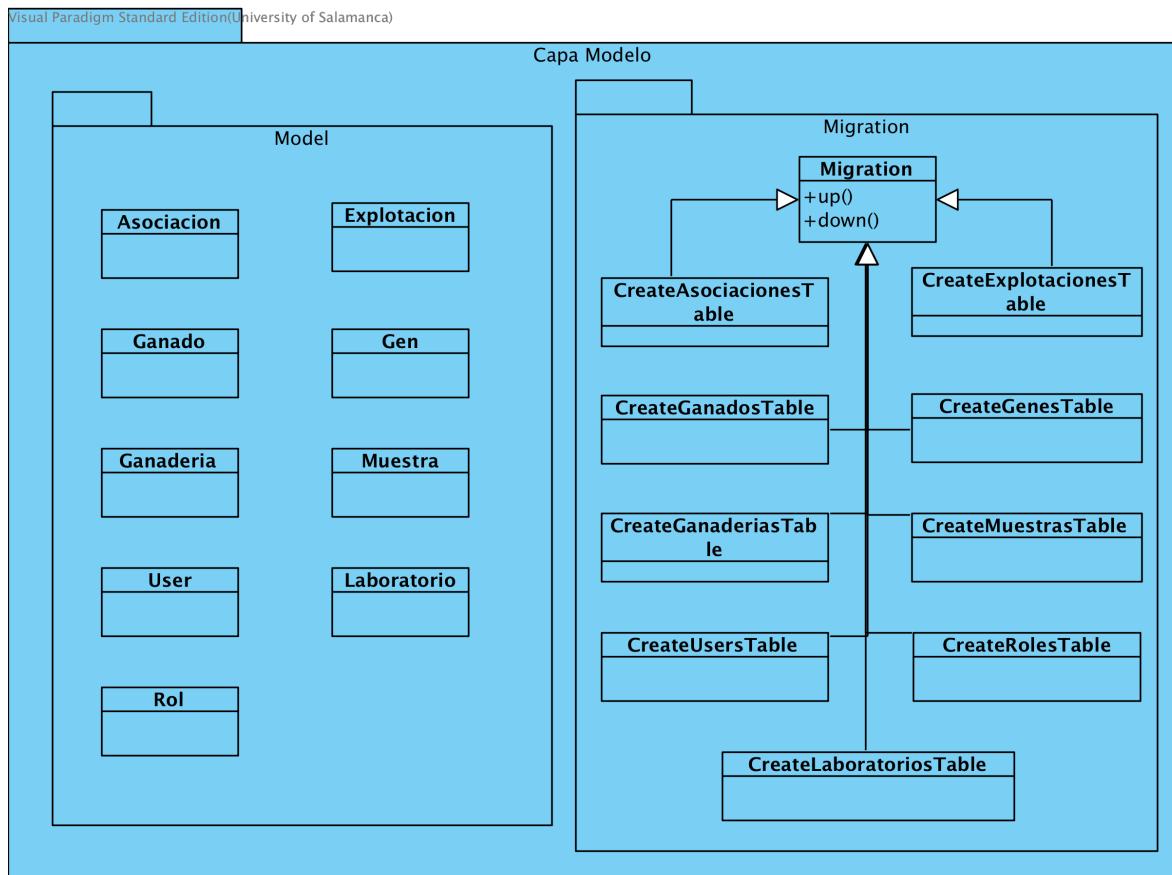
**Ilustración 38 Capa de Rutas de la aplicación**

Las rutas definen los tipos de petición que van atender, los posibles parámetros que pueden contener, el middleware a usar y la llamada a los controladores que van a hacer.

La utilización de rutas hace que el desarrollo de la aplicación se asemeje bastante al desarrollo orientado a servicios y que se nos presenten muchas facilidades a la hora de crear una API.

### 5.2.2. Modelo

El modelo es la capa donde se encuentra la lógica de negocio, la información que se debe de almacenar en las bases de datos y es independiente de la vista y el controlador.



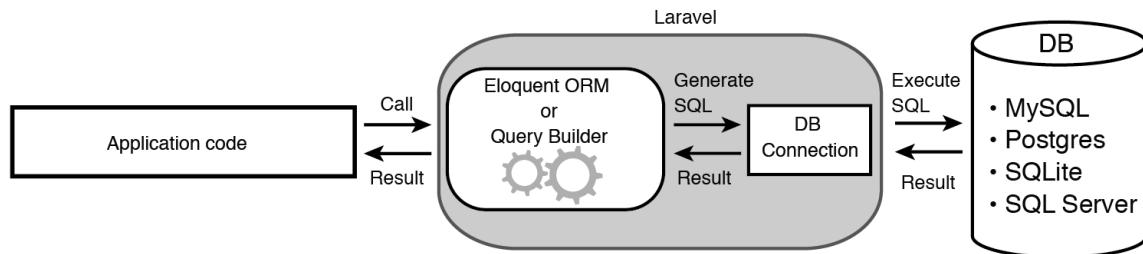
**Ilustración 39 Capa Modelo del sistema desarrollado**

El modelo de Laravel no difiere mucho de otro cualquiera en el paradigma de la orientación a objetos, sin embargo, es importante señalar que para el almacenamiento persistente de datos y las relaciones entre diferentes entidades (en Laravel llamadas “Model”) del modelo está fuertemente relacionado con el mapeo de datos relacional del **ORM Eloquent**. Para crear un modelo en Laravel necesitaremos crear una “migración” de esta forma el motor de Eloquent será capaz de interpretar las acciones que se están realizando sobre el Modelo y traducirlas a la base de datos persistente.

Por poner un ejemplo, creamos el Modelo “Asociacion”, podemos crear atributos para él como si de una clase PHP normal se tratase, definir los getter, setter... Finalmente habremos creado un Modelo totalmente funcional de forma local, pudiendo crear nuevas instancias u objetos de él. Sin embargo, éste no será persistente.

Con este propósito se creará una Migración, ésta consiste básicamente en la expresión del modelo en una sintaxis similar a SQL de forma que el ORM lo entienda y pueda ligar los atributos del Modelo a la Migración y de esta forma hacerlos persistentes en la base de datos que estemos utilizando.

Además de los atributos de las clases para crear cada Modelo se pueden utilizar funciones para relacionar un Modelo con otro y que **Eloquent** las plasme sobre la base de datos, de esta forma podremos por ejemplo crear una relación “*Asociacion hasMany Ganaderia*” que para acceder de forma inversa deberá estar declarada como “*Ganaderia belongsTo Asociacion*”.



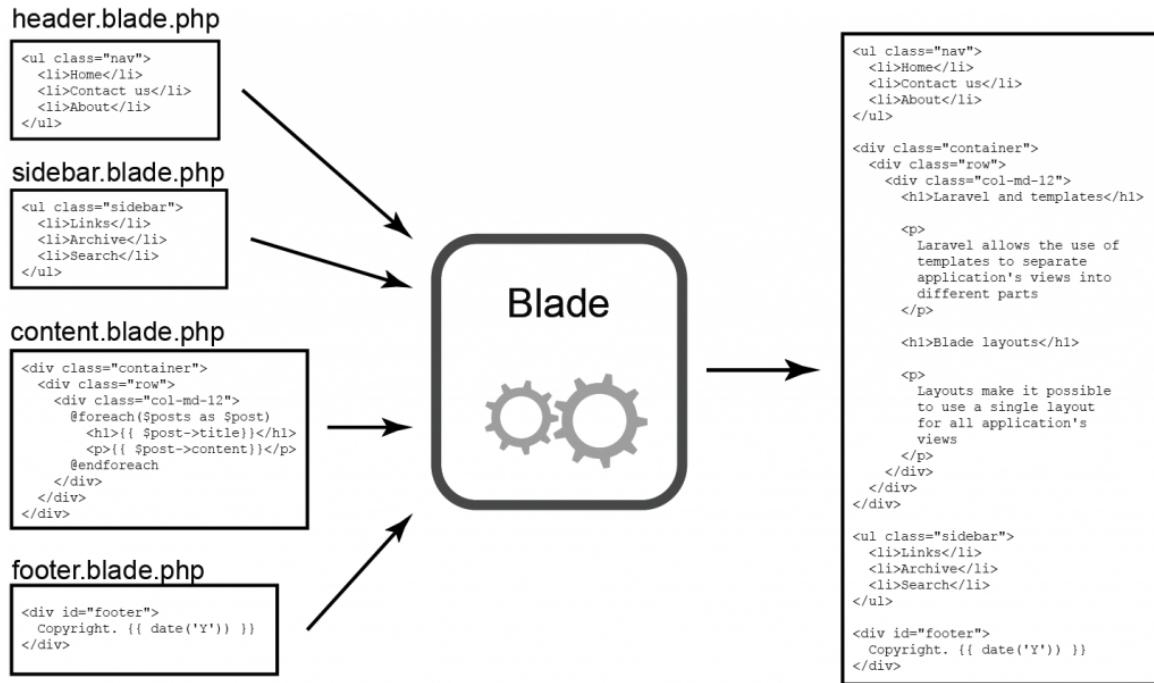
#### Ilustración 40 Funcionamiento de Eloquent con almacenamiento persistente

Además, para facilitar el uso general del sistema y empezar a tener información guardada en nuestro Modelo, Laravel nos proporciona dos herramientas que combinadas resultan muy potentes:

Los **seeders** se encargan de llenar el Modelo mediante el uso del componente **Faker** que nos aporta valores “aleatorios” pero orientados a la vez al tipo de dato que esperamos tener (un teléfono, una dirección de correo electrónico, un apellido...). Desde los **seeders** es posible además definir las relaciones entre los diferentes Modelos también de forma aleatoria.

#### 5.2.3. Vista

Laravel utiliza el procesamiento de plantillas **Blade** lo cual favorece un código mucho más limpio en las vistas.



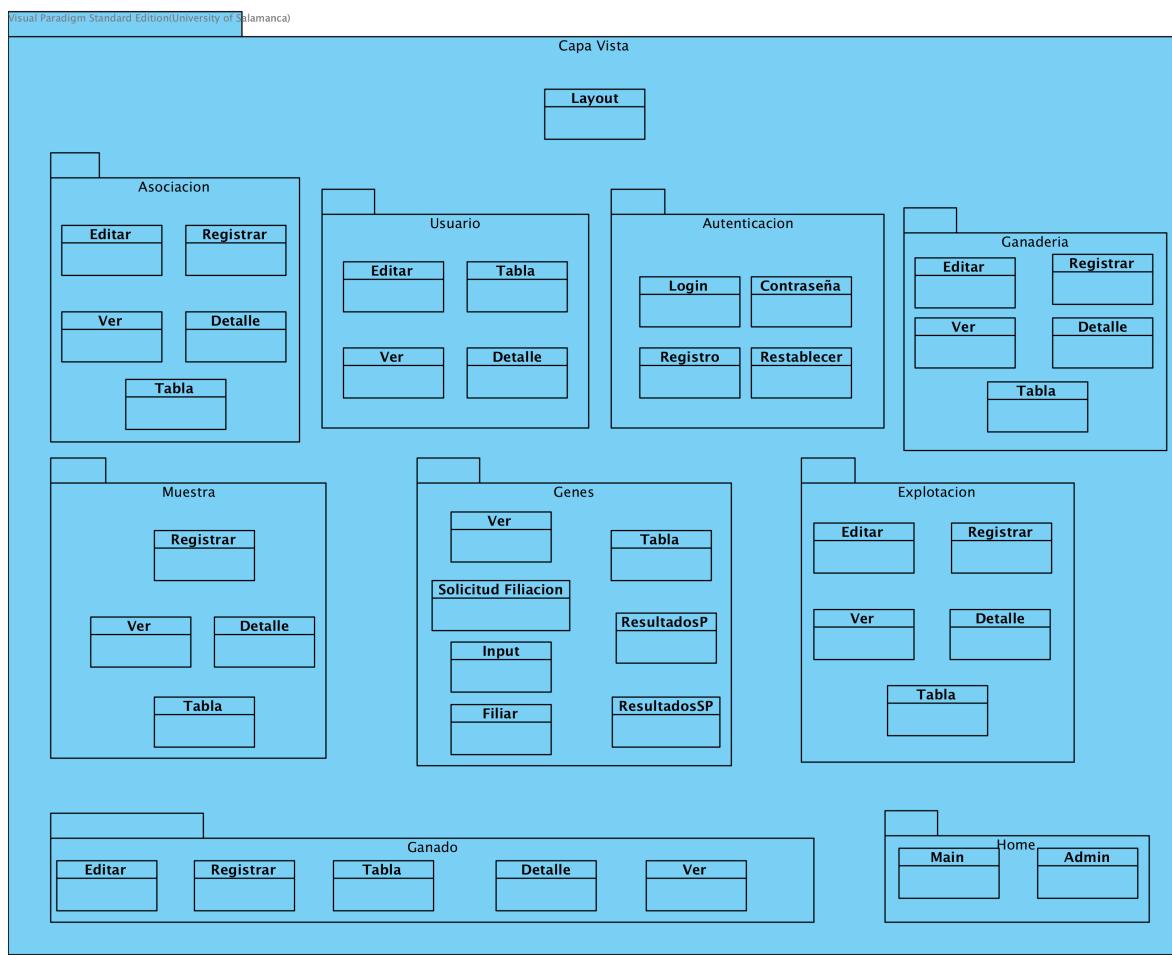
**Ilustración 41 Estructura de las vistas en Laravel Blade**

Las plantillas en **Blade** son archivos de texto plano que contiene todo el HTML de la página con etiquetas que representan elementos o zonas a incluir en la plantilla, o vistas parciales como se conocen en otros frameworks PHP. Sin embargo, en Blade estos elementos incrustados se organizan en un sólo archivo. Esta es una idea muy interesante de Laravel que mejora la organización de las vistas y su rendimiento. Sobre todo cuando las vistas pueden llegar a ser muy complejas incluso con elementos anidados. En el render de una Vista completa en Laravel se usan dos archivos: la plantilla definiendo el HTML global y las zonas a incluir.

De esta forma diseñamos un “layout” principal con “@yield” que contenga los elementos comunes a todas las páginas, como en el caso de nuestra web, la “navbar” y el “footer” y después podemos anidar otras vistas mediante “@extends”, “@include”, “@section”.

En el desarrollo del proyecto hemos organizado las vistas por paquetes de forma que cada *Modelo* tiene un paquete de vistas con el que interactúa mediante su controlador.

## Memoria



**Ilustración 42 Contenido de la Capa Vista**

Como hemos comentado anteriormente, cada una de las vistas parciales de cada paquete se integran dentro de una vista principal llamada Layout.

### 5.2.4. Controlador

Atiende a las peticiones del usuario a través de las rutas (a su vez llamadas por las vistas). Es la forma de interactuar del usuario con la vista.

El controlador está dividido en dos partes:

- **Controlador del lado del cliente:** utilizado para cambios en la interfaz y peticiones asíncronas.
- **Controlador del lado del servidor:** se encarga de atender las peticiones del lado del cliente y trasladarlas al modelo.

La capa en el cliente está organizada en clases o funciones ya que JavaScript considera las funciones como clases y viceversa. JavaScript actúa como una interfaz entre los controladores del servidor y la interfaz que ve el usuario, enviando las peticiones y también actualizando la vista dependiendo de ciertos eventos.

En el servidor, tenemos un controlador por cada una de las clases de nuestro modelo (Asociación, Ganadería, Explotación...) a lo que añadimos también un controlador para las importaciones y exportaciones, y otro para gestionar las vistas de inicio y "landing". También existen controladores para la autenticación y la recuperación de contraseñas. Todos estos controladores heredan de la clase de Laravel "Controller".

Además, para el uso de ciertos controladores, se interpone un middleware. El middleware se utiliza como filtro en los controladores y rutas, de forma que por ejemplo establecemos filtros de autenticación, de validación de formularios o de roles.

Podemos consultar un diagrama de la capa Controlador de nuestra aplicación en la Ilustración 34 .

### 5.3. Implementación del algoritmo de análisis genético.

Para el desarrollo del algoritmo se investigó primero sobre la realización de filiaciones paternales. Con este propósito la tutora Elena Solera facilitó artículos en los que se explicaba la forma de trabajar en este tipo de tareas de forma teórica [18]. También nos proporcionó documentos de Excel con ejemplos de filiaciones que se habían realizado por ella en el pasado.

Tras la lectura de éstos se empezó a trabajar con el algoritmo de filiación de Evett y Weir. Inicialmente se observaron los ejemplos realizados por Elena para hacernos a la idea de cómo funcionaba:

Ilustración 43 Ejemplo de una filiación realizada por Innovagenomics

Después de esto se empezó a trabajar a mano realizando las mismas filiaciones de los ejemplos que nos habían proporcionado y comparando los resultados. Para ello, se utilizó un documento proporcionado por la tutora Elena Solera donde se encontraban las formulas desglosadas por el tipo de filiación a realizar [2]:

GenotipoC	GenotipoM	GenotipoAF	PI (1)	PI (2)	PI (3)
AiAi	AiAi	AiAi	1/pi	$1+3\theta /4\theta +(1-\theta )pi$	$1/pi(1-2\theta AT)+2\theta AT$
		AiAj	1/2pi	$1+3\theta /2[3\theta +(1-\theta )pi]$	$1/2pi(1-2\theta AT)+2\theta AT$
		AjAk	0	0	0
AiAj	AiAi	AiAi	1/pi	$1+3\theta /3\theta +(1-\theta )pi$	$1/pi(1-2\theta AT)+2\theta AT$
		AiAj	1/2pi	$1+3\theta /2[2\theta +(1-\theta )pi]$	$1/2pi(1-2\theta AT)+2\theta AT$
		AjAk	0	0	0
AiAj	AjAj	AjAj	1/pj	$1+3\theta /2\theta +(1-\theta )pj$	$1/pj(1-2\theta AT)+2\theta AT$
		AjAj	1/2pj	$1+3\theta /2[\theta +(1-\theta )pj]$	$1/2pj(1-2\theta AT)+2\theta AT$
		AjAk	1/2pj	$1+3\theta /2[\theta +(1-\theta )pj]$	$1/2pj(1-2\theta AT)+2\theta AT$
		AkAl	0	0	0
AiAj	AiAi	AiAi	$1/(pi+pj)$	$1+3\theta /4\theta +(1-\theta )(pi+pj)$	$1/(pi+pj)(1-2\theta AT)+2\theta AT$
		AiAj	$1/(pi+pj)$	$1+3\theta /4\theta +(1-\theta )(pi+pj)$	$1/(pi+pj)(1-2\theta AT)+2\theta AT$
		AjAk	$1/2(pi+pj)$	$1+3\theta /2[3\theta +(1-\theta )(pi+pj)]$	$1/2(pi+pj)(1-2\theta AT)+2\theta AT$
		AkAl	0	0	0
AiAk	AjAj	AjAj	1/pj	$1+3\theta /2\theta +(1-\theta )pj$	$1/pj(1-2\theta AT)+2\theta AT$
		AjAj	1/2pj	$1+3\theta /2[\theta +(1-\theta )pj]$	$1/2pj(1-2\theta AT)+2\theta AT$
		AjAl	1/2pj	$1+3\theta /2[\theta +(1-\theta )pj]$	$1/2pj(1-2\theta AT)+2\theta AT$
		AkAl	0	0	0

**Ilustración 44 Tabla de fórmulas utilizadas para la filiación parental**

Como vemos en la tabla tenemos dos tipos de datos el **valor de los alelos** (por ejemplo Ai) que es un **numero entero** y la **probabilidad** o coeficiente que tiene un **alelo** en nuestra base de datos (pi o pj).

Cuando se termina de comprender el proceso de filiación se empieza a implementar el algoritmo en nuestro sistema.

Primero se decide crear una función que nos devuelva la probabilidad de el valor X de un alelo dado un determinado marcador, por ejemplo, el valor 160 en un alelo del marcador TGLA53. Para ello se recorre la base de datos de genes y se va acumulando las veces que aparece en una variable, finalmente se calcula el coeficiente de probabilidad que tiene. Según lo estudiado sirve para que a la hora de aplicar el algoritmo se sepa si la aparición de un determinado valor es excepcional o no y de esta

forma asignarle un coeficiente. Una vez hecho esto se puede comenzar a implementar el propio algoritmo de filiación.

Este se consigue mediante la acumulación de condiciones y llamadas a la función anteriormente explicada. Primero comprobamos los dos **alelos del hijo**, para saber si es homocigótico (ambos alelos iguales) o heterocigótico (ambos alelos diferentes), esto se correspondería con las comprobaciones de la primera columna de la tabla. Después pasamos a comprobar los dos **alelos de la madre**, existen varias posibilidades que debemos de filtrar:

Hijo homocigótico:

- Alelos homocigóticos al igual que el hijo.
- Alelos heterocigóticos, al menos uno coincide con los del hijo.

Hijo heterocigótico:

- Alelos homocigóticos y al menos uno coincide con los del hijo.
- Alelos heterocigóticos idénticos a los del hijo.
- Alelos heterocigóticos y uno coincide con los del hijo

Finalmente, pasamos a las comprobaciones sobre el **padre** que dependerán de las anteriores y de los propios valores de los alelos del padre (tercera columna de la tabla). Determinarán finalmente el alelo que se escoge para determinar su frecuencia en la base de datos con la función anteriormente explicada y dividirla por el factor que coincide con las condiciones comprobadas en la columna PI(1).

Al poco tiempo empezamos a tener una versión que ya nos aporta los resultados numéricos.

```
array:3 [▼
  0 => array:12 [▼
    0 => 4.3512195121951
    1 => 5.3095238095238
    2 => 5.8684210526316
    3 => 2.0792540792541
    4 => 13.313432835821
    5 => 2.5485714285714
    6 => 2.133971291866
    7 => 1.6218181818182
    8 => 2.0318906605923
    9 => 0.69362363919129
    10 => 2.646884272997
    11 => 3.7478991596639
  ]
  1 => 462828.46854597
  2 => 0.99999783937699
]
```

#### Ilustración 45 Primeros resultados efectivos de la aplicación del algoritmo.

En la Ilustración 45 podemos los resultados obtenidos al aplicar el algoritmo, el primer array (el elemento 0) son los índices de paternidad de cada uno de los marcadores y los elementos primero y segundo son el índice de paternidad total y el índice de paternidad expresado en coeficiente o porcentaje, respectivamente.

## Memoria

Además, se construyó una interfaz alrededor del algoritmo de filiación para que tanto la solicitud de la filiación como los propios resultados fueran lo más familiares para el usuario del laboratorio al que está destinado su uso como se puede ver en las siguientes ilustraciones.

Solicitar Filiación



819JANSBDHE93

Editar Perfil Genético



BNA8g1LAKS910

Editar Perfil Genético



PHK61829239HK

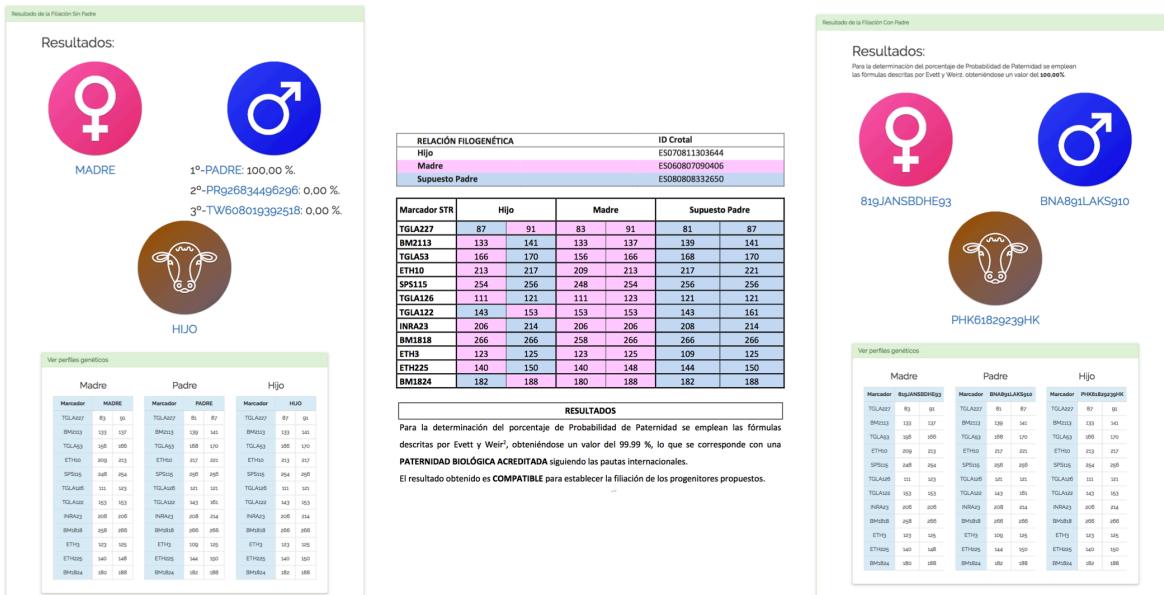
Editar Perfil Genético

Ver perfiles genéticos

Madre			Padre			Hijo		
Marcador	819JANSBDHE93		Marcador	BNA8g1LAKS910		Marcador	PHK61829239HK	
TGLA227	83	91	TGLA227	81	87	TGLA227	87	91
BM2113	133	137	BM2113	139	141	BM2113	133	141
TGLA53	156	166	TGLA53	168	170	TGLA53	166	170
ETH10	209	213	ETH10	217	221	ETH10	213	217
SPS115	248	254	SPS115	256	256	SPS115	254	256
TGLA126	111	123	TGLA126	121	121	TGLA126	111	121
TGLA122	153	153	TGLA122	143	161	TGLA122	143	153
INRA23	206	206	INRA23	208	214	INRA23	206	214
BM1818	258	266	BM1818	266	266	BM1818	266	266
ETH3	123	125	ETH3	109	125	ETH3	123	125
ETH225	140	148	ETH225	144	150	ETH225	140	150
BM1824	180	188	BM1824	182	188	BM1824	182	188

Filiación con padre Filiación sin padre

Ilustración 46 Interfaz para la solicitud de una filiación.



**Ilustración 47 Resultados de una filiación sin padre conocido, resultados según los presenta Innovagenomics y resultados de una filiación con padre conocido.**



## 6. Conclusión

Tras haber finalizado el proyecto es necesario realizar un análisis y una valoración del largo proceso seguido para sacar las conclusiones que se obtienen durante la realización del mismo.

Primeramente, se debe señalar que se han alcanzado los objetivos y requisitos que se marcaron al principio del proyecto. Se ha construido una herramienta que permite la gestión de una asociación ganadera y todas las entidades que dependen de ésta. Además, se ha ampliado el sistema mediante la introducción del módulo genético en el que se incluye la funcionalidad de la filiación genética. Dicho módulo no estaba contemplado en primera instancia.

Durante el período de desarrollo del proyecto, no solo se han aplicado los conocimientos adquiridos a lo largo de los años de carrera, sino que también se han ampliado y orientado a procesos de desarrollo de mayor envergadura. En este punto es importante remarcar el trabajo directo con empresas en calidad de clientes y la gestión de todo este proceso.

Se ha construido una aplicación que es portable e interoperable. Es decir, esta aplicación puede ser instalada en cualquier servidor Web que soporte PHP y Laravel. Además, esto facilita su uso a los usuarios finales ya que no requiere de instalaciones para utilizarla y al programador le libera de la gestión de los recursos computacionales.

La aplicación hace un uso importante de las últimas tecnologías disponibles empleando las tecnologías HTML5, CSS3, JavaScript y PHP que son consideradas como las más novedosas y respaldadas lo que facilitará en el futuro el desarrollo y mantenimiento de la aplicación. Además el uso de éstas tecnologías se ha desarrollado rigiéndose por los estándares [35] y [17]. Se ha utilizado una filosofía orientada a objetos y los principios de alta cohesión y bajo acoplamiento, lo que da como resultado unos componentes de software que son altamente reutilizables.

También se quiere remarcar que, aunque la aplicación se ha desarrollado en colaboración con la empresa Innovagenomics y gran parte de la funcionalidad está basada en sus necesidades, se ha creado una aplicación lo más estándar posible para que pueda ser adoptada por otras empresas que tengan necesidades parecidas. Además, se ha garantizado que la transición desde otras aplicaciones o herramientas sea lo más fácil posible. Para esto, se ha diseñado una interfaz que está centrada en el usuario y que responde a sus necesidades.

Respecto al módulo genético señalar que es de gran utilidad para los laboratorios, tanto por el fácil acceso a los datos como y su organización que son mucho mejores que la actual como por el cálculo automático de las propias filiaciones, proceso que normalmente ocupa un gran espacio de tiempo a los laboratorios. En definitiva, la aplicación es útil tanto para las asociaciones como los laboratorios.

Gracias a las herramientas que aporta Laravel (CSRF Validation) se ha garantizado la robustez de la aplicación en los formularios a la hora de validar los campos. También se ha asegurado el

## Memoria

funcionamiento de los permisos y las responsabilidades, de tal forma que no se pueda acceder a información ni utilizar funcionalidades sobre las que el usuario no tiene permisos.

En el aspecto personal, se han cumplido con creces los objetivos que se citaban al comienzo de esta memoria. Se han afianzado y aplicado la totalidad de los conocimientos que se han adquirido durante los cuatro últimos años. También se han adquirido unos conocimientos bastante notables en programación de aplicaciones Web dinámicas mediante el uso de Laravel y la combinación de este con JavaScript. Se considera que aún más importante el aprendizaje transversal gracias al cual se ha podido adquirir conocimientos en campos tan variados como los de la genética, el funcionamiento de las empresas del sector ganadero o el diseño de iconos para aplicaciones.

## 7. Líneas de trabajo futuro

Tras el desarrollo completo de la herramienta descrita, se proponen algunas ampliaciones que sería interesante realizar:

- **Actualización de los recursos técnicos:** tras terminar el desarrollo de la aplicación han surgido nuevas versiones de las tecnologías usadas que introducen cambios importantes. Estas son:
  - **Laravel 5.5 (LTS):** Es conveniente mudar el sistema a la siguiente versión de soporte a largo plazo de Laravel que está programada para lanzarse en julio o agosto de 2017 [19]. Dará soporte a la corrección de fallos durante dos años y también tres años de actualizaciones para la seguridad.
  - **Bootstrap 4:** Que actualmente se encuentra ya en las últimas versiones pre-release.
- **Creación de una API pública:** de forma que los desarrolladores de otras aplicaciones puedan embeber, portar o interactuar con nuestro sistema. Se pueden crear servicios web con las acciones que realizan los controladores para que de esta forma los terceros puedan utilizar nuestro sistema.
- Mejoras e introducción de nuevas funciones en módulo genético:
  - **Introducción de nuevos algoritmos:** Para realizar cálculos dados los datos genéticos sobre diferentes valores como puede ser por ejemplo la calidad de la leche.
  - **Mejora de los algoritmos de filiación:** Aun que actualmente funcionan perfectamente, se pueden seguir introduciendo mejoras y nueva funcionalidad a la hora de mostrar los resultados.
- Sistema de notificaciones:
  - **Correo:** al realizar cambios, enviar muestras o realizar filiaciones que se introduzcan funciones para notificar a los involucrados mediante un correo electrónico.
  - **Web:** Mismo propósito que las notificaciones por correo pero que se ejecuten mediante el navegador web como ya se viene viendo últimamente en nuevas aplicaciones como por ejemplo Facebook.
  - **Gestión y configuración:** Poder gestionar los ajustes de estas notificaciones.
- **Mejoras en seguridad, fiabilidad y estabilidad:** Aun que el sistema ya cumple estos requisitos es importante mantenerlo actualizado respecto a estas tres vertientes.



## 8. Referencias

- [1] Agritecsoft.com. (2017). *Vaquitec. Software avanzado para granjas vacuno carne y leche.* [online] Available at: <http://www.agritecsoft.com/vaquitec/es/> [Accessed 22 Jun. 2017].
- [2] *Calculos de Paternidad.* [ebook] Available at: [http://www.gepisfg.org/archivos/201301/Paternidad%20\(2\).pdf](http://www.gepisfg.org/archivos/201301/Paternidad%20(2).pdf) [Accessed 12 May 2017].
- [3] Contributors, phpMyAdmin. (2017). *phpMyAdmin.* [online] phpMyAdmin. Available at: <https://www.phpmyadmin.net/> [Accessed 22 Jun. 2017].
- [4] Cooper, G. and Hausman, R. (2008). *La célula.* Madrid: Marbán.
- [5] Corchado, J. (2017). *El sector primario apuesta por la investigación de la Universidad de Salamanca.* [online] Juan M. Corchado. Available at: <https://corchado.net/2017/04/19/el-sector-primario-apuesta-por-la-investigacion-de-la-universidad-de-salamanca/> [Accessed 20 Jun. 2017].
- [6] Durán Toro, A. (2017). *Página de descarga de programa.* [online] Lsi.us.es. Available at: [http://www.lsi.us.es/descargas/descarga\\_programas.php?id=3](http://www.lsi.us.es/descargas/descarga_programas.php?id=3) [Accessed 25 Jun. 2017].
- [7] Durán y Bernárdez. (2002). Metodología de Elicitación de Requisitos.
- [8] Ehsan, M. (2017). *Introduction to Laravel Framework - Laravel Book.* [online] Laravelbook.com. Available at: <http://laravelbook.com/laravel-introduction/> [Accessed 22 Jun. 2017].
- [9] Es.wikipedia.org. (2017). *Ácido desoxirribonucleico.* [online] Available at: [https://es.wikipedia.org/wiki/%C3%81cido\\_desoxirribonucleico?oldformat=true#/media/File:Chromosome-es.svg](https://es.wikipedia.org/wiki/%C3%81cido_desoxirribonucleico?oldformat=true#/media/File:Chromosome-es.svg) [Accessed 22 Jun. 2017].
- [10]Es.wikipedia.org. (2017). *Gen.* [online] Available at: <https://es.wikipedia.org/wiki/Gen> [Accessed 22 Jun. 2017].
- [11]Evett, I. and Weir, B. (2017). *Interpreting DNA Evidence. Statistical Genetics for Forensic Scientists.* [ebook] The Forensic Science Service United Kingdom & Department of Statistics North Carolina State University, Chapter 6. Available at: [http://www.bio.bg.ac.rs/materijali\\_predmeta/forensic-dna-statistic-evett-i-weir.pdf](http://www.bio.bg.ac.rs/materijali_predmeta/forensic-dna-statistic-evett-i-weir.pdf) [Accessed 12 May 2017].
- [12]Genome.gov. (2017). *Glosario Hablado de Términos Genético.* [online] Available at: <https://www.genome.gov/glossarys/index.cfm?id=4> [Accessed 19 Jun. 2017].
- [13]Git-scm.com. (2017). *About - Git.* [online] Available at: <https://git-scm.com/about> [Accessed 22 Jun. 2017].

- [14]Group, D. (2017). *Welcome! - The Apache HTTP Server Project*. [online] Httpd.apache.org. Available at: <https://httpd.apache.org/> [Accessed 22 Jun. 2017].
- [15]Hurle, B. (2017). *Glosario de Términos Genéticos*. [online] Genome.gov. Available at: <https://www.genome.gov/glossarys/index.cfm?id=86> [Accessed 22 Jun. 2017].
- [16]Informática ganadera. (2017). *Software ganadero*. [online] Available at: <https://www.ganaderosonline.com/vacuno-de-carne> [Accessed 22 Jun. 2017].
- [17]Jigsaw.w3.org. (2017). *Resultados del Validador CSS del W3C para http://prodiasv04.fis.usal.es/css/stylesheet.css (CSS versión 3)*. [online] Available at: <https://jigsaw.w3.org/css-validator/validator?uri=http%3A%2F%2Fprodiasv04.fis.usal.es%2Fcss%2Fstylesheet.css&profile=css3&usermedium=all&warning=1&vextwarning=&lang=es> [Accessed 20 Jun. 2017].
- [18]Lagos L., M., Poggi M., H. and Mellado S., C. (2017). *Conceptos básicos sobre el estudio de paternidad*. [ebook] Santiago, Chile: Pontificia Universidad Católica de Chile. Available at: <http://www.scielo.cl/pdf/rmc/v139n4/art19.pdf> [Accessed 20 Jun. 2017].
- [19]Laravel News. (2017). *Laravel 5.5 Will Be The Next LTS Release*. [online] Available at: <https://laravel-news.com/laravel-5-5-lts> [Accessed 22 Jun. 2017].
- [20]León, J. (2017). El sector primario y agroalimentario recibirá 1.404 millones de euros de ayudas públicas | Agricultura y Ganadería | Junta de Castilla y León. [online] Jcyl.es. Available at: [http://www.jcyl.es/web/jcyl/AgriculturaGanaderia/es/Plantilla100Detalle/1246464862173/\\_/1284291145915/Comunicacion?plantillaObligatoria=PlantillaContentoNoticiaHome](http://www.jcyl.es/web/jcyl/AgriculturaGanaderia/es/Plantilla100Detalle/1246464862173/_/1284291145915/Comunicacion?plantillaObligatoria=PlantillaContentoNoticiaHome) [Accessed 21 Jun. 2017].
- [21]Mapama.gob.es. (2017). *Departamento de gestión de recursos zoogenéticos Laboratorios de Sanidad y Genética animal - Ganadería - mapama.es*. [online] Available at: [http://www.mapama.gob.es/es/ganaderia/temas/laboratorios/recursos\\_zoogeneticos.aspx](http://www.mapama.gob.es/es/ganaderia/temas/laboratorios/recursos_zoogeneticos.aspx) [Accessed 19 Jun. 2017].
- [22]Moreno García, M. (2016). *Apuntes de Ingeniería del Software II, Tema 2*.
- [23]Moreno García, M. (2016). *Apuntes de Ingeniería del Software II, Tema 3*.
- [24]Mozilla Developer Network. (2017). *AJAX*. [online] Available at: <https://developer.mozilla.org/en-US/docs/Glossary/AJAX> [Accessed 22 Jun. 2017].
- [25]Mozilla Developer Network. (2017). *CSS*. [online] Available at: <https://developer.mozilla.org/es/docs/Web/CSS> [Accessed 22 Jun.

- [26]Mozilla Developer Network. (2017). *HTML*. [online] Available at: <https://developer.mozilla.org/en-US/docs/Glossary/HTML> [Accessed 22 Jun. 2017].
- [27]Mozilla Developer Network. (2017). *JavaScript*. [online] Available at: <https://developer.mozilla.org/es/docs/Web/JavaScript> [Accessed 22 Jun. 2017].
- [28]Mozilla Developer Network. (2017). *jQuery*. [online] Available at: <https://developer.mozilla.org/en-US/docs/Glossary/jQuery> [Accessed 22 Jun. 2017].
- [29]office.com. (2017). *Software de administración de proyectos / Microsoft Project*. [online] Available at: <https://products.office.com/es-es/project/project-and-portfolio-management-software?tab=tabs-1> [Accessed 20 Jun. 2017].
- [30]Php.net. (2017). *PHP: ¿Qué es PHP? - Manual*. [online] Available at: <http://php.net/manual/es/intro-whatis.php> [Accessed 22 Jun. 2017].
- [31]Phpd.org. (2017). *phpDocumentor*. [online] Available at: <https://www.phpdoc.org/about> [Accessed 22 Jun. 2017].
- [32]Products.office.com. (2017). *Software de hojas de cálculo Microsoft Excel 2016*. [online] Available at: <https://products.office.com/es-es/excel> [Accessed 22 Jun. 2017].
- [33]Ramon, E. (2017). *Productos con Denominación de Origen*. [online] Available at: <http://www.versalamanca.com/denominacion-origen-salamanca.html> [Accessed 22 Jun. 2017].
- [34]Techterms.com. (2017). *MySQL Definition*. [online] Available at: <https://techterms.com/definition/mysql> [Accessed 22 Jun. 2017].
- [35]Validator.w3.org. (2017). *Showing results for http://prodiasv04.fis.usal.es/-Nu Html Checker*. [online] Available at: <https://validator.w3.org/nu/?doc=http%3A%2F%2Fprodiasv04.fis.usal.es%2F> [Accessed 18 Jun. 2017].