



**Universitat Autònoma
de Barcelona**

**Desarrollo de una plataforma
bioinformática para la gestión de
datos fenotípicos y moleculares**

Memoria del proyecto
de Ingeniería Técnica en
Informática de Gestión

realizado por

Adrián Forte Romera

y dirigido por

Gonzalo Vera Rodríguez

Escola d'Enginyeria

Sabadell, Junio de 2010

El abajo firmante, **Gonzalo Vera Rodríguez**, profesor de la Escuela de Ingeniería de la UAB,

CERTIFICA:

Que el trabajo al que corresponde la presente memoria ha estado realizado bajo su dirección por **Adrián Forte Romera**.

Y para que conste firma la presente.

Sabadell, Junio del 2010

Firmado: Gonzalo Vera Rodríguez.

Resumen del proyecto

Este proyecto está orientado a ayudar a un grupo de investigadores del Departamento de Ciencia Animal y de los Alimentos (UAB), que se dedican a recopilar datos genómicos obtenidos en experimentos para su posterior análisis en sus investigaciones. La mala gestión de estos datos ocasiona que exista duplicidad y pérdida de información, ya que se trabaja sobre ficheros Excel y cada investigador posee su propia versión del archivo de resultados de los experimentos, con lo que esto puede conllevar a la repetición de experimentos con unos costes elevados. Este grupo de investigadores necesita una herramienta online con la que solucionar esta mala gestión por parte de algunos de los usuarios y que presente la información contenida en estos ficheros de una forma clara y concisa.

El objetivo es tener toda la información obtenida en los experimentos en un único lugar, asegurando la integridad de los datos y pudiendo mejorar la comunicación entre los investigadores.

Otro de los objetivos es minimizar el impacto que supondrá el paso de los datos en Excel a la aplicación, esto se conseguirá gracias a la opción de subir archivos. Estos archivos tendrán un formato especificado en las entrevistas utilizando las aportaciones de Miguel Pérez Enciso, miembro del Departamento de Ciencia Animal y de los Alimentos de la UAB.

La aplicación constará de dos partes bien diferenciadas. La primera es la parte de los usuarios, donde se podrá crear proyectos, insertar, modificar o eliminar datos, y por supuesto, se podrá consultar la información ya existente en la aplicación.

La segunda parte es la del administrador, que como tal podrá dar de alta a nuevos usuarios, restaurar versiones anteriores de la base de datos, eliminar usuarios y consultar las acciones realizadas por los usuarios.

Una vez se ha implementado la aplicación se minimizará la mala gestión de los datos y se mejorará la comunicación entre los investigadores, ya que todos tendrán la misma información y cualquier cambio podrá ser visto por todos.

Índice

1. Introducción

1.1 Motivación	1
1.2 Objetivos.....	2
1.3 Estado del arte.....	2
1.4 Estructura de la memoria.....	3

2. Estudio de viabilidad

2.1 Introducción.....	4
2.2 Objeto	4
2.2.1 Descripción de la situación actual.....	4
2.2.2 Perfil del usuario	5
2.2.3 Objetivos	5
2.2.4 Fuentes de información	5
2.3 Sistema a realizar	5
2.3.1 Descripción	5
2.3.2 Modelo de desarrollo.....	7
2.3.3 Planificación	7
2.3.4 Recursos de desarrollo	9
2.3.5 Recursos de implementación.....	9
2.3.6 Evaluación de riesgos	10
2.3.7 Alternativas.....	11
2.4 Análisis de coste-beneficio.....	12
2.5 Conclusiones	14

3. Análisis

3.1 Introducción.....	15
3.2 Requerimientos funcionales	15
3.3 Requerimientos no funcionales	16
3.4 Información proporcionada	17
3.5 Diagrama de casos de uso.....	19
3.5.1 Investigador.....	19
3.5.2 Administrador.....	20
3.6 Descripción de casos de uso	21
3.6.1 Investigador.....	21
3.6.2 Administrador.....	27
3.7 Diagrama de actividades	28
3.7.1 Investigador.....	28
3.7.2 Administrador.....	31

4. Marco tecnológico	
4.1 Introducción.....	32
4.2 Tecnologías web	32
4.3 Bases de datos	34
4.4 Apache.....	34
4.5 Notepad++.....	35
4.6 R.....	35
4.7 mkInstaller	35
5. Diseño de la aplicación	
5.1 Introducción.....	36
5.2 Modelo de datos.....	36
5.2.1 Diseño del modelo de datos.....	36
5.3 Interfaz gráfica	38
5.3.1 Estructura de la página	38
5.3.2 Menú de la aplicación	38
5.3.3 Zona de información	40
5.3.4 Botones.....	40
5.3.5 Breadcrumb	41
5.4 Base de datos	41
5.4.1 Diagrama.....	41
5.4.2 Estructura.....	45
5.5 Planificación.....	52
6. Implementación	
6.1 Implementación de la interfaz.....	54
6.1.1 Implementación de la estructura de la página.....	54
6.1.2 Menú de la aplicación	55
6.1.3 Zona de información	57
6.2 Estructura de la aplicación	57
6.3 Codificación de las clases.....	60
6.4 Ejemplos de codificación	63
7. Pruebas	
7.1 Pruebas realizadas por el alumno.....	65
7.1.1 Módulo de proyectos y Módulo importar archivos.....	65
7.1.2 Módulo insertar información.....	66
7.1.3 Módulo modificar información.....	67
7.1.4 Módulo consultas y Módulo de operaciones con datos	69
7.1.5 Módulo sesiones	75
7.1.6 Módulo administrador	76
7.2 Validación del usuario final	78
7.3 Valoración de los resultados de validación del usuario final	79

8. Conclusiones	
8.1 Consecución de objetivos	80
8.2 Planificación real.....	80
8.3 Líneas de ampliación	83
8.4 Valoración personal	83
9. Bibliografía	
9.1 Fuentes bibliográficas	84
9.2 Fuentes electrónicas	84
9.2.1 Fuentes para el desarrollo de la aplicación.....	84
9.2.2 Fuentes relacionadas con la temática de la aplicación.....	85
Anexo I: Creación de ficheros.....	86
Anexo II: Manual de usuario	94
Anexo III: Manual de instalación.....	99

1. Introducción

1.1 Motivación

El principal problema que presenta el grupo de investigadores del Departamento de Ciencia Animal y de los Alimentos (UAB) es la mala gestión de los datos obtenidos de los distintos experimentos realizados, esto conlleva la pérdida y duplicidad de la información.

Los datos que se generan en los experimentos y que se guardarán en la aplicación son fenotipos y genotipos [14 -19]. Una explicación simplificada de estos dos términos es que el fenotipo es cualquier característica o rasgo observable de un individuo, como el color del cabello o el peso, y el genotipo es toda la información contenida en los cromosomas, esto quiere decir que hay información que no es visible. Estos dos términos responden a la siguiente fórmula:

$$\text{Ambiente} + \text{Genotipo} = \text{Fenotipo}$$

En la Figura 1 se puede observar una aproximación de cómo los investigadores obtienen los datos de los distintos individuos.

Un grupo de investigadores realiza un experimento con la información de unos individuos

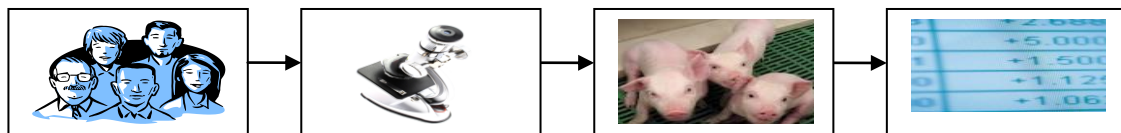


Figura 1: Esquema de la obtención de los datos por los investigadores.

El proyecto está orientado a realizar el diseño y creación de una base de datos que permita la inserción de información de los diferentes tipos de individuos, que al tratarse de un departamento de la facultad de veterinaria serán animales, en este caso serán cerdos, pero en previsión de una posible necesidad de introducir otro tipo de individuos, como podrían ser vegetales, se implementará la aplicación de manera que no sea necesario rediseñarla para asumir dichos cambios. También se desarrollará una interfaz web que facilite el trabajo que se realizará sobre la información contenida en esta base de datos, y que permitirá un visionado más cómodo a los usuarios.

Con el desarrollo de estos dos puntos se permitirá una mayor rapidez en la consulta de información además de permitir el acceso a los datos desde cualquier lugar, gracias a que se trata de una aplicación online.

1.2 Objetivos

El objetivo principal del proyecto es desarrollar una herramienta con la que se pueda insertar datos de los diferentes tipos de individuos, sin que esto ocasione grandes problemas o se tenga que rediseñar la aplicación en el caso de cambiar el tipo de espécimen de los experimentos.

Además tiene que diseñarse una interfaz que permita consultar, insertar, modificar o eliminar estos datos de una forma dinámica e intuitiva para el usuario. Gracias al desarrollo de esta interfaz también se tendrá la seguridad de que ningún usuario accederá a la información a la que no esté autorizado.

Otro de los objetivos que se persigue es el de poder realizar análisis estadísticos sencillos predefinidos, para así optimizar el tiempo de trabajo, y no tener que exportar la información y realizar los cálculos en programas externos, aunque esta posibilidad también será posible.

1.3 Estado del arte

Actualmente hay diferentes bases de datos o aplicaciones bioinformáticas con las que poder trabajar. Una de esas posibles opciones es Molgenis [20], una plataforma para crear prototipos de biosoftware. El problema de esta aplicación es que no está diseñada para el tipo de experimentos que realiza la unidad de genética, y además se tendrían demasiadas opciones, que no se utilizarían, y que abrumarían a los usuarios.

Otra de las posibles opciones sería BASE [21] (BioArray Software Environment), una plataforma para la gestión y el análisis de todas las áreas de experimentación de un microarray. Esta aplicación también se ha descartado ya que no cumple con todas las necesidades que se tienen y se requeriría más tiempo en adecuarla a las necesidades de la unidad de genética y mejora que en realizar una aplicación a medida.

Asimismo existe BioMart [22] que es un sistema orientado a la gestión de datos y permite trabajar con diferentes bases de datos. El inconveniente de esta aplicación es que posee demasiadas opciones que no serán utilizadas y no reúne las características necesarias para complacer a los usuarios.

Lo que se pretende en este proyecto es crear una aplicación sencilla e intuitiva que se ajuste a las necesidades de los usuarios finales. Así que después de analizar las diferentes alternativas existentes en el mercado se ha optado por crear una aplicación a medida que satisfaga las necesidades concretas de la unidad de genética y mejora.

1.4 Estructura de la memoria

La memoria se divide en 9 capítulos a través de los cuales se explicará cómo se ha desarrollado el proyecto.

En este primer capítulo se ha dado una pequeña introducción al proyecto, a los objetivos que se quieren alcanzar y a las alternativas existentes en el mercado.

En el segundo capítulo se presentará el estudio de viabilidad del proyecto. En él se mostrará la situación actual, la planificación y el coste de realizar el proyecto, además de los recursos necesarios para el desarrollo y puesta en marcha de la aplicación. También se evaluará si el proyecto es viable técnica, económica y temporalmente.

En el tercer capítulo se realizará un análisis en profundidad de los requisitos funcionales y no funcionales del proyecto. Además se verán los diferentes casos de uso que tendrá cada usuario de la aplicación.

En el cuarto capítulo se describirá el marco tecnológico para el desarrollo de la aplicación.

El quinto capítulo estará dedicado al diseño de la aplicación. Aquí se explicará el modelo de datos creado y el formato de la información proporcionada, además se mostrará el diseño de la base de datos, explicando todos los apartados que la componen. Por último también se evaluará la interfaz gráfica.

En el sexto capítulo se expondrá los detalles de la implementación, como son el estilo de codificación, la estructura de los archivos, y además se explicarán las clases creadas para este proyecto.

El séptimo capítulo será el de pruebas. Aquí se comprobará el buen funcionamiento de la aplicación realizando diversos ensayos y se tendrá una valoración del usuario final una vez finalizada la primera versión de la aplicación.

En el octavo capítulo se expondrán las conclusiones finales del desarrollo del proyecto.

En el noveno capítulo se encontrará la bibliografía consultada durante el diseño y desarrollo de la aplicación.

Finalmente, en los anexos se explicarán el formato que deben tener los ficheros que se quieran subir a la aplicación, así como un pequeño manual de uso y puesta en marcha del proyecto.

2. Estudio de viabilidad

2.1. Introducción

En este capítulo se analizará la situación actual y los requisitos del proyecto y a continuación se mostrarán los objetivos principales así como el tipo de usuario al que se destinará la aplicación. También se mostrará el modelo de desarrollo, así como los recursos necesarios para llevarlo a cabo, su planificación y los costes imputables a estos. Teniendo en cuenta todos los factores anteriores, se desarrollará la viabilidad del proyecto.

2.2. Objeto

2.2.1 Descripción de la situación actual

Actualmente los datos son almacenados en ficheros Excel que contienen la información de los experimentos. El principal problema es que cada investigador posee su fichero, y con esto surgen a la vez varios problemas debido a que la información no está unificada, como pueden ser la pérdida accidental de datos, la repetición de experimentos, etc. Estos problemas son los que se pretende solucionar con la realización de este proyecto. En la Figura 2 podemos ver un esquema de la situación actual. Como podemos observar el investigador recopila información y la almacena en ficheros Excel, y realiza análisis estadísticos en el lenguaje R.

Por lo tanto lo que se pretende es agrupar todos los datos en una sola base de datos que contenga una interfaz gráfica para facilitar la consulta o modificación.

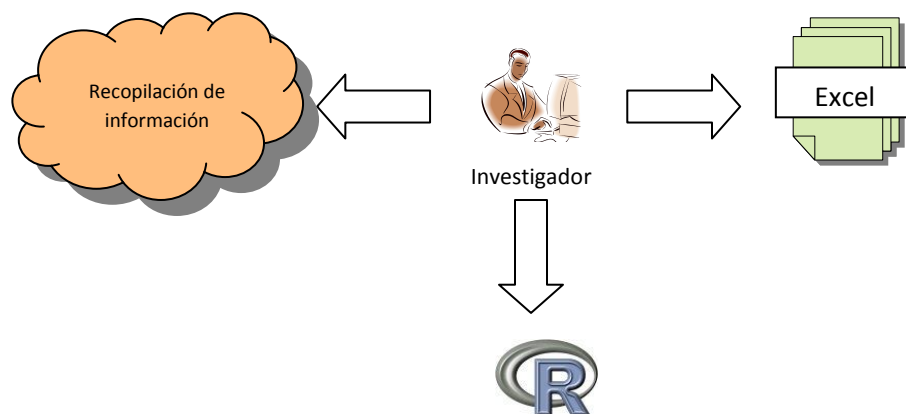


Figura 2: Esquema de la situación actual

2.2.2 Perfil del usuario

Esta aplicación será utilizada por los investigadores de la unidad de genética y mejora del Departamento de Ciencia Animal y de los Alimentos (UAB) que necesiten esta información para sus investigaciones.

También existirá un tipo de usuario que se encargará de la administración de la aplicación y de dar de alta a nuevos usuarios.

2.2.3 Objetivos

Los objetivos que se persiguen con este proyecto son proporcionar una aplicación multiusuario que facilite el intercambio de información, para así evitar posibles errores. También se persigue el objetivo de proporcionar una interfaz gráfica lo más intuitiva posible, que facilite la adaptabilidad de los usuarios a la nueva forma de trabajo.

En cuanto a las actividades que permitirá la aplicación, los usuarios tendrán la posibilidad tanto de insertar nuevos datos como modificarlos, además de consultar y eliminar información.

También se tendrá la opción de realizar análisis estadísticos sencillos predefinidos en lenguaje R con los datos almacenados.

2.2.4 Fuentes de información

La recogida de información para el proyecto comenzará en primer lugar con diferentes entrevistas con el profesor de la unidad de genética y mejora para tener una idea general de cómo se distribuyen los datos actualmente y así poder diseñar la base de datos.

En una segunda fase se recogerá información más enfocada a la implementación de la aplicación utilizando diferentes medios electrónicos, como páginas web dedicadas a lenguajes de programación [5-10], y escritos, como libros, relacionados con las distintas tecnologías utilizadas para el desarrollo, que pueden encontrarse en la biblioteca [1-4].

2.3. Sistema a realizar

2.3.1 Descripción

Con la información proporcionada por la unidad de genética y mejora, se diseñará e implementará una base de datos que permita el almacenamiento de los datos generados en los experimentos de genómica. Una vez implementada esta aplicación

los investigadores tendrán una herramienta de almacenamiento común y multiusuario que les evitará los posibles errores ocasionados por tener la información en archivos separados.

Después de crear la base de datos se diseñará e implementará una interfaz web para poder gestionar la información de la base de datos de una forma más fácil e intuitiva, y que además, permita subir archivos Excel con los datos a almacenar. Estos archivos tendrán una forma específica que también será fijada en este proyecto.

En la interfaz web se permitirá la elección de los diferentes experimentos, así como los datos asociados a estos experimentos. Una vez elegidos los datos, estos serán presentados de forma visual en la misma interfaz web.

Una vez seleccionados los datos se dará la oportunidad de realizar diferentes análisis estadísticos sencillos predefinidos con esta información y los resultados también serán mostrados a través de la interfaz gráfica.

Como se puede observar en la Figura 3 el sistema a realizar tiene diferentes páginas a las que podrá acceder el investigador. Una vez esté identificado, podrá decidir qué hacer, consultar o modificar los datos ya existentes en la base de datos o insertar nuevos datos mediante la utilización de la aplicación o subiendo la información en un archivo.

Además una vez consultados los datos o realizado un análisis estadístico con R, se podrán exportar los datos a un archivo.

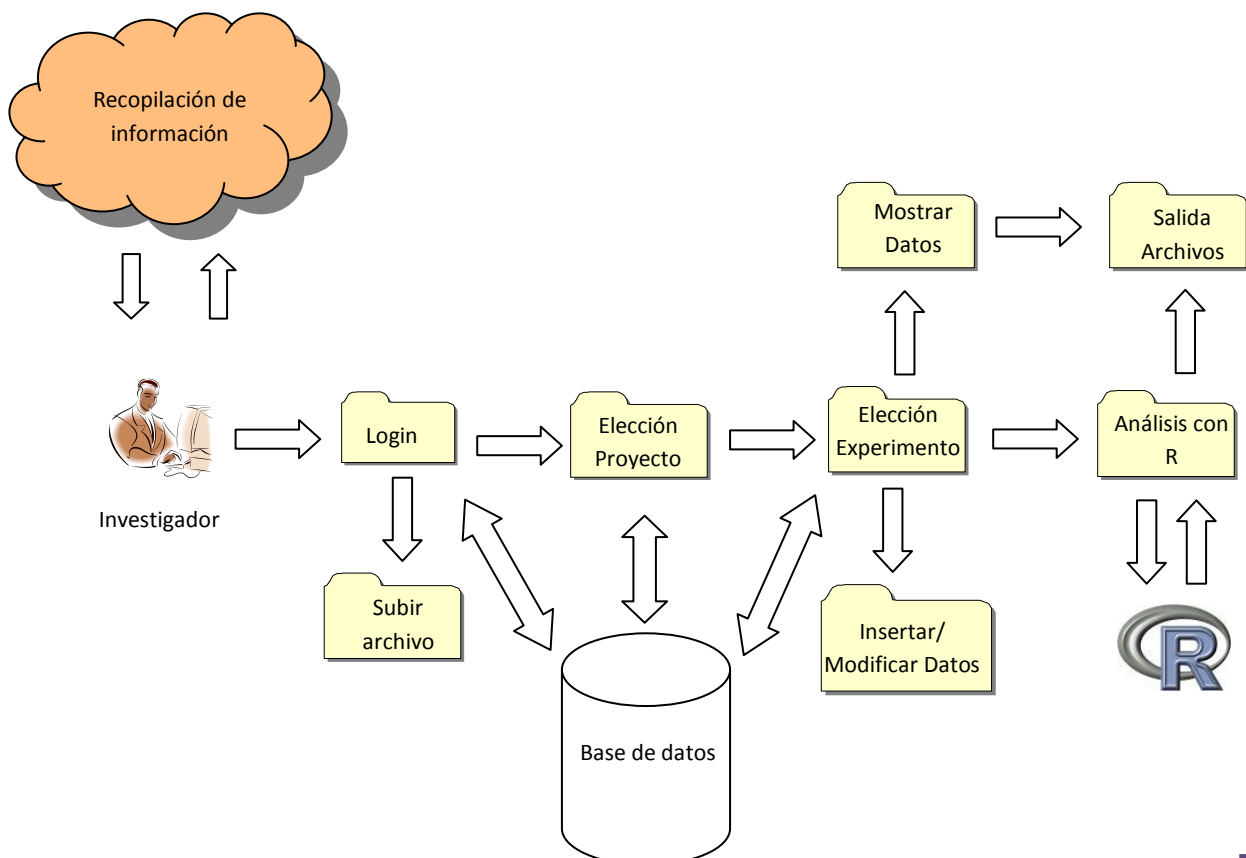


Figura 3: Esquema del sistema a desarrollar

2.3.2 Modelo de desarrollo

El modelo de desarrollo elegido para realizar este proyecto es el modelo lineal ya que el trabajo a realizar se puede dividir en diferentes fases o etapas sucesivas.

2.3.3 Planificación

A continuación se describe como se ha planificado el proyecto y las fases que se deberán realizar durante el desarrollo del mismo.

Fases del proyecto

Descripción de las tareas	Duración
Entrevistas con el profesor	40 h
Estudio previo del proyecto	5 h
Estudio de viabilidad	35 h
Diseño de la base de datos	35 h
Diseño de la interfaz de usuario	30 h
Programación de la aplicación	100 h
Programación de la base de datos	2 h
Programación de la interfaz de usuario	30 h
Programación de parser para la subida de archivos Excel	25 h
Programación de un script para la comunicación entre la aplicación y el lenguaje de análisis estadístico	10 h
Programación de hojas de estilo	23 h
Comprobación de cumplimiento de estándares	10 h
Pruebas	35 h
Elaboración de la documentación	50 h
Total	330 h

En la Figura 4 se puede observar el diagrama de Gantt del proyecto, con la estimación de la duración de cada una de las tareas una vez asignados los recursos.

Se puede ver que el proyecto comenzará el día 29 de octubre del 2009, y finalizará el día 10 de mayo del 2010, en total serán aproximadamente 8 meses de trabajo.

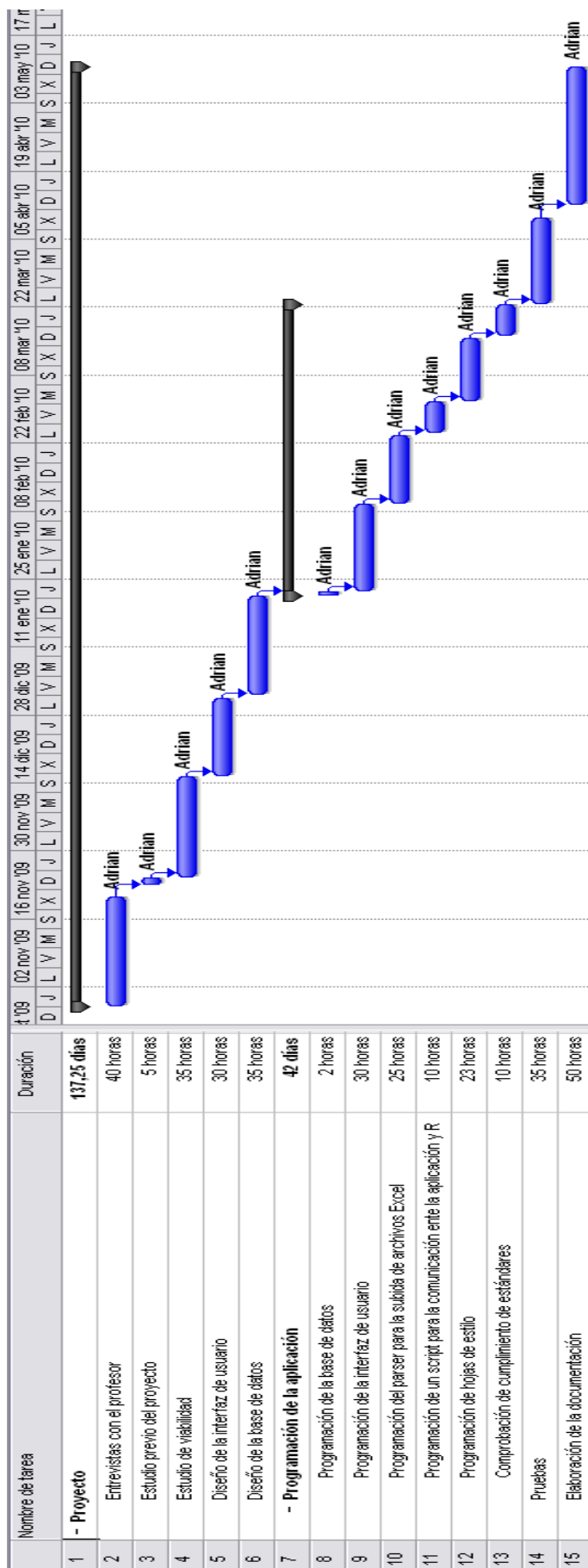


Figura 4: Diagrama de Gantt

2.3.4 Recursos de desarrollo

El desarrollo de la aplicación se llevará a cabo mediante el siguiente hardware y software.

- Hardware:
 - Procesador: Athlon 64 x2 4400+.
 - Memoria RAM: 2 GB.
 - Disco Duro: 250 GB.
 - Conexión a Internet.
- Software:
 - Servidor Web: Apache HTTP Server 2.2.
 - Programación Web: PHP, JavaScript, CSS.
 - Lenguaje estadístico: R.
 - Base de datos: MySQL.
 - Editor de base de datos: MySQL Administrator.
 - Editor de texto: Notepad++
 - Sistema operativo: Windows XP.
 - Navegador: Firefox, Internet Explorer.
 - Documentación del proyecto: MS Office y MS Project.

2.3.5 Recursos de implementación

Para implementar la aplicación se han considerado los siguientes requisitos mínimos para el servidor y para que el usuario puede acceder.

- Hardware del servidor:
 - Procesador: Pentium 4 o superior.
 - Memoria RAM: 1 GB.
 - Espacio Disco Duro: 10 GB.
 - Conexión a Internet
- Software del servidor:
 - Servidor web: Compatible con PHP y SSL.
 - Programación web: PHP.
 - Base de datos: MySQL.
- Hardware del usuario:
 - PC con conexión a Internet.
- Software del usuario:
 - Navegador web.

2.3.6 Evaluación de riesgos

A continuación se describen los posibles problemas que pueden surgir durante la realización del proyecto y una vez se ha finalizado, y también se darán sus posibles soluciones.

2.3.6.1 Lista de riesgos

Durante la realización del proyecto pueden surgir los siguientes problemas:

- **R1 – Incompatibilidad con navegadores:** Pueden haber problemas con el visionado de las páginas dependiendo del navegador utilizado debido a la diferente interpretación que estos realizan de los estándares web.
- **R2 – Problemas de conexión al servidor:** El fallo en la conexión al servidor puede suponer un retraso en la realización del proyecto.
- **R3 – Pérdida de información:** Puede ocurrir que se pierda la versión de la aplicación con la que se está trabajando.

También pueden surgir problemas después de la realización del proyecto:

- **R4 – Problemas de conexión:** Un fallo en la conexión a Internet puede suponer un retraso en los experimentos.
- **R5 – Problemas en el servidor:** Cualquier fallo en el servidor haría que fuera imposible acceder a la página, y por tanto, a la base de datos.
- **R6 – Software anticuado:** El software utilizado quede desfasado al haber otro más actual.
- **R7 – Base de datos corrupta:** La inserción de datos incorrectos puede ocasionar que se tenga información errónea.

2.3.6.2 Plan de contingencia

En este punto se muestran las posibles soluciones a los problemas anteriores.

- **R1 –** La solución sería realizar la aplicación de manera que cumpla con los estándares de la W3C, y por tanto, funcionará en los navegadores que cumplan con dichos estándares.
- **R2 –** La solución sería instalar un servidor local en el que poder realizar el proyecto en el caso de que el servidor no funcione.
- **R3 –** La solución sería realizar copias de seguridad de la aplicación y restaurarla en caso de fallo.

- **R4** – La solución sería buscar un lugar con acceso a la red.
- **R5** – La solución sería crear copias de seguridad de la base de datos y en el caso de que ocurra el problema instalar la aplicación y la copia de seguridad en otro servidor.
- **R6** – La solución sería actualizar la aplicación para que utilizara el nuevo software o seguir utilizando el software anticuado.
- **R7** – La solución sería realizar copias de seguridad de la base de datos cada vez que se realicen cambios.

2.3.7 Alternativas

A continuación se presentan las posibles alternativas al software utilizado para la implementación de la aplicación. Estas alternativas podrán ser utilizadas en caso de que la opción seleccionada no cumpla con las expectativas esperadas.

- **Base de datos:** Existen diferentes alternativas a MySQL, como pueden ser PostgreSQL, Oracle y MS SQL Server, pero se ha elegido MySQL por ser gratuito, multiplataforma y ser una de las bases de datos más utilizadas.
- **Programación Web:** Los tres lenguajes más utilizados a día de hoy para el desarrollo web son J2EE, ASP.NET y PHP. Se ha elegido PHP porque es un lenguaje ampliamente extendido en el desarrollo web además de ser gratuito y multiplataforma.
- **Servidor web:** Podría utilizarse cualquier servidor que soportará PHP y MySQL, pero se ha elegido Apache por ser multiplataforma y ser uno de los más utilizados.
- **Análisis estadísticos:** Hay diferentes alternativas a R, como puede ser Matlab, pero se ha optado por R por ser el lenguaje que usan en la unidad de genética y mejora y al mismo tiempo ser gratuito.

2.4. Análisis de coste-beneficio

Todo proyecto conlleva unos costes que han de tenerse en cuenta antes de comenzar a implementarlo. En el caso de este proyecto existen los siguientes costes:

Coste de material:

Recursos	Coste
Apache HTTP Server 2.2	0 €
PHP	0 €
MySQL	0 €
MySQL Administrator	0 €
Notepad++	0 €
R	0 €
Amortización PC programador	0 €
Amortización MS Office, MS Project	0 €
Servidor final	0 €
Total	0 €

Coste de personal:

El proyecto será realizado por el estudiante, con un coste de 15 €/hora.

Tarea	Horas	Coste
Entrevistas con el profesor	40	600,00 €
Estudio previo del proyecto	5	75,00 €
Estudio de viabilidad	35	525,00 €
Diseño de la interfaz de usuario	30	450,00 €
Diseño de la base de datos	35	525,00 €
Programación de la aplicación	100	1500,00 €
Programación de la base de datos	2	30 €
Programación de la interfaz de usuario	30	450 €
Programación de parser para la subida de archivos Excel	25	375 €
Programación de un script para la comunicación entre la aplicación y el lenguaje de análisis estadístico	10	150 €
Programación de hojas de estilo	23	345 €
Comprobación de cumplimiento de estándares	10	150 €
Pruebas	35	525,00 €
Elaboración documentación	50	750,00 €
Total	330	4950,00 €

El coste total previsto para el proyecto es de 4950 €. A este coste habría que añadir las horas que ha dedicado el tutor del proyecto, así como las personas de la unidad de

genética y mejora, pero en este caso no se han contabilizado ya que este coste esta dentro de los gastos de la universidad.

Beneficio del proyecto:

Para calcular el beneficio se considerarán las siguientes estimaciones.

La aplicación la utilizarán 10 usuarios, con el sistema actual cada usuario tarda 10 minutos en operar con los datos. Si se supone que al mes debe utilizar la información 10 veces cada usuario, se tiene que pierden 1000 minutos (16 horas y 40 minutos) en manipular la información cada mes, entre todos los usuarios. Además también pierde un tiempo en buscar los ficheros donde se encuentra la información, que se ha estimado que son 7 minutos cada vez, esto suma 700 minutos (11 horas y 40 minutos) al tiempo anterior.

También se pierde tiempo en utilizar esa información en análisis estadísticos con lo que se ha considerado un tiempo de 2 minutos cada vez, con lo que sería 200 minutos al mes entre todos los usuarios.

Por otra parte, se pierde un tiempo en la comunicación entre los investigadores para intercambiar información, este tiempo se ha estimado en 120 minutos (2 horas) usuario/mes, con lo que tendríamos un tiempo de 1200 minutos.

Así se tiene que en total se pierden al mes:

- 1000 minutos en operar con la información.
- 700 minutos en buscar el fichero.
- 200 minutos en realizar análisis estadísticos.
- 1200 minutos en intercambiar información.

Con lo que en total se pierden al mes 3100 minutos (51 horas y 40 minutos), con lo que cada usuario pierde al mes 310 minutos (5 horas y 10 minutos).

Si además consideramos que los usuarios tienen un sueldo 1600 €/mes y que trabajan al mes 9600 minutos (160 horas) por usuario (192000 minutos entre todos), se tiene que el gasto al mes de manipular la información es de 51,66 €/usuario (516,67 entre todos los usuarios).

Con la implementación de la aplicación estos tiempos descenderían drásticamente, y se tendría que el tiempo de operar con los datos pasa de 10 a 2 minutos. La duración de utilizar la información en análisis estadísticos pasaría de 2 minutos a 1 minuto. Además los tiempos de buscar el fichero e intercambiar información desaparecerían al tener una base de datos en común. Con lo que quedaría:

- 200 minutos en operar con la información.
- 0 minutos en buscar el fichero.
- 100 minutos en realizar análisis estadísticos.
- 0 minutos en intercambiar información.

Por lo tanto ahora en total se pierden 300 minutos (5 horas), 30 minutos por cada usuario, que supondría un coste de 5 €/usuario (50 € entre todos los usuarios).

Así se tendrá que el cliente se ahorrará $516,67 - 50 = 466,67€$ al mes. Considerando esta cifra y persiguiendo que la aplicación esté amortizada en 24 meses se ha considerado que el margen será de $(466,67 - (4950/24))/2 = 130,21$ por mes, y en total quedará un margen de 3125,04, de esta manera el precio de venta quedará:

$$\text{Precio de venta} = \text{coste} + \text{margen} = 4950 + 3125,04 = 8075,04$$

Con lo que el precio de venta del proyecto será de 8075,04 €, y estará amortizado en unos 24 meses.

También se tendrá que tener en cuenta que la aplicación impedirá la pérdida total de información ya que se realizarán copias de seguridad de la base de datos, y se evitará la repetición de experimentos que tendrían un coste mayor que la realización de esta aplicación.

Con todo esto y teniendo en cuenta que se trata de un proyecto final de carrera, estos costes pueden considerarse nulos ya que todos los materiales utilizados son gratuitos o pueden utilizarse gratuitamente en la universidad. El coste de personal es asumido por el alumno ya que es un trabajo que forma parte de la carrera en curso.

2.5. Conclusiones

Se han analizado las ventajas e inconvenientes que podría ocasionar el desarrollo del proyecto y se ha comprobando que es viable técnica, temporal y económicamente.

Es viable técnicamente ya que como se ha visto el software usado es de los más utilizados y eso conlleva que este bien documentado y testeado. También es factible temporalmente, ya que se ha visto en la planificación que se podrá realizar durante un año académico y así mismo es factible económicamente, ya que los gastos de personal son asumidos por el alumno al tratarse del proyecto de final de carrera y los materiales utilizados son proporcionados por la universidad. Por lo tanto se puede considerar que es un proyecto viable.

3. Análisis

3.1 Introducción

Este apartado ayudará a comprender cuál es el problema o necesidad que se pretende solucionar. Así se analizarán los requerimientos del software, para obtener un buen resultado y poder predecir los posibles errores. También se recopilarán los datos con los que trabaja la unidad de genética y se realizará una primera aproximación al modelo de datos.

También se verán las acciones que podrán realizar los usuarios de esta aplicación. Estos usuarios serán los investigadores, que son las personas que realizan los experimentos con los datos que les llegan desde el laboratorio y el administrador, que aparte de poder realizar las acciones de un investigador también podrá realizar las acciones administrativas de la aplicación.

3.2 Requerimientos funcionales

Los requisitos funcionales que debe cumplir el proyecto son los siguientes:

- *Eficiencia y facilidad de uso:* La aplicación debe ser intuitiva y que la navegación sea clara haciendo que el usuario se sienta cómodo al trabajar con ella.
- *Control de acceso:* Es importante tener un buen nivel de seguridad para impedir el acceso a cuentas ajenas, y más si se trata de una cuenta de administrador. Por lo tanto es necesario un protocolo de seguridad basado en usuario y contraseña.
- *Gestión de usuario:* Los usuarios que sean administradores podrán dar de alta a otros usuarios que podrán ser administradores o usuarios normales.
- *Gestión de contenidos:* Todos los usuarios podrán modificar e insertar nuevos datos en la base de datos, así como descargarlos en ficheros.
- *Registro de actividades:* Los administradores tendrán una lista con las actividades que ha realizado cada usuario y la fecha en la que se ha producido.
- *Copias de seguridad:* Los administradores también tendrán la posibilidad de descargarse una copia de seguridad de la base de datos.

3.3 Requerimientos no funcionales

Los requerimientos no funcionales que debe cumplir el proyecto son los siguientes:

- *Seguridad:* El proyecto tiene que garantizar cierta privacidad para los datos de los usuarios, por lo tanto la base de datos deberá estar protegida con contraseña y todas las contraseñas de los usuarios estarán cifradas utilizando el algoritmo MD5.
- *Cumplimiento de estándares:* La aplicación deberá cumplir con los estándares para su correcta visualización en los distintos navegadores. Se utilizará el estándar XHTML de W3C, separando la estructura de su presentación.
- *Control de errores:* Todos los posibles errores cometidos por los usuarios deberán estar contemplados y controlados
- *Amigabilidad:* La aplicación deberá ser fácil de usar para el usuario.
- *Interfaces:* Se deberán considerar los formatos de los archivos de salida y su compatibilidad con las diferentes herramientas de ofimática.
- *Mantenibilidad:* La aplicación deberá ser sencilla de mantener y se desarrollarán documentos de implementación y de uso de la aplicación.
- *Portabilidad:* La aplicación deberá ser capaz de migrar de una plataforma a otra sin que esto represente grandes inconvenientes al usuario. Esto se conseguirá parametrizando las variables de configuración del sistema.
- *Registro de eventos:* Se realizará un registro de eventos del sistema, empleando la información de la sesión del usuario.
- *Ayudas:* Se presentarán ayudas en línea para orientar al usuario en el uso de la interfaz.
- *Contingencias:* Se deberán prever contingencias para eventos de caída del sistema.
- *Conexiones:* Se permitirán múltiples conexiones al sistema simultáneamente.

3.4 Información proporcionada

La información se obtuvo en un primer momento con entrevistas con Miguel Pérez Enciso que dio una primera aproximación de los datos que se tendrían y como se relacionaban. Durante las entrevistas se proporcionó información que estaba en archivos Excel, con el formato mostrado en la Figura 5. Esta información representaba principalmente los fenotipos y genotipos de individuos estudiados en experimentos y se tenía el problema de que la información estaba repetida en las diferentes hojas, con lo que se procedió a identificar cada elemento con el fin de poder agruparlos en una sola base de datos relacional. Durante su análisis se observó que los datos tenían una estructura definida (tipos de datos) y que estaban divididos en grupos según el tiempo de muestra, estos grupos se denominaron lotes. También se observó que los lotes pertenecían a experimentos distintos.

Además cada dato perteneciente a un lote tenía un tipo de dato distinto, con lo que se decidió que la aplicación debería guardar estos últimos para que fuera flexible con los distintos experimentos y los datos pertenecientes a estos.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	TATUAJE	SEXO	PADRE	MADRE	EDAD_SAC	LOTE	PESO NETO	Nº DIA2	Nº LAB	C14:0	C16:0	C18:1	C18:0	C18:1 (n-9)	C18:1 (n-7)	C18:2	C18:3	C20:1	C20:2
2	704851	1	12603177	12603062	175	2	80.16	29	29	1.83	23.37	2.72	11.52	42.03	2.96	13.20	0.90	0.92	0.56
3	704852	1	12603177	12603062	175	2	84.77	14	14	2.12	23.47	3.73	9.08	43.34	3.84	12.19	0.93	0.82	0.49
4	704853	1	12603177	12603062	175	2	79.87	51	50	1.38	22.32	2.07	12.55	42.98	2.71	13.72	0.87	0.77	0.62
5	704854	1	12603177	12603062	175	2	89.95	2	2	1.80	23.02	2.92	10.45	43.37	3.24	12.79	0.90	0.96	0.56
6	704855	2	12603177	12603062	175	2	73.34	49	48	1.65	23.38	2.61	10.70	43.57	3.00	12.99	0.89	0.66	0.55
7	704856	2	12603177	12603062	175	2	82.66	56	55	1.59	24.11	2.26	13.07	40.52	2.66	13.57	0.95	0.75	0.53
8	704857	2	12603177	12603062	175	2	77.88	50	49	1.60	24.36	2.46	12.25	42.39	2.73	11.98	0.85	0.85	0.54
9	801162	1	12603176	12603375	171	9	83.90	21	457	1.56	21.47	2.78	10.85	47.90	3.46	9.75	0.85	0.88	0.50
10	801163	1	12603176	12603375	171	9	73.63	70	495	1.28	21.01	2.33	11.98	45.83	2.95	12.12	1.12	0.84	0.52
11	801164	1	12603176	12603375	171	9	67.30	32	466	1.61	20.59	3.33	9.52	46.85	3.62	12.06	1.25	0.68	0.49
12	801165	1	12603176	12603375	171	9	68.93	40	474	1.47	20.81	2.42	11.55	46.03	3.04	12.07	1.17	0.91	0.54
13	801166	1	12603176	12603375	171	9	77.09	7	443	1.53	19.06	2.37	10.72	49.59	3.45	10.64	0.89	1.13	0.62
14	801167	1	12603176	12603375	171	9	11.76	61	401	1.45	20.31	2.71	11.02	45.60	3.22	12.08	1.17	0.81	0.65

Figura 5: Formato del archivo Excel proporcionado

Gracias a las entrevistas y a la información proporcionada se llegó a la conclusión de que los datos se deberían agrupar de una forma similar a la que se muestra en la Figura 6.

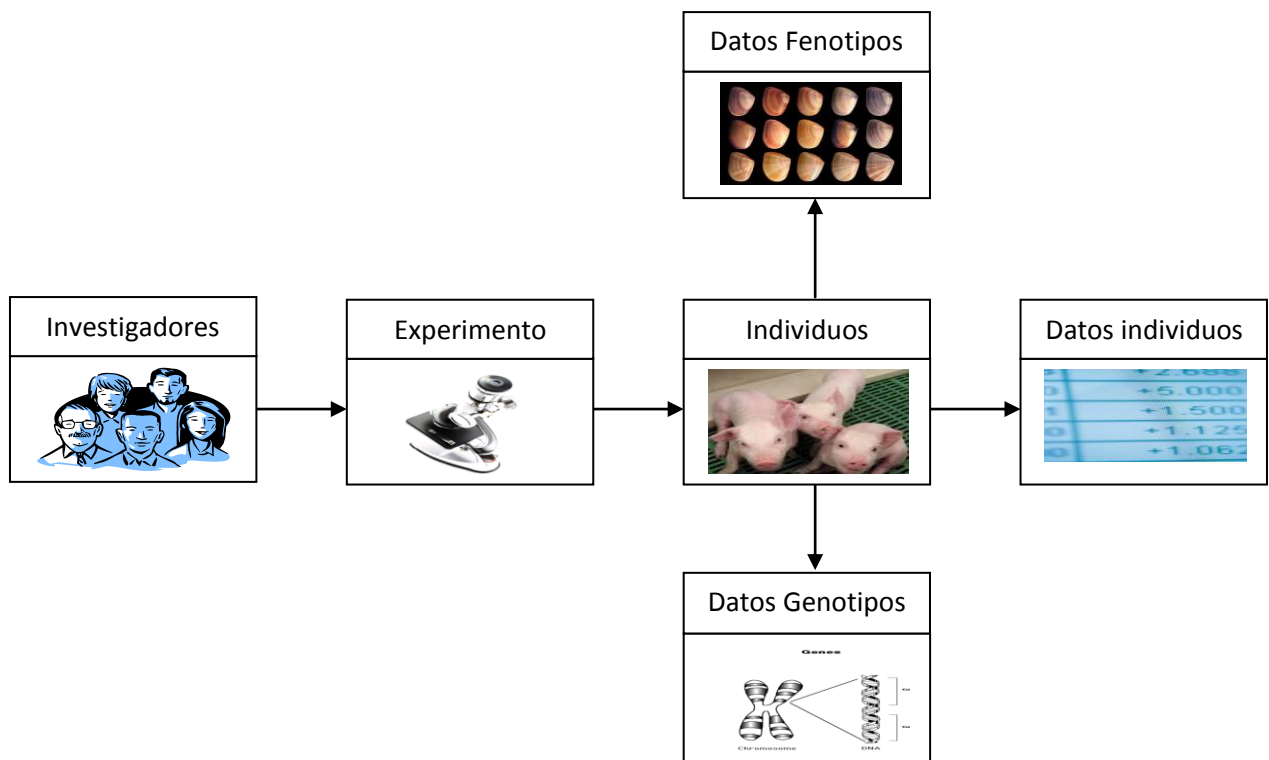


Figura 6: Primera aproximación del modelo de tipo de datos.

3.5 Diagramas de casos de uso

Con los diagramas de caso de uso se mostrarán las diversas acciones que podrán realizar los diferentes usuarios de la aplicación.

3.5.1 Investigador

En la Figura 7 se puede observar el diagrama de casos de uso del usuario investigador.

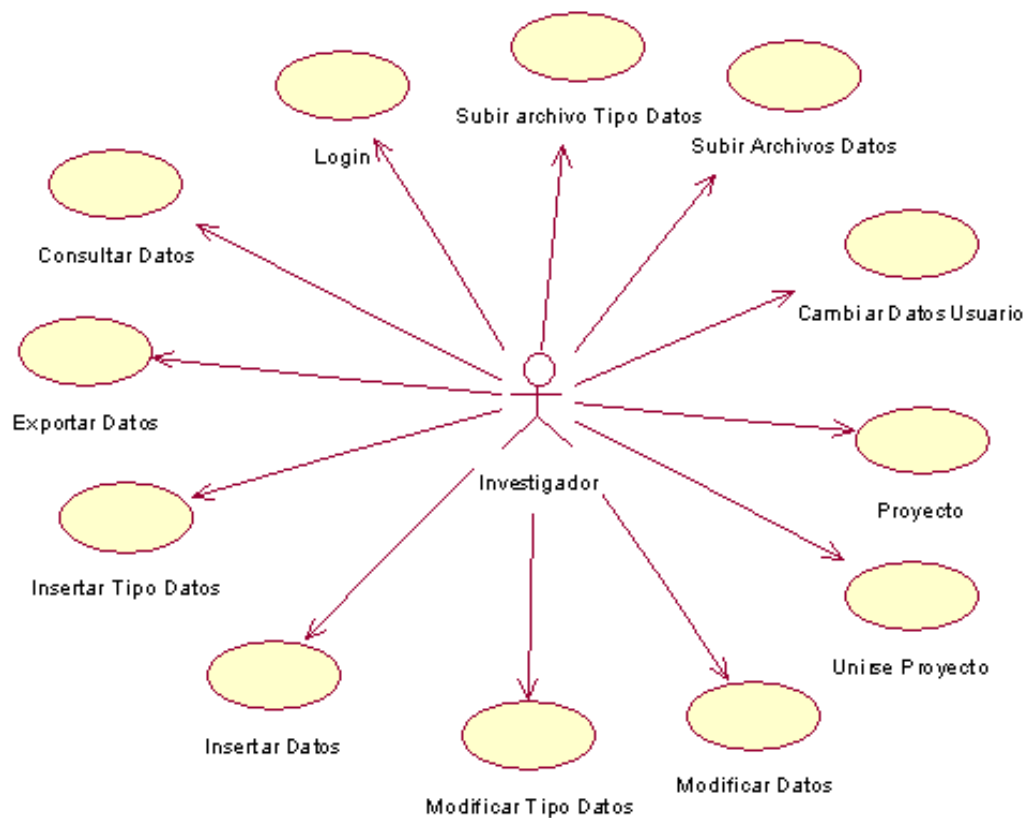


Figura 7: Diagrama de casos de uso del investigador.

3.5.2 Administrador

En la Figura 8 se muestra el diagrama de casos de uso del Administrador de la aplicación. Este diagrama contiene los mismos casos de uso que el del investigador con la excepción de que se han añadido los casos de Backup de la base de datos, Consulta de Logs y el Registro de usuario.

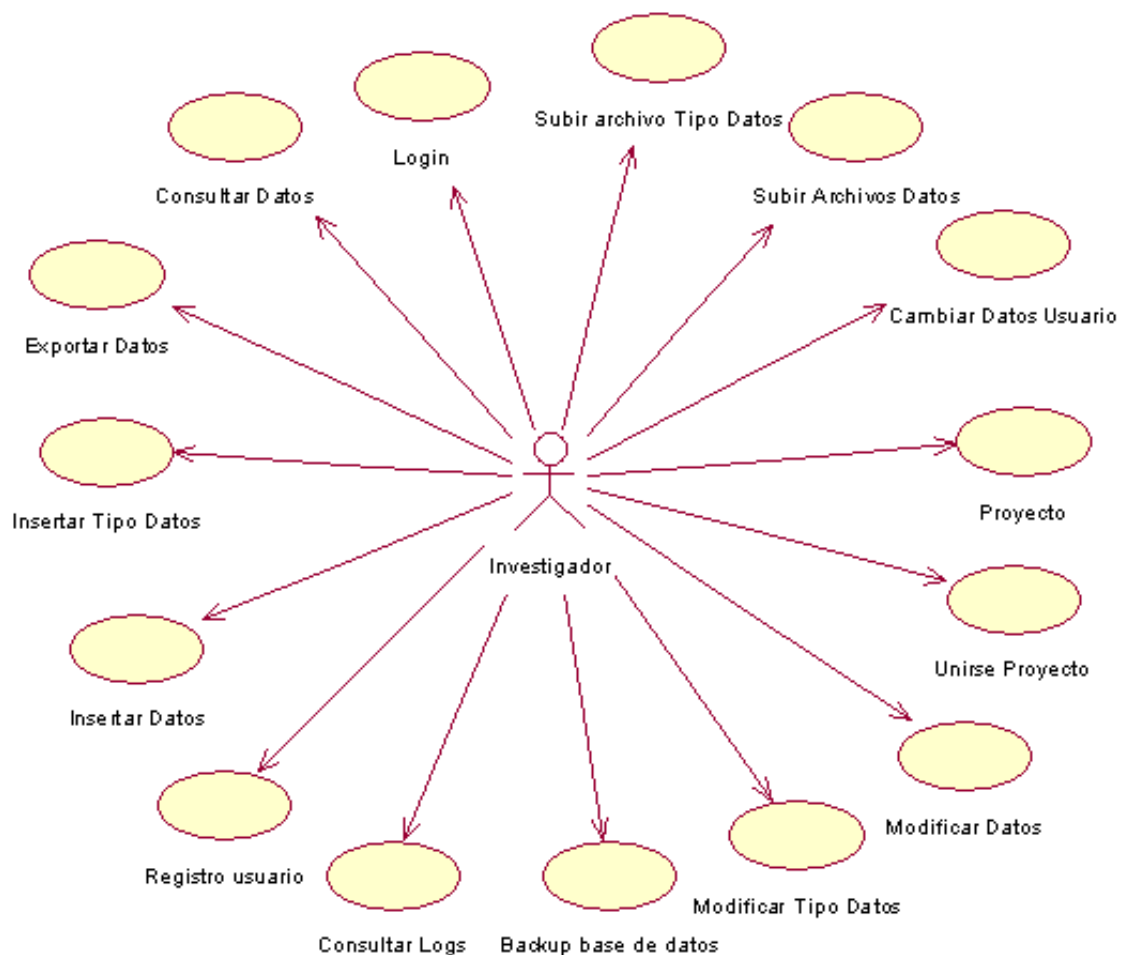


Figura 8: Diagrama de casos de uso del Administrador.

3.6 Descripción de casos de uso

A continuación se describirán los casos de uso del apartado anterior de forma más detallada.

3.6.1 Investigador

En este apartado se describen las actividades que puede realizar el usuario que no es administrador.

Caso de uso	Login
Descripción	Los usuarios que estén registrados podrán identificarse para así acceder a las siguientes secciones de la aplicación. Para ello se introducirá el usuario y contraseña en el menú correspondiente y se pulsará el botón <i>Enviar</i> .
Actores	Investigador, Administrador
Pre-Condición	Entrar en la dirección URL, estar registrado.
Flujo Principal	Después de introducir los datos en el formulario y pulsar el botón <i>Enviar</i> se accederá a la página principal de la aplicación donde se mostrará un menú con los diferentes apartados.
Flujo Alternativo	<u>Datos Incorrectos:</u> Si alguno de los datos introducidos no coincide con los que hay en la base de datos, se mostrará un mensaje de error en el navegador. <u>Contraseña olvidada:</u> Si el usuario se ha olvidado de la contraseña tendrá que regenerarla respondiendo a una pregunta de seguridad que habrá escrito anteriormente.

Caso de uso	Cambiar datos de usuario
Descripción	El usuario podrá cambiar todos sus datos de registro, también el nombre de usuario, si no está siendo usado.
Actores	Investigador, Administrador.
Pre-Condición	Estar identificado.
Flujo principal	Una vez rellenado el formulario y pulsado el botón enviar se mostrará un mensaje indicando que los datos han sido modificados correctamente.
Flujo Alternativo	<u>Algún campo está vacío:</u> Si alguno de los campos obligatorios está vacío se mostrará un mensaje de error indicándolo.

Caso de uso	Proyecto
Descripción	Los usuarios podrán crear o eliminar un proyecto.
Actores	Investigador, Administrador.
Pre-Condición	Estar identificado.
Flujo Principal	El usuario del proyecto pulsará sobre el botón de <i>Crear proyecto</i> y se le mostrará un formulario con los datos necesarios para crear un nuevo proyecto.
Sub-Flujo	<u>Botón eliminar proyecto:</u> Pulsando este botón se le preguntará si desea eliminar el proyecto y en caso afirmativo se procederá a su eliminación.
Flujos Alternativos	<u>El proyecto ya existe:</u> Si el usuario intenta crear un proyecto que ya existe previamente, se mostrará por pantalla un error advirtiendo que la abreviatura del proyecto escrita ya está siendo utilizada por un proyecto anterior.

Caso de uso	Unirse Proyecto
Descripción	Los usuarios podrán unirse o borrarse de un proyecto.
Actores	Investigador, Administrador.
Pre-Condiciones	Estar identificado, que el proyecto y el usuario a unir existan.
Flujo Principal	El jefe del proyecto seleccionará a los usuarios que quiere que se unan al proyecto y pulsará el botón <i>Unirse</i> . Una vez pulsado dicho botón, se mostrará un mensaje que informará de que la operación se ha realizado correctamente.
Sub-Flujo	<u>Botón eliminar del proyecto:</u> Pulsando este botón se mostrará un mensaje preguntando si realmente se desean eliminar los usuarios seleccionados, y en caso afirmativo se eliminarán del proyecto.

Caso de uso	Subir archivo tipo de datos
Descripción	Los investigadores podrán subir un archivo con los diferentes tipos de datos que serán utilizados en la aplicación. Para ello se seleccionará el archivo y se pulsará el botón <i>Subir</i> .
Actores	Investigador, Administrador.
Pre-Condición	Estar identificado.
Flujo Principal	Después de seleccionar el archivo y pulsar el botón <i>Subir</i> , se analizarán los datos para ver si son correctos y se mostrará una página que nos preguntará si queremos insertar la información en la base de datos.
Flujo Alternativo	<u>Datos Incorrectos:</u> Si alguno de los datos del archivo es incorrecto se mostrará una página donde se informará del error y que permitirá descargar un archivo con información sobre dicho error. <u>Datos Existentes:</u> Si los datos ya existen en la base de datos se mostrará un mensaje indicándolo y se podrá descargar un fichero con los errores.

Caso de uso	Subir archivo de datos
Descripción	Los investigadores podrán subir un archivo con los diferentes datos que serán añadidos a la base de datos. Para ello se seleccionará el archivo y se pulsará el botón <i>Subir</i> .
Actores	Investigador, Administrador.
Pre-Condiciones	Estar identificado, que los tipos de datos existan en la aplicación.
Flujo Principal	Después de seleccionar el archivo y pulsar el botón <i>Subir</i> , se analizarán los datos para ver si son correctos y se mostrará una página donde preguntará si queremos insertar la nueva información en la base de datos.
Flujo Alternativo	<p><u>Datos Incorrectos:</u> Si alguno de los datos del archivo es incorrecto se mostrará una página que informará al usuario sobre el problema ocurrido y le dará la opción de guardar el informe del error en un archivo.</p> <p><u>Datos Existentes:</u> Si la información ya existe en la base de datos se mostrará una página indicando el error y se permitirá descargarlos en un archivo.</p>

Caso de uso	Consultar datos
Descripción	Los usuarios podrán seleccionar que datos quieren que se muestren por pantalla dependiendo de si son fenotipos, genotipos o características del individuo.
Actores	Investigador, Administrador.
Pre-Condiciones	Estar identificado, haber insertado datos y tipos de datos previamente.
Flujo Principal	Después de seleccionar el tipo de consulta a realizar se mostrará por pantalla los datos seleccionados.
Sub-Flujos	<p><u>Botón Exportar Datos:</u> Si se pulsa el botón <i>Exportar</i>, los datos seleccionados serán exportados a un archivo para que el usuario pueda descargarlos a su ordenador.</p>

Caso de uso	Insertar datos
Descripción	Los investigadores podrán insertar nuevos datos rellorando el formulario y pulsando el botón <i>Enviar</i> .
Actores	Investigador, Administrador.
Pre-Condiciones	Estar identificado, que exista un proyecto, que existan los tipos de datos.
Flujo Principal	Una vez dentro del apartado <i>Gestión de Datos</i> se mostrarán los enlaces, uno para insertar los fenotipos, otro para insertar los genotipos, otro para insertar los marcadores, otro para insertar los lotes y el último para insertar las características del individuo.
Sub-Flujos	<p><u>Fenotipos:</u> Se mostrará un formulario con los datos necesarios para insertar un nuevo fenotipo al individuo.</p> <p><u>Genotipo:</u> Se mostrará un formulario para insertar los datos necesarios para crear un nuevo genotipo para un individuo.</p> <p><u>Marcador:</u> Se mostrará un formulario para añadir un nuevo marcador a la aplicación.</p> <p><u>Características del individuo:</u> Se mostrará un formulario para insertar las características del individuo.</p> <p><u>Lote:</u> Se mostrará un formulario para añadir un nuevo lote al proyecto especificado.</p>
Flujos Alternativos	<p><u>Tipos de datos del fenotipo, genotipo, marcador, característica del individuo o lote no existen:</u> Si el tipo de dato no existe en la base de datos se mostrará un mensaje indicándolo.</p> <p><u>El individuo no existe:</u> Si se intenta insertar información sobre un individuo que no existe en la base de datos se mostrará un mensaje de error indicándolo.</p> <p><u>Datos incorrectos:</u> Si alguno de los datos insertados es incorrecto se mostrará un mensaje de error donde se informará al usuario.</p>

Caso de uso	Insertar tipo de datos
Descripción	Los investigadores podrán insertar nuevos tipos de datos rellenando el formulario y pulsando el botón <i>Enviar</i> .
Actores	Investigador, Administrador.
Pre-Condiciones	Estar identificado.
Flujo Principal	Una vez rellenados los campos del formulario y pulsado el botón <i>Enviar</i> se mostrará un mensaje indicando que los datos se han guardado correctamente.
Flujo Alternativo	<p><u>Datos incorrectos:</u> Si alguno de los datos introducidos es incorrecto se mostrará un mensaje indicando el error producido.</p> <p><u>Datos existentes:</u> Si el tipo de dato ya existe en la base de datos se mostrará un mensaje por pantalla indicado esta incidencia.</p>

Caso de uso	Exportar Datos
Descripción	El investigador podrá exportar los datos de un proyecto a un archivo para poderlos guardar fuera de la aplicación.
Actores	Investigador, Administrador.
Pre-Condiciones	Estar identificado, que haya datos y tipos de datos.
Flujo Principal	El investigador elegirá el proyecto del cual quiere exportar la información y el tipo de datos a guardar (fenotipos, genotipos o características del individuo).

Caso de uso	Modificar tipo de datos
Descripción	Los investigadores podrán modificar/eliminar los tipos de datos existentes en la aplicación rellenando el formulario y pulsando el botón <i>Modificar/eliminar</i> .
Actores	Investigador, Administrador.
Pre-Condiciones	Estar identificado, que el dato a modificar exista.
Flujo Principal	Una vez rellenados los campos del formulario y pulsado el botón <i>Modificar/Eliminar</i> se mostrará un mensaje indicando que los datos se han modificado/eliminado correctamente.
Flujo Alternativo	<p><u>Datos incorrectos:</u> Si alguno de los datos introducidos es incorrecto se mostrará un mensaje indicando el error producido.</p>

Caso de uso	Modificar datos
Descripción	Los investigadores podrán modificar/eliminar los datos rellenando el formulario y pulsando el botón <i>Modificar/Eliminar</i> .
Actores	Investigador, Administrador.
Pre-Condiciones	Estar identificado, que el dato a modificar exista.
Flujo Principal	Una vez dentro del apartado <i>Gestión de Datos</i> se mostrarán los enlaces, uno para modificar los fenotipos, otro para modificar los genotipos, otro para modificar los marcadores, otro para modificar los lotes y el último para modificar las características del individuo.
Sub-Flujos	<p><u>Fenotipos:</u> Se mostrará un formulario con los datos necesarios para modificar un fenotipo ya existente.</p> <p><u>Genotipo:</u> Se mostrará un formulario para modificar los genotipos ya existentes.</p> <p><u>Marcador:</u> Se mostrará un formulario para modificar los marcadores.</p> <p><u>Características del individuo:</u> Se mostrará un formulario para modificar las características del individuo.</p> <p><u>Lote:</u> Se mostrará un formulario para modificar el lote de un proyecto.</p>
Flujos Alternativos	<p><u>Tipos de datos del fenotipo, genotipo, marcador, característica del individuo o lote no existen:</u> Si el tipo de dato existe en la base de datos se mostrará un mensaje indicándolo.</p> <p><u>El individuo no existe:</u> Si se intenta modificar información sobre un individuo que no existe en la base de datos se mostrará un mensaje de error indicándolo.</p> <p><u>Datos incorrectos:</u> Si alguno de los datos modificados es incorrecto se mostrará un mensaje de error donde se informará al usuario.</p>

3.6.2 Administrador

El administrador podrá realizar las mismas tareas que el investigador además de registrar a los usuarios que considere oportuno, consultar los logs, etc. Por tanto, en este apartado solo se describirá los casos de uso que no estén incluidos en el apartado 3.5.1.

Caso de uso	Registro de usuarios
Descripción	El administrador será el encargado de registrar a los usuarios que podrán utilizar la aplicación. Para llevar a cabo esta operación el administrador deberá rellenar el formulario indicado en dicho apartado.
Actores	Administrador.
Pre-Condición	Estar identificado como administrador.
Flujo Principal	Una vez rellenado el formulario y pulsado el botón <i>Enviar</i> se mostrará por pantalla un mensaje indicando que el usuario ha sido registrado correctamente y se enviará un email al nuevo usuario indicándole que ya está registrado.
Flujo Alternativo	<u>El usuario ya existe:</u> Si el usuario ya existe en la base de datos se mostrará un mensaje por pantalla indicando el error que se ha producido. <u>Algún campo está vacío:</u> Si alguno de los campos está vacío se enseñará un mensaje de error indicando el problema.

Caso de uso	Consultar Logs
Descripción	Todas las operaciones que realicen los usuarios se registrarán en una tabla de la base de datos, que posteriormente podrá consultar el administrador de la aplicación.
Actores	Administrador.
Pre-Condiciones	Estar identificado como administrador, que se hayan realizado alguna acción en la aplicación.
Flujo principal	Una vez el administrador a entrado en la página de logs se mostrarán todas las operaciones que han realizado los usuarios sobre la base de datos.

Caso de uso	Backup base de datos
Descripción	El administrador podrá guardar una copia del estado actual de la base de datos.
Actores	Administrador.
Pre-Condición	Estar identificado como administrador.
Flujo principal	Cuando el administrador pulse sobre el link del backup se le permitirá descargar un fichero SQL, que podrá utilizar en cualquier otra base de datos.

3.7 Diagrama de actividades

Con el diagrama de actividades se podrán ver los diferentes caminos que se podrán seguir al realizar una acción, también se tendrá un primer contacto con las acciones que se podrán llevar a cabo dentro de la aplicación.

3.7.1 Investigador

En primer lugar el investigador deberá identificarse en la aplicación, si la contraseña introducida no es correcta se volverá a la página de principal de identificación. Una vez identificado, se le permitirá importar un archivo, realizar una consulta, crear o unirse a un proyecto, insertar datos o modificar la información de usuario. Una vez se ha realizado una acción el usuario volverá a la página principal y tendrá la posibilidad de realizar otras acciones. En la Figura 9 se puede observar con más detalle las actividades que podrá realizar el investigador desde que se registre. Hay que tener en cuenta que el gráfico se ha dividido en 2 imágenes para un visionado más cómodo.

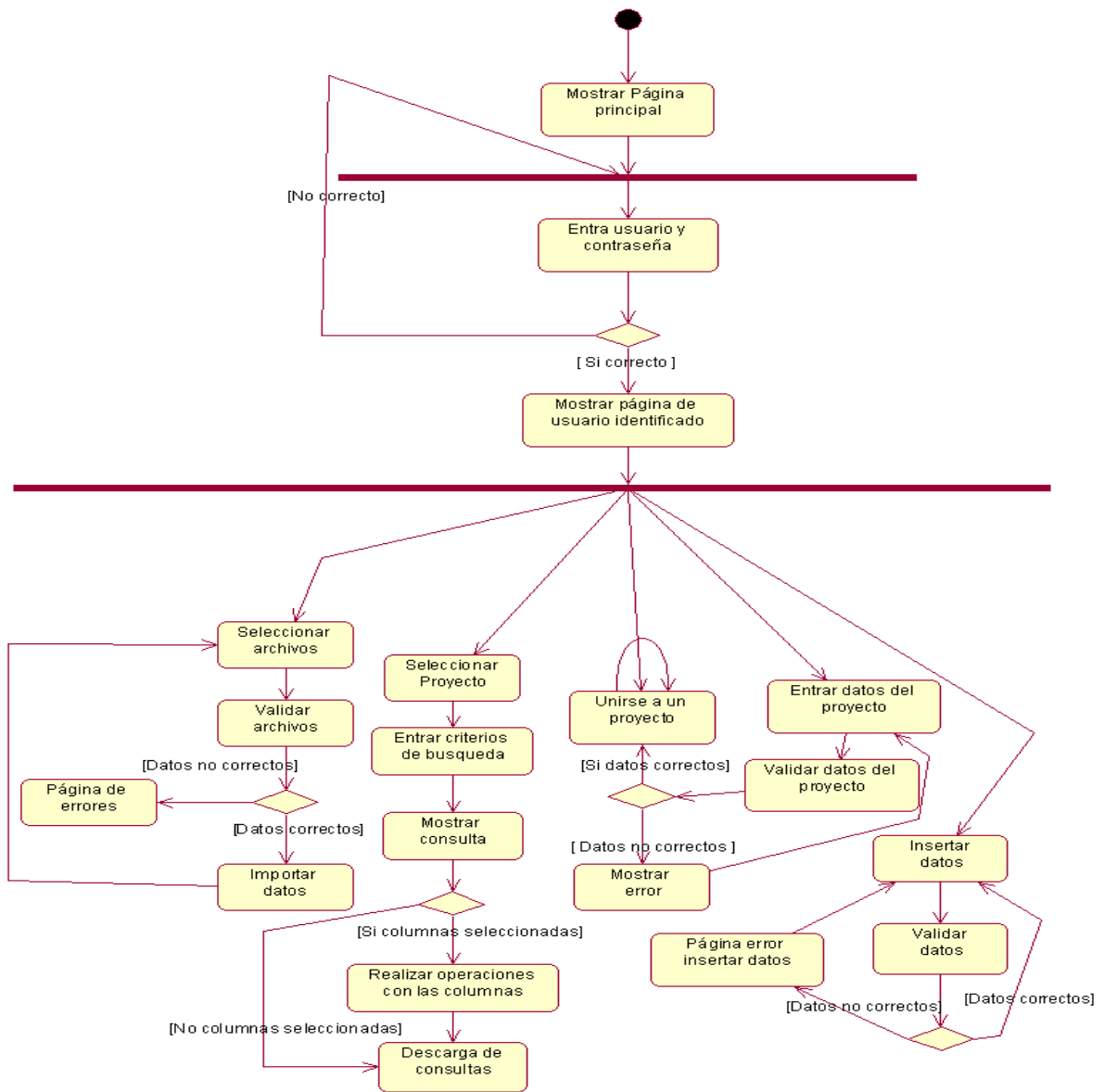


Figura 9 (a): Diagrama de actividad del investigador.

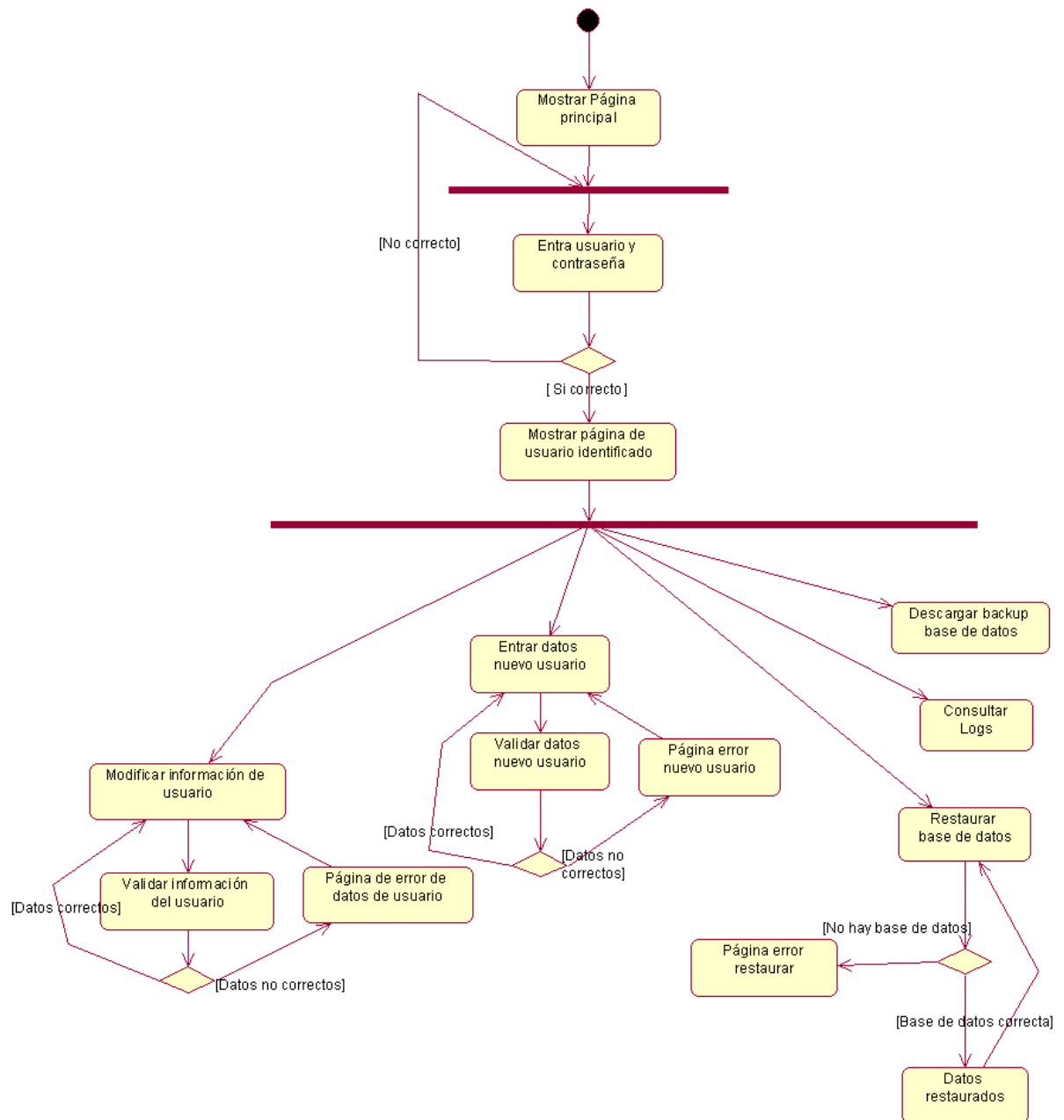


Figura 9 (b): Diagrama de actividad del investigador.

3.7.2 Administrador

La primera acción que podrá realizar el administrador será identificarse en la aplicación. Una vez identificado, se le permitirá realizar las mismas tareas que el investigador además de poder registrar a nuevos usuarios, consultar los logs de la aplicación, descargar una copia de la base de datos o restaurarla. En la Figura 10 se puede observar con más detalle las actividades que podrá realizar el administrador después de registrarse.

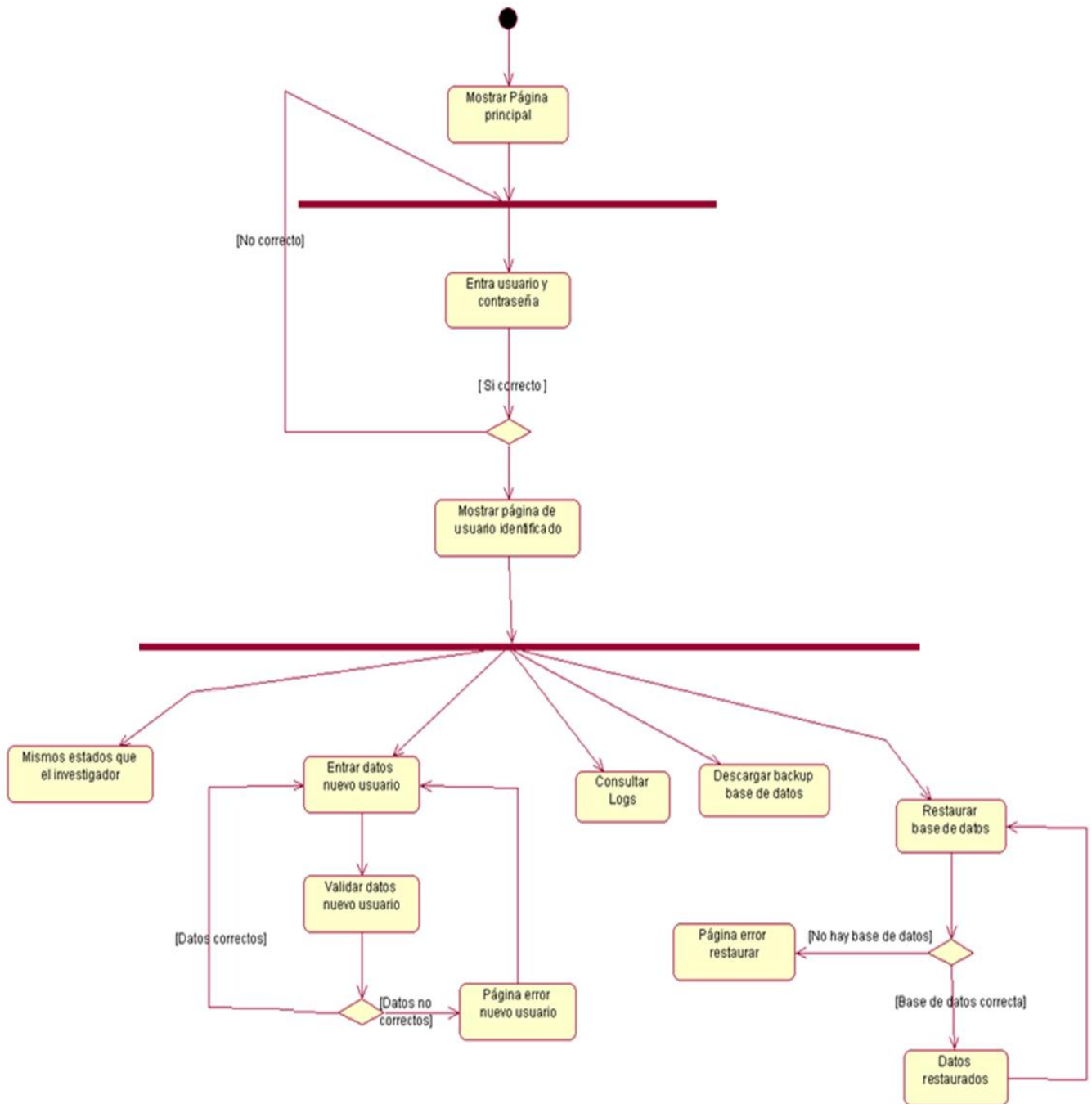


Figura 10: Diagrama de actividad del administrador.

4. Marco tecnológico

4.1 Introducción

En este apartado se explicarán las diferentes herramientas utilizadas para el desarrollo del proyecto y porque se han seleccionado estas y no otras, además también se explicarán los lenguajes de programación usados para su realización.

4.2 Tecnologías web

A continuación se describirán las características de varias tecnologías utilizadas para programar aplicaciones web.

Se ha elegido PHP [5] por ser uno de los lenguajes de servidor más populares entre las páginas web y ser de código abierto y multiplataforma. Como todo lenguaje de servidor, su código se ejecuta en el servidor web, transformando la página original escrita en PHP en código HTML que pueda reproducir cualquier navegador.



Es un lenguaje muy rápido con una gran cantidad de librerías que permiten generar desde archivos PDF hasta imágenes.

Sus mayores ventajas son que permite conexiones con la mayoría de base de datos, entre las que destaca MySQL, que es libre y gratuito, y de la cual hay una gran cantidad de documentación. Además se cuenta con la ventaja de que es un lenguaje conocido por el desarrollador y esto ahorraría esfuerzos en aprender un nuevo lenguaje.

Se ha descartado la tecnología .NET por tener normalmente un coste mayor que el resto de tecnologías, y pese a que ya hay alternativas libres bastante buenas aun está demasiado sujeto a plataformas de pago. Una de sus ventajas es que soporta diversos lenguajes de desarrollo.

También se ha descartado Java por qué no era necesaria tanta potencia y complejidad para el desarrollo de este proyecto y tiene el inconveniente de que su curva de aprendizaje es demasiado elevada, aunque tiene a su favor que es multiplataforma y además es gratuito.

Una vez se ha seleccionado el lenguaje principal que se utilizará para desarrollar la aplicación, hay que mencionar que se utilizará conjuntamente con otros tres lenguajes muy habituales en aplicaciones web y que son XHTML, CSS y JavaScript y que se describirán a continuación. También se utilizará un framework escrito en JavaScript llamado JQuery que permitirá poner efectos visuales a la aplicación. Otra de las

herramientas utilizadas será PHPMailer que es una clase escrita en PHP que se encargará de enviar emails a los usuarios de la aplicación

XHTML

Se trata del estándar utilizado para las páginas web, con el que sólo se debe representar la estructura de la página, sin aplicar ninguna propiedad de estilo, ya que de esto se encargará el CSS.

CSS

Es el lenguaje utilizado para definir la presentación de las páginas escritas en XHTML. Este lenguaje permite tener todos los aspectos visuales en un solo archivo y así facilitar su posterior modificación, además de disminuir el peso total de la web.

Javascript

Es muy similar a Java y permite la interacción entre el usuario y la web. A diferencia de PHP, Javascript es un lenguaje que es interpretado por el navegador.

jQuery

jQuery [8] es un framework escrito en JavaScript que permite simplificar la manera de interactuar con los documentos HTML, además permite realizar animaciones en JavaScript de una forma rápida y concisa.

Este framework se ha utilizado para crear los menús y hacer diversos efectos visuales en el proyecto.

DataTables

DataTables [9] es un plug-in para la librería JQuery. Esta aplicación permite crear tablas más atractivas utilizando las mismas herramientas que JQuery.

Esta aplicación se ha utilizado para mostrar las consultas a la base de datos en tablas con paginación, y así hacer más agradable el visionado de los datos.

jQuery Tooltip

jQuery Tooltip es una librería JavaScript que extiende las propiedades de JQuery y la cual da la posibilidad de personalizar los tooltips mediante CSS.

Esta librería se ha utilizado para mostrar ventanas de información en las páginas, además también se ha utilizado para mostrar la información conveniente de los genotipos.

PHPMailer

PHPMailer es una clase escrita en PHP que facilita el envío de correos, añadiendo facilidad en el envío de correos adjuntos, en formato HTML y con diferentes codificaciones. Además una de las cualidades que más interés presenta es que funciona con múltiples servidores de correo.



Esta clase se utilizará para el envío de correos entre los usuarios y para el envío del correo de bienvenida de la aplicación.

Perl

Perl es un lenguaje para manipular textos, ficheros y procesos. Este lenguaje proporciona una forma fácil y legible para realizar trabajos que normalmente se desarrollarían en C o en algún Shell. Podría decirse que Perl está a caballo entre un lenguaje de alto nivel (tipo C) y un 'Commands shell'. Es un lenguaje muy utilizado para la elaboración de aplicaciones CGI.

4.3 Base de datos

PostgreSQL

Posee una gran escalabilidad. Es capaz de ajustarse al número de CPU's y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneamente de manera correcta. Sus inconvenientes son que consume gran cantidad de recursos, es de 2 a 3 veces más lento que MySQL.

Oracle

Uno de los mejores gestores de base de datos ya que es de los que cumplen más condiciones del estándar SQL. Ha sido descartado por ser más lento que MySQL y porque el proyecto se decidió desarrollar con software libre y este es software propietario.

MySQL

El sistema gestor de base de datos elegido es MySQL [10], que al igual que PHP, es libre, gratuito y multiplataforma. Su integración con PHP es buena, y hasta ahora han sido el conjunto más utilizado para el desarrollo de páginas web. Es uno de los gestores más rápidos, y esa ha sido una de las razones por las que se ha seleccionado este gestor.



PHPMyAdmin

Como se ha seleccionado el gestor de base de datos MySQL se ha optado por esta herramienta que permite realizar las operaciones con la base de datos MySQL desde el navegador. Se trata de una aplicación escrita en PHP, y que por tanto necesita un servidor web con PHP instalado para funcionar.

4.4 Apache

Se necesita un servidor web para interpretar las páginas PHP y que proporcione el código HTML al navegador, en este caso se ha utilizado Apache que, al igual que PHP y MySQL, es libre, gratuito y multiplataforma.

4.5 Notepad ++

Se trata de un editor de texto bastante potente, excelente para programación gracias a su rapidez y su reconocimiento de lenguajes. Se ha elegido este editor por consumir pocos recursos.

4.6 R

R [13] es un lenguaje de programación orientado a objetos y destinado al análisis estadístico y la representación de datos. R se puede considerar como una implementación diferente de S. Hay algunas diferencias importantes, pero mucho del código escrito para S puede ser ejecutado bajo R.



Se ha elegido este lenguaje por ser cada vez más utilizado en la comunidad bioinformática.

4.7 mkInstaller

MkInstaller [12] es una aplicación que sirve para crear un instalador del sitio web y facilita la distribución del mismo. Además permite definir diferentes variables necesarias para la puesta en marcha de la aplicación.

5. Diseño de la aplicación

5.1 Introducción

En este capítulo se van a utilizar diferentes técnicas y principios con el propósito de definir un sistema con suficiente detalle como para poder realizarlo físicamente.

El objetivo principal es producir un modelo o representación del sistema que pueda ser utilizado en la fase de implementación. Por tanto se verá tanto el diseño de la interfaz gráfica como el diseño de la base de datos a partir de los datos proporcionados.

5.2 Modelo de datos

En esta fase se transformará la información proporcionada, que ha sido mostrada en el capítulo de análisis, en la estructura de datos necesaria para la implementación de la base de datos. Este diseño previo será vital para un buen funcionamiento de la aplicación y para que el usuario final tenga una experiencia positiva al utilizarla.

5.2.1 Diseño del modelo de datos

Después del primer análisis se llegó a la conclusión que el modelo de datos tenía que ser algo parecido a la Figura 11. Como se puede observar se tiene el grupo de tablas llamado “Datos”, que es donde se guardará la información obtenida en las distintas investigaciones. Esta tabla ha surgido de la fusión de las tablas de fenotipos y genotipos mostradas en el capítulo de análisis y por lo tanto guardará toda la información referente a ellos. También se pueden observar el grupo de tablas “Datos Individuo” e “Individuo” que guardarán la información obtenida/medida de cada uno de los especímenes estudiados, en el primer grupo, y la información común del individuo, en el segundo.

Otro grupo de tablas es el de “Tipo de datos”, será en este donde se guardarán los diferentes tipos de datos utilizados en la aplicación, como pueden ser edad, peso, microsatélites, y que tendrán que ser definidos por el usuario para poder comenzar a utilizar la aplicación.

También se tiene el grupo de tablas “Experimentos” que es donde se guardará la información referente a los experimentos, como puede ser los individuos que pertenecen a él, o una descripción por ejemplo.

Junto con el grupo de experimentos se tiene el grupo de “Lotes”, que son las agrupaciones de observaciones realizadas durante el mismo período de tiempo.

Además también está la parte de Investigadores que es donde se guardará la información referente a estos.

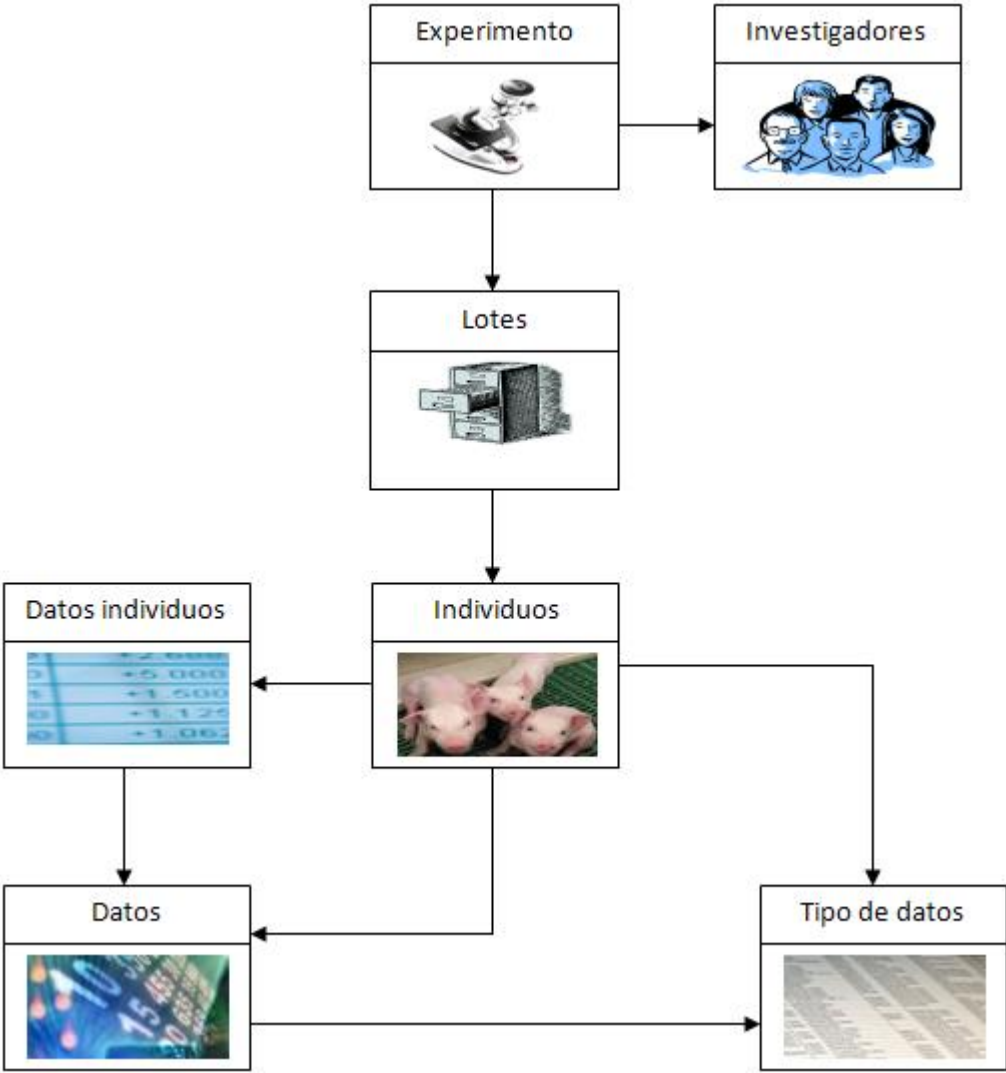


Figura 11: Tipo de datos.

5.3 Interfaz gráfica

Para la interfaz gráfica se ha utilizado un diseño de Free CSS Templates. Estos diseños tienen licencia Creative Commons 2.5 que permite su uso tanto comercial como no comercial y también permite su modificación por parte de los desarrolladores. Estos diseños son 100% CSS, para poder intercambiarlos, si se desea, de forma rápida.

5.3.1 Estructura de la página

En la Figura 12 se puede ver la cual será la estructura de la interfaz. Como se puede apreciar las páginas se dividen en cuatro zonas. La cabecera, donde se mostrará una imagen de la facultad de veterinaria e informará del departamento para el cual ha sido realizado este proyecto. El pie de página que informará que la aplicación cumple con el estándar XHTML. La zona de la izquierda será donde se muestre el menú con las distintas opciones que habrá disponibles para los usuarios de la aplicación. Y por último la zona central que será donde se muestre toda la información al usuario.

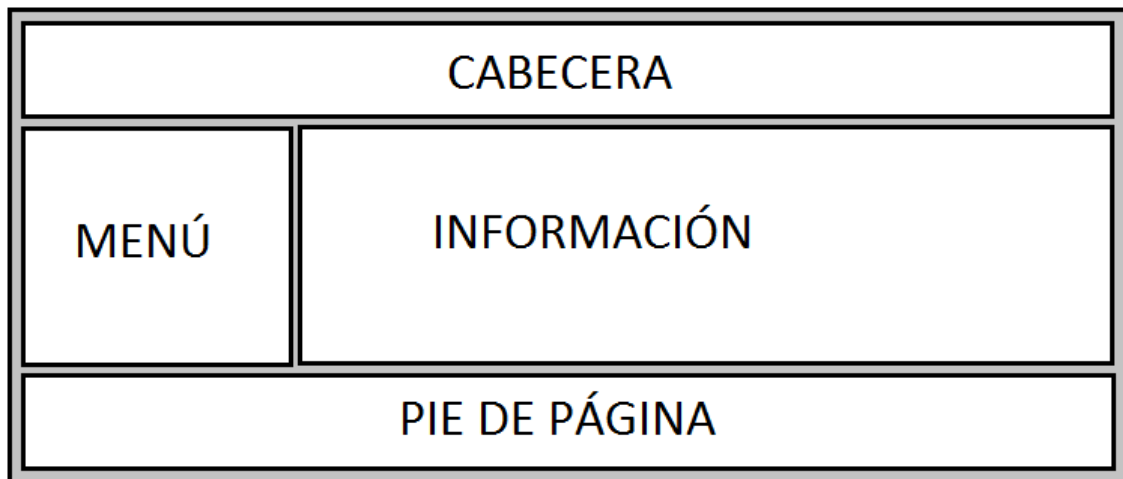


Figura 12: Página principal de la aplicación.

5.3.2 Menú de la aplicación

5.3.2.1 Menú del Investigador

Como se puede apreciar en la Figura 13 el menú será en forma de acordeón, con lo que se nos mostrarán las acciones una vez se ha pulsado sobre una cabecera del mismo. Esta forma de menú es práctica ya que solo se muestran las opciones de la cabecera elegida mientras el resto permanece oculto hasta que es requerido, con lo que ocupa menos espacio que un menú convencional.

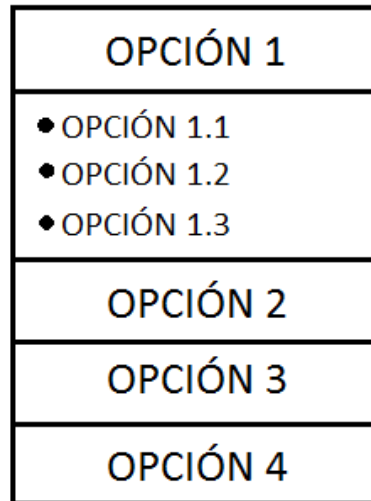


Figura 13: Menú de la aplicación.

5.3.2.2 Menú del administrador

El menú del administrador tiene las mismas opciones que el menú de un usuario no administrador, con la única diferencia de que se posee una zona más que es donde estarán las acciones de administración de la aplicación. En la Figura 14 se puede apreciar como quedaría el menú del administrador.

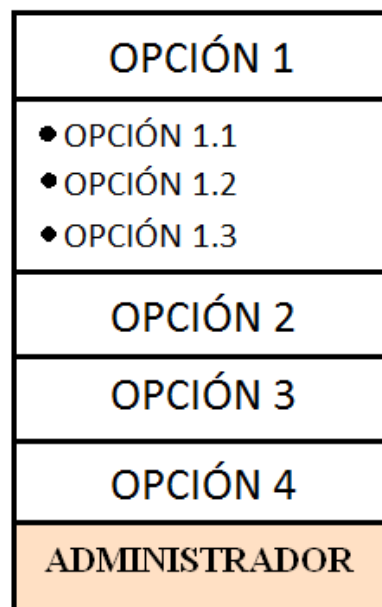


Figura 14: Menú del administrador.

5.3.3 Zona de información

En la zona de información es donde trabajará el usuario con los datos almacenados en la base de datos. Es donde se le mostrarán los formularios, las consultas y los resúmenes. En la Figura 15 se puede apreciar cómo se mostrará un formulario en esta zona central.

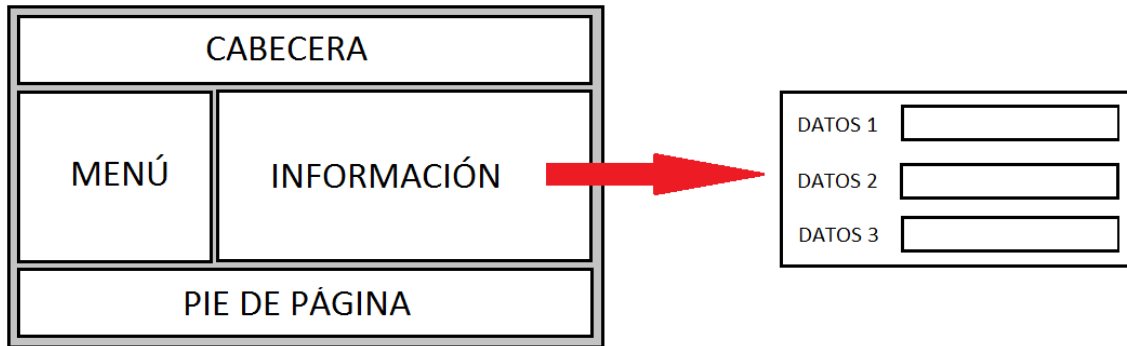


Figura 15: Formulario en la zona de información.

5.3.4 Botones

Dependiendo de la actividad que se elija se mostrarán unos botones u otros, en la Figura 16 se muestran los botones que se pueden encontrar dentro de la aplicación:



Figura 16: Ejemplos de botones dentro de la aplicación.

A continuación se explicará para que se utilizará cada uno de ellos:

- El icono 1 se mostrará cuando se tenga que guardar información en la base de datos.
- El icono 2 surgirá cuando se vaya a modificar información de la base de datos.
- El icono 3 aparecerá cuando se vaya a eliminar información.
- El icono 4 saldrá cuando después del formulario hay una nueva página. Indica que hay más páginas por rellenar.
- El icono 5 indicará que se podrá descargar la información a un archivo.

- El icono 6 significará que se podrá descargar el resumen de las operaciones.
- El icono 7 aparecerá cuando se pretenda subir un archivo a la aplicación.
- El icono 8 se mostrará cuando se quiera eliminar un proyecto.
- El icono 9 se podrá utilizar cuando se pretenda dar de alta un nuevo proyecto.
- El icono 10 servirá para añadir un nuevo usuario a un proyecto.
- El icono 11 se deberá utilizar para dar de baja a un usuario del proyecto.
- El icono 12 es el icono de información y por lo tanto aparecerá en todas las páginas de la aplicación.
- El icono 13 aparecerá cuando se genere una nueva contraseña.
- El icono 14 servirá para salir de la aplicación.

5.3.5 Breadcrumb

El Breadcrumb mostrará el camino que se ha seguido hasta llegar a la página actual. Este se basa en la navegación paso a paso, con lo que si se quiere volver a una página anterior solo se tendrá que pulsar sobre el paso correspondiente y se accederá a ella. En la Figura 17 se puede apreciar cómo se diseñará el breadcrumb.



Figura 17: Ejemplo de Breadcrumb

5.4 Base de datos

La base de datos es una de las partes más importante del proyecto ya que es en ella donde se almacenará toda la información utilizada por la aplicación. Y además de la base de datos dependerá el buen funcionamiento del programa.

5.4.1 Diagrama

Como se puede ver en la Figura 18 tenemos la base de datos diseñada para este proyecto a partir del modelo de datos, como se puede observar está dividida en diferentes apartados que se describirán a continuación:

- (A) En estas tablas es donde se guardará la información de los diferentes experimentos, así como los diferentes investigadores que han participado en ellos.
- (B) Aquí se guardará la información de los investigadores y las preferencias que han seleccionado en la aplicación.
- (C) En esta parte es donde se guardará la información de los individuos. En C1 y C3 se guardarán los Fenotipos y Genotipos respectivamente y en C2 se guardará la información común de los individuos, como por ejemplo el padre y madre de un individuo.
- (D) En D1 es donde se guardará el tipo de datos que se utilizará en la aplicación. En D2 se almacenarán los datos de cada individuo dependiendo del tipo al que pertenezcan, entero, coma flotante o cadena.
- (E) En esta tabla se guardará los datos sobre el investigador que ha modificado la base de datos.
- (F) Esta tabla guardará la versión de la base de datos actual y guardará el usuario que ha realizado la modificación.
- (G) Esta parte guardará las sesiones del investigador así como las posibles sesiones que se le envíen por parte de otros usuarios.
- (H) Aquí se guardarán los mensajes internos que se envíen entre los investigadores.

Se ha optado por esta forma de diseñar la base de datos, porque se pretendía crear una aplicación que soportara cambios en los individuos y los datos obtenidos de ellos, por ejemplo, cambiar la información de animales por la de vegetales, o añadir un nuevo fenotipo a los ya existentes. Esta funcionalidad de la base de datos se consigue gracias a que se ha implementado una tabla con los tipos de datos utilizados ("TipoDatos"), de esta manera si se necesitara un nuevo dato solo se tendría que insertar en esta tabla y ya estaría disponible en la aplicación.

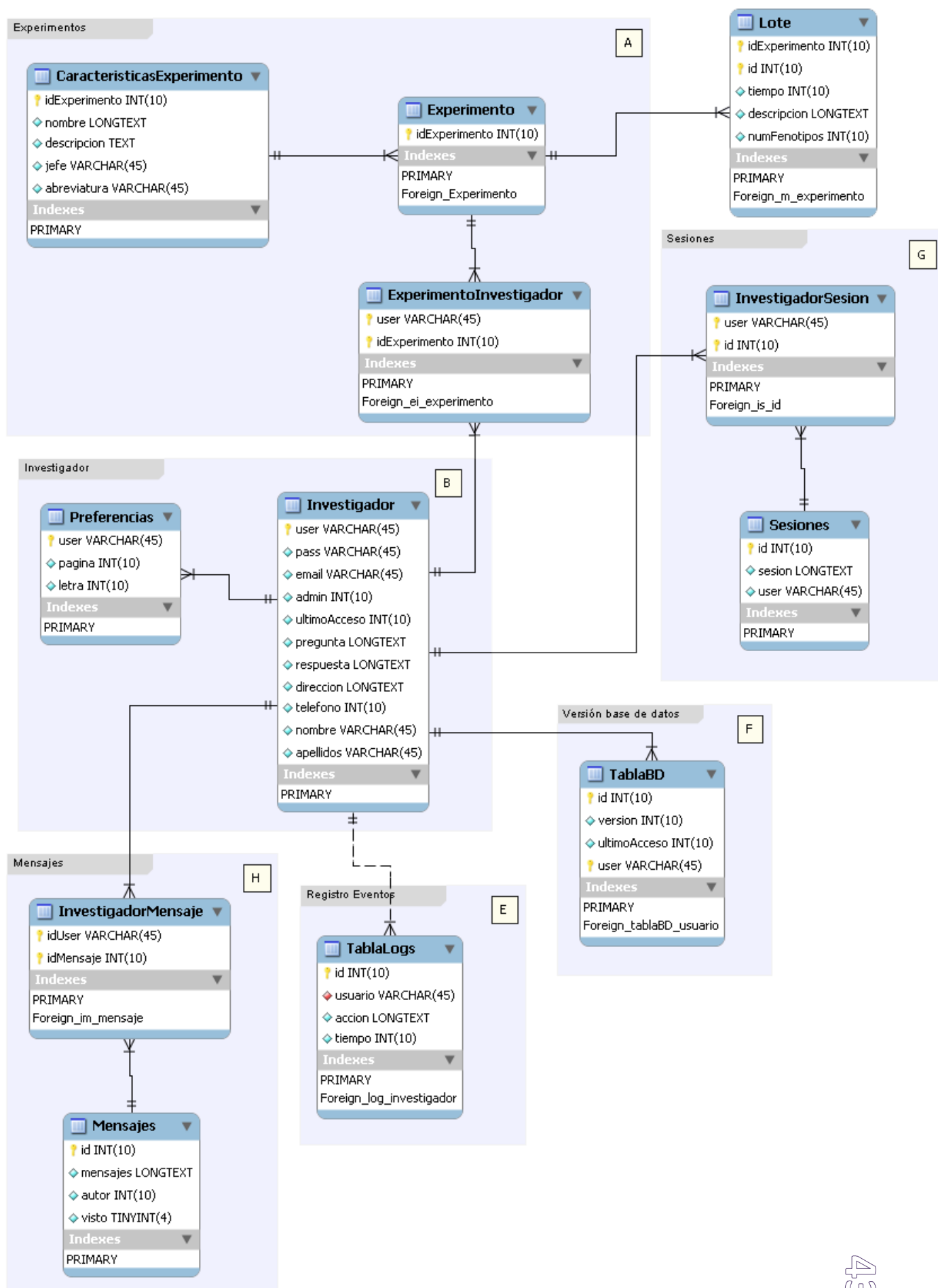


Figura 18 (a): Diagrama de las tablas de la base de datos

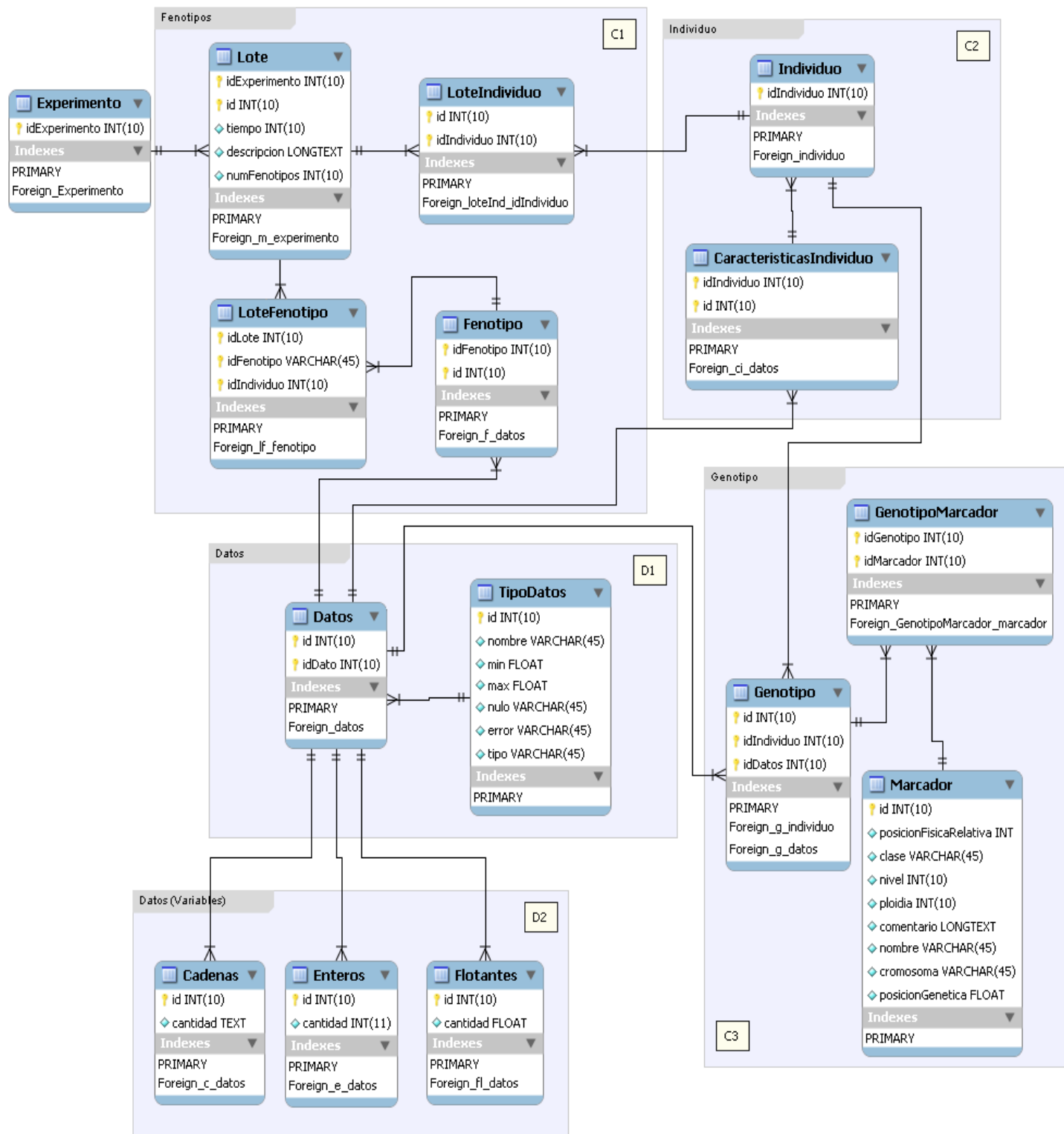


Figura 18 (b): Diagrama de las tablas de la base de datos

5.4.2 Estructura

En este apartado se mostrará con más detalle las diferentes tablas que forman parte de la base de datos y se explicarán cada uno de sus campos y la función que tienen dentro de esta.

Tabla Investigador: Esta tabla contiene toda la información necesaria para identificar a un usuario de la aplicación, además también sirve para indicar si dicho usuario tiene permisos de administrador.

Campo	Tipo	Descripción
<i>user</i>	varchar	Clave primaria. Este campo contiene el identificador de los investigadores.
<i>pass</i>	varchar	Contraseña del usuario.
<i>email</i>	varchar	Contiene la dirección de correo electrónico del usuario.
<i>admin</i>	int	Indica si el usuario es administrador (1) o usuario (0).
<i>ultimoAcceso</i>	int	Guarda la fecha del último acceso del usuario.
<i>pregunta</i>	longtext	Guarda la pregunta que se realizará en caso de perder la contraseña.
<i>respuesta</i>	longtext	Contiene la respuesta asociada a la pregunta.
<i>direccion</i>	longtext	Contiene la dirección del usuario.
<i>telefono</i>	int	Guarda el teléfono del usuario.
<i>nombre</i>	varchar	Este campo guarda el nombre del usuario.
<i>apellidos</i>	varchar	Aquí se guardan los apellidos.

Tabla Mensajes: Contiene los mensajes internos enviados entre los investigadores.

Campo	Tipo	Descripción
<i>id</i>	int-auto incremental	Clave primaria. Identificador de cada uno de los mensajes.
<i>mensajes</i>	longtext	Contiene el mensaje escrito por el usuario.
<i>autor</i>	int	Identifica al usuario que ha escrito el mensaje.
<i>visto</i>	boolean	Indica si el usuario ha visto el mensaje.

Tabla InvestigadorMensaje: Tabla que une al usuario con sus mensajes.

Campo	Tipo	Descripción
<i>idUser</i>	varchar	Clave primaria y foránea. Contiene al usuario que tiene un mensaje.
<i>idMensaje</i>	int	Clave primaria y foránea. Identificador del mensaje.

Tabla Sesiones: Contiene las sesiones guardadas por el usuario y las sesiones que le han enviado otros usuarios.

Campo	Tipo	Descripción
<i>id</i>	int	Clave primaria. Identificador de las sesiones guardadas en la aplicación.
<i>sesion</i>	longtext	Contiene la sesión guardada.
<i>user</i>	varchar	Usuario que ha enviado o guardado la sesión.

Tabla InvestigadorSesion: Tabla que une las sesiones con los investigadores.

Campo	Tipo	Descripción
<i>user</i>	varchar	Clave primaria y foránea. Contiene el usuario al cual pertenece la sesión.
<i>id</i>	int	Clave primaria y foránea. Contiene el identificador de la sesión.

Tabla Preferencias: Esta tabla guarda las preferencias que el usuario ha elegido en la aplicación.

Campo	Tipo	Descripción
<i>user</i>	varchar	Clave primaria y foránea. Contiene el usuario que ha elegido sus preferencias.
<i>pagina</i>	int	Contiene las preferencias elegidas para la paginación de los resultados.
<i>letra</i>	int	Contiene el tamaño de la letra de la aplicación.

Tabla CaracteristicasExperimento: Guarda toda la información acerca de los proyectos.

Campo	Tipo	Descripción
<i>idExperimento</i>	int-auto incremental	Clave primaria. Contiene el identificador del experimento.
<i>nombre</i>	longtext	Guarda el nombre del experimento.
<i>descripcion</i>	text	Este campo contiene una pequeña descripción de lo que se hace en el experimento.
<i>jefe</i>	varchar	Identifica al jefe del proyecto.
<i>abreviatura</i>	varchar	Contiene la abreviatura del nombre del proyecto.

Tabla Experimento: Guarda todos los identificadores de los experimentos.

Campo	Tipo	Descripción
<i>idExperimento</i>	int	Clave primaria y foránea. Contiene los identificadores de los experimentos.

Tabla ExperimentoInvestigador: Guarda los usuarios que están inscritos en un experimento.

Campo	Tipo	Descripción
<i>user</i>	varchar	Clave primaria y foránea. Este campo guarda el usuario.
<i>idExperimento</i>	int	Clave primaria y foránea. Contiene el identificador del experimento.

Tabla TipoDatos: Contiene todos los datos que se utilizarán en la aplicación.

Campo	Tipo	Descripción
<i>id</i>	int-auto incremental	Clave primaria. Identificador de los datos.
<i>nombre</i>	varchar	Nombre de los datos.
<i>min</i>	float	Valor mínimo que puede tener el dato. En caso de que el dato sea una cadena este campo será null.
<i>max</i>	float	Valor máximo que puede tener el dato. En caso de que el dato sea una cadena este campo será null.
<i>nulo</i>	varchar	Valor que tendrá el dato si no tiene un valor concreto.
<i>error</i>	varchar	Valor que tendrá el dato si es un valor erróneo.
<i>tipo</i>	varchar	Nos dice de qué tipo es el dato (Entero, Float o Cadena).

Tabla Datos: Esta tabla sirve para unir las tablas que contienen los valores de los datos con la tabla TipoDatos y con las tablas de Individuo, Fenotipo y Genotipo.

Campo	Tipo	Descripción
<i>id</i>	int-auto incremental	Clave primaria. Contiene el identificador de los datos.
<i>idDato</i>	int	Clave primaria y foránea. Guarda el identificador del tipo de dato.

Tabla Individuo: Esta tabla guarda los identificadores de los individuos.

Campo	Tipo	Descripción
<i>idIndividuo</i>	int	Clave primaria y foránea. Contiene los identificadores de los individuos.

Tabla CaracteristicasIndividuo: Esta tabla une a cada individuo con sus datos.

Campo	Tipo	Descripción
<i>idIndividuo</i>	int-auto incremental	Clave primaria. Guarda el identificador del individuo.
<i>id</i>	int	Clave primaria y foránea. Contiene el identificador de los datos.

Tabla Lote: Guarda la información sobre los lotes de los fenotipos.

Campo	Tipo	Descripción
<i>idExperimento</i>	int	Clave primaria y foránea. Contiene el identificador del experimento.
<i>id</i>	int-auto incremental	Clave primaria. Contiene el identificador del lote.
<i>tiempo</i>	int	Guarda el período en el cual se ha realizado las medidas del lote.
<i>descripcion</i>	longtext	Contiene una pequeña descripción del lote.
<i>numFenotipos</i>	int	Contiene el número de fenotipos que tiene este lote.

Tabla Fenotipo: Esta tabla une cada muestra con los datos de los fenotipos que pertenecen a ella.

Campo	Tipo	Descripción
<i>idFenotipo</i>	int	Clave primaria. Contiene el identificador del fenotipo.
<i>id</i>	int	Clave primaria y foránea. Contiene el identificador de los datos.

Tabla LoteFenotipo: Esta tabla une los lotes con los fenotipos.

Campo	Tipo	Descripción
<i>idLote</i>	int	Clave primaria y foránea. Contiene el identificador del lote.
<i>idFenotipo</i>	varchar	Clave primaria. Contiene el identificador del fenotipo.
<i>idIndividuo</i>	int	Clave primaria. Contiene el identificador del individuo al cual pertenece el fenotipo.

Tabla LoteIndividuo: Esta tabla une los lotes con los individuos.

Campo	Tipo	Descripción
<i>id</i>	int	Clave primaria y foránea. Contiene el identificador del lote.
<i>idIndividuo</i>	int	Clave primaria y foránea. Contiene el identificador del individuo.

Tabla Genotipo: Esta tabla une cada genotipo con los datos y el individuo al que pertenece.

Campo	Tipo	Descripción
<i>id</i>	int-auto incremental	Clave primaria. Contiene el identificador del Genotipo.
<i>idIndividuo</i>	int	Clave primaria y foránea. Contiene el identificador del individuo.
<i>idDatos</i>	int	Clave primaria y foránea. Contiene el identificador de los datos.

Tabla Marcador: Esta tabla guarda toda la información que es fija para el genotipo.

Campo	Tipo	Descripción
<i>id</i>	int-auto incremental	Clave primaria. Identificador del marcador.
<i>posicionFisicaRelativa</i>	int	Guarda la posición física del marcador.
<i>clase</i>	varchar	Contiene el tipo al que pertenece el marcador.
<i>nivel</i>	int	Guarda el número de valores que puede tomar el marcador.
<i>ploidia</i>	int	Contiene la cantidad de muestras de un marcador que tiene la tabla genotipos.
<i>comentario</i>	longtext	Contiene un pequeño comentario sobre el marcador.
<i>nombre</i>	varchar	Contiene el nombre del marcador.
<i>cromosoma</i>	varchar	Contiene el cromosoma de este marcador.
<i>posicionGenetica</i>	float	Guarda la posición genética del marcador.

Tabla GenotipoMarcador: Esta tabla une los genotipos con la tabla marcadores.

Campo	Tipo	Descripción
<i>idGenotipo</i>	int	Clave primaria y foránea. Contiene el identificador del genotipo.
<i>idMarcador</i>	int	Clave primaria y foránea. Contiene el identificador del marcador.

Tabla Enteros: Esta tabla guarda los datos que son enteros.

Campo	Tipo	Descripción
<i>id</i>	int	Clave primaria y foránea. Contiene el identificador del dato.
<i>cantidad</i>	int	Este campo guarda el valor del dato.

Tabla Flotantes: Esta tabla guarda los datos de tipo coma flotante.

Campo	Tipo	Descripción
<i>id</i>	int	Clave primaria y foránea. Contiene el identificador del dato.
<i>cantidad</i>	float	Este campo guarda el valor del dato.

Tabla Cadenas: Esta tabla guarda los datos que son del tipo cadena.

Campo	Tipo	Descripción
<i>id</i>	int	Clave primaria y foránea. Contiene el identificador del dato.
<i>cantidad</i>	text	Este campo guarda el valor del dato.

Tabla TablaLogs: En esta tabla se guardan los datos sobre el investigador que ha modificado la base de datos y el tipo de modificación que ha realizado.

Campo	Tipo	Descripción
<i>id</i>	int-auto incremental	Clave primaria. Este campo es el identificador de la información introducida en la tabla.
<i>usuario</i>	varchar	Contiene el identificador del usuario que ha producido un log.
<i>accion</i>	longtext	Contiene la descripción del log.
<i>tiempo</i>	int	Contiene la fecha en la que se ha producido el log.

Tabla TablaBD: En esta tabla se guarda la información acerca de la versión de la base de datos que se está utilizando en la aplicación, además también se contiene información referente a cuando se produjo esta versión y quien la realizo.

Campo	Tipo	Descripción
<i>id</i>	int-auto incremental	Clave primaria. Este campo es el identificador de la información introducida en la tabla.
<i>version</i>	int	Contiene la versión de la base de datos.
<i>ultimoAcceso</i>	int	Contiene la fecha y hora en la que se cambió de base de datos.
<i>user</i>	varchar	Clave primaria y foránea. Contiene al usuario que realizó el cambio.

5.5 Planificación

A continuación se describen las variaciones que ha sufrido la planificación una vez se ha avanzado en el proyecto. Como podemos ver ha surgido nuevas tareas y la duración de otras a cambiado, pero el esqueleto inicial se mantiene intacto.

Fases del proyecto

Descripción de las tareas	Duración
Estudio previo del proyecto	6 h
Entrevistas con el tutor	1 h
Estudio previo del proyecto	5 h
Estudio de viabilidad	45 h
Entrevistas con el tutor	10 h
Estudio de viabilidad	35 h
Diseño de la base de datos	40 h
Estudio de la información proporcionada	10 h
Aportaciones del profesor de la unidad genética	10 h
Diseño de la base de datos	20 h
Diseño de la interfaz de usuario	40 h
Entrevistas con tutor y con profesor de unidad de genética	10 h
Diseño de la interfaz de usuario	30 h
Programación de la aplicación	120 h
Programación de la base de datos	2 h
Programación de la interfaz de usuario	50 h
Programación de parser para la subida de archivos Excel	25 h
Programación de un script para la comunicación entre la aplicación y R	10 h
Programación de hojas de estilo	23 h
Comprobación de cumplimiento de estándares	10 h
Pruebas	35 h
Elaboración de la documentación	50 h
Total	336 h

La única tarea que ha sido modificada son las entrevistas con el profesor de la Unidad de genética y mejora y con el tutor del proyecto. Esta tarea se ha puesto como subtareas de otras ya que se considera que las entrevistas se realizarán durante todo el proyecto.

La fecha de entrega no se verá seriamente afectada ya que se tenía un margen de demora para posibles imprevistos y además se ha podido dedicar más tiempo y por consiguiente se compensará el retraso con una dedicación mayor.

En la Figura 19 se puede observar el diagrama de Gantt del proyecto, con las nuevas estimaciones de la duración de cada una de las tareas una vez asignados los recursos.

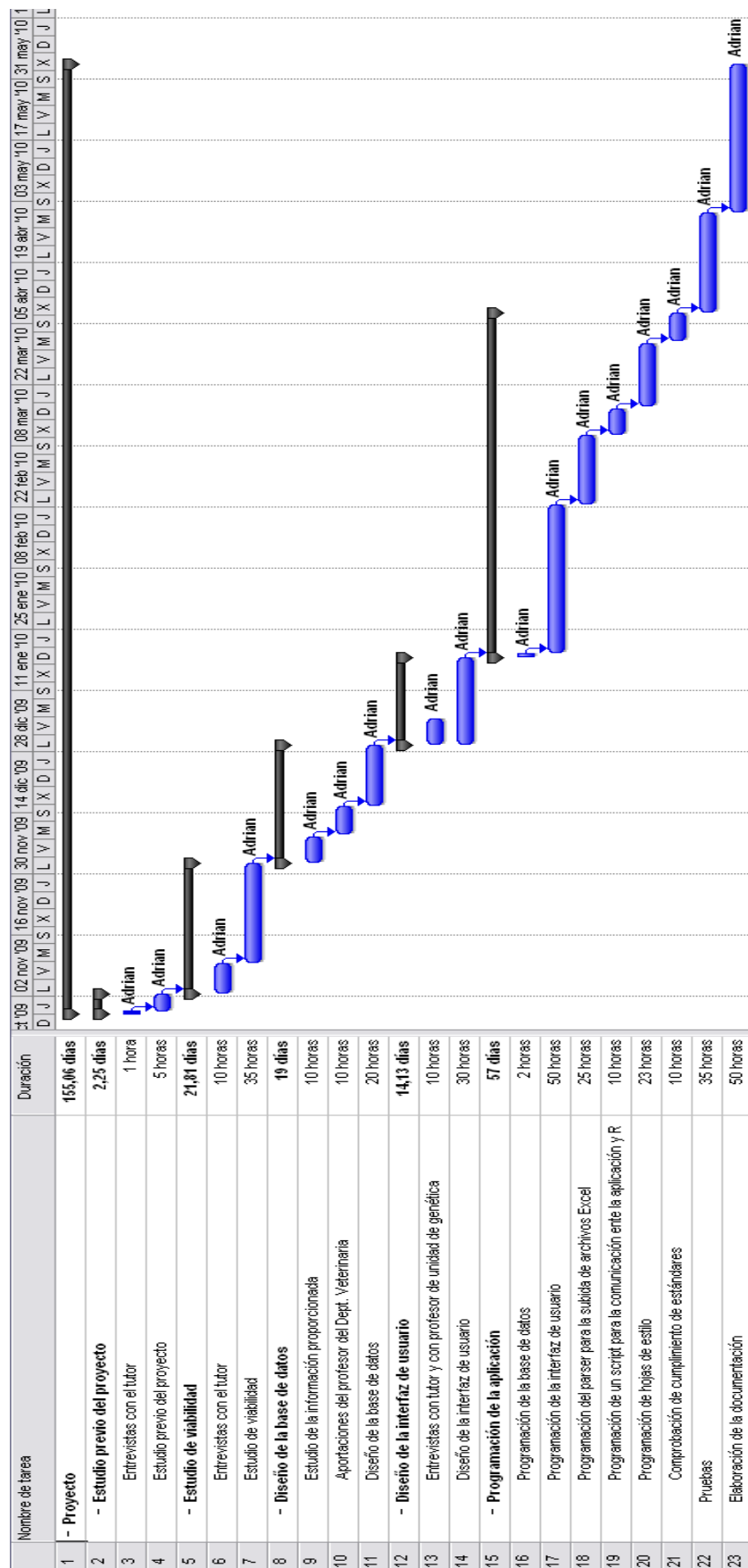


Figura 19: Diagrama de Gantt

6. Implementación

En este apartado se verá el estilo de codificación, las clases y variables utilizadas y algunos ejemplos de código.

En resumen lo que se pretende es transportar el diseño definido en apartados anteriores a un código escrito, en este caso se ha seleccionado PHP.

6.1 Implementación de la interfaz

6.1.1 Implementación de la estructura de la página

Como se puede ver en la Figura 20 se conserva la estructura de la página explicada en el capítulo de diseño. En la cabecera se ha colocado una imagen de la facultad de veterinaria y se ha indicado el departamento al que pertenecen los futuros usuarios de la aplicación. En el pie de página se ha añadido un link para comprobar que la página cumple con el estándar de XHTML. En la zona izquierda se puede observar cómo se ha implementado el menú de la aplicación. Y por último en la zona central será donde se mostrarán los datos que se han insertado en la aplicación.

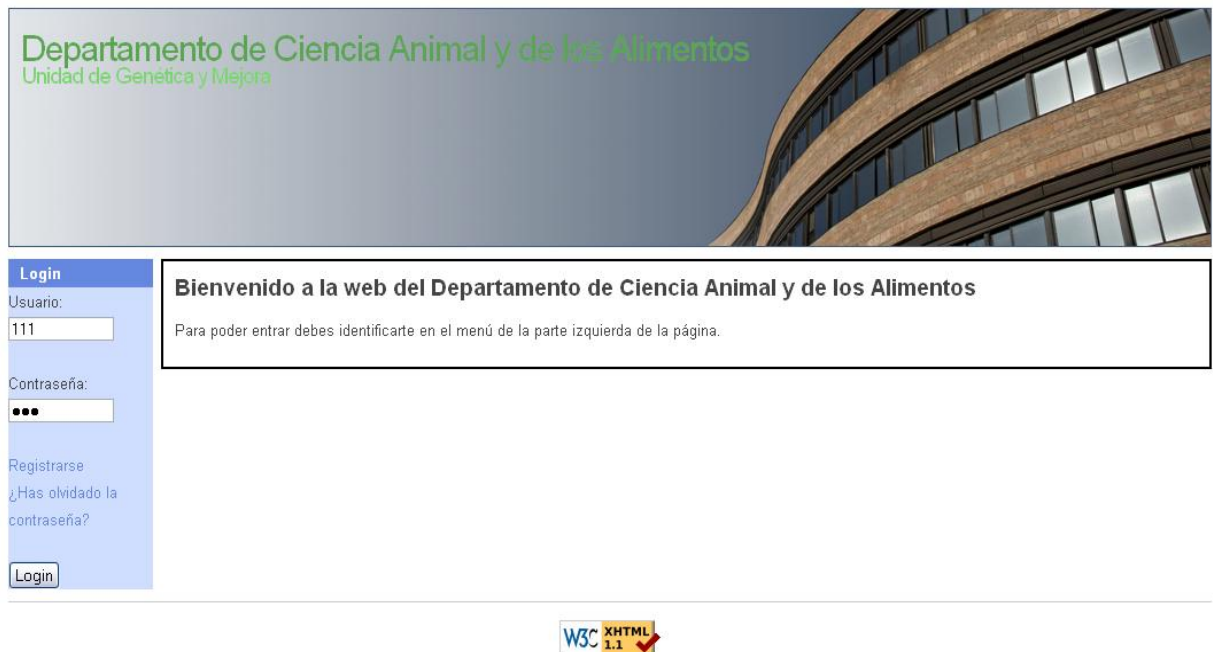


Figura 20: Implementación de la estructura de la página.

6.1.2 Menú de la aplicación

6.1.2.1 Menú del Investigador

En la Figura 21 se puede observar el menú de los investigadores que se ha implementado para la aplicación. Como se puede ver es en forma de acordeón y se divide en diferentes apartados que se explicarán a continuación.

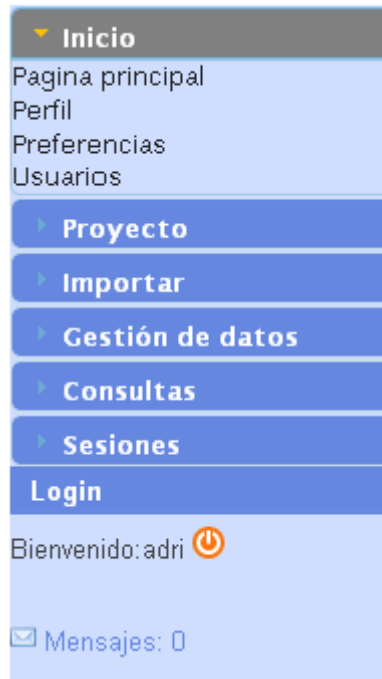


Figura 21: Menú de los investigadores

- Menú Inicio: En este menú se podrá ir a la página principal de la aplicación. Además se podrá cambiar el perfil y las preferencias de los usuarios. También se podrá enviar mensajes al resto de personas registradas en la aplicación.
- Menú Proyecto: Aquí se podrá dar de alta un nuevo proyecto y se podrá modificar o eliminar un proyecto ya existente.
- Menú Importar: En este menú se podrán subir los archivos con la información a insertar en la aplicación.
- Menú Gestión de datos: En esta zona se podrán insertar los datos a mano o se podrán modificar o eliminar los datos ya existentes.
- Menú Consultas: Se podrán consultar los tipos de datos, la información de los individuos así como sus fenotipos y genotipos.
- Menú Sesiones: En este menú se podrán guardar o recuperar las consultas realizadas. También se podrán enviar al resto de los usuarios o recuperar las consultas que nos envíen.

6.1.2.2 Menú del administrador

En la Figura 22 se presenta el menú del administrador. Como se puede observar todas las zonas son idénticas a las del menú del investigador, excepto la última que es exclusiva del administrador, y le ayudara a gestionar la aplicación.

Este último menú se podrá descargar una copia de seguridad de la base de datos y restaurar una copia de la versión anterior. También se podrá dar de alta a un nuevo usuario y eliminar a uno existente. Otra de las opciones será ver los registros que ha producido el uso de la aplicación por parte de los usuarios.

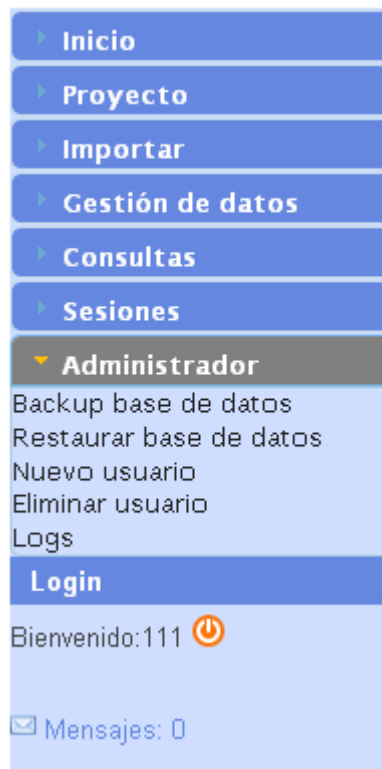
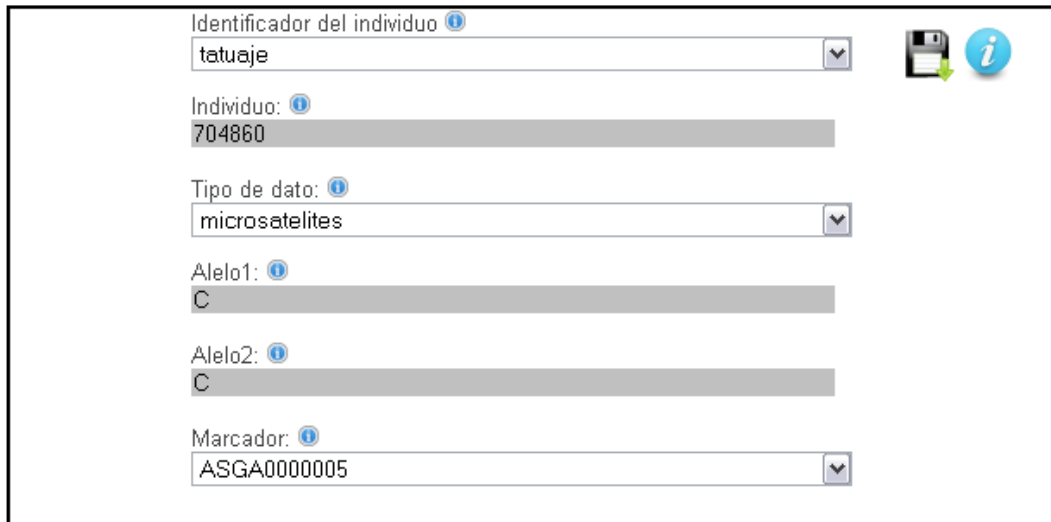


Figura 22: Implementación del menú del administrador

6.1.3 Zona de información

En la Figura 23 se puede ver un ejemplo de lo que aparecerá en la zona central de la aplicación. Esta zona presentará la información que hay almacenada en la base de datos y nos mostrará distintos formularios para interactuar con ella.



The screenshot shows a web form with the following fields and values:

- Identificador del individuo: tatuaje
- Individuo: 704860
- Tipo de dato: microsatelites
- Alelo1: C
- Alelo2: C
- Marcador: ASGA0000005

There are also icons for saving (floppy disk) and help (info 'i') in the top right corner.

Figura 23: Ejemplo de información presentada en la zona de información.

6.2 Estructura de la aplicación

En la Figura 24 se puede apreciar la estructura que tiene la aplicación. Como se puede apreciar se divide en diferentes módulos que serán expuestos a continuación.

- Módulo Página principal: Es la página inicial y es desde donde se podrá acceder al resto de las acciones de la aplicación.
- Módulo Proyectos: En este módulo se podrán crear nuevos proyectos y eliminar y modificar los proyectos ya existentes.
- Módulo Mensajes: En este módulo los usuarios podrán enviar emails y mensajes internos al resto de personas que utilicen la aplicación.
- Módulo Consultas: En este módulo se podrán consultar los datos insertados por los usuarios en la aplicación. También se podrán filtrar para mostrar solo la información útil para la persona que lo esté consultando.
- Módulo Operaciones: En este módulo se podrán realizar operaciones con los datos consultados y filtrados.
- Módulo Importar Archivos: Aquí se seleccionarán los archivos a subir a la aplicación y se comprobará su integridad para posteriormente insertar la información en la base de datos.

- Módulo Insertar información a mano: En este módulo se podrán insertar nuevos datos rellorando el formulario que se presenta dentro de la aplicación.
- Módulo Modificar Información: Aquí se podrán modificar los datos existentes dentro de la aplicación. Para poder cambiarlos primero habrá que consultar los datos a modificar.
- Módulo Sesiones: En este módulo se podrán guardar las consultas realizadas por los usuarios así como recuperarlas. También se podrán enviar y recuperar las sesiones recibidas.
- Módulo Administrador: En este módulo el usuario administrador podrá gestionar la aplicación y dar de alta a nuevos usuarios.

A parte de estos módulos también existen dos carpetas dinámicas donde se guardarán las copias de seguridad de la base de datos. Estas copias se realizan cada vez que se insertan datos en la aplicación. La segunda carpeta será utilizada para guardar temporalmente los archivos subidos, las operaciones realizadas, etc. Esta carpeta será automática y periódicamente vaciada por la aplicación.

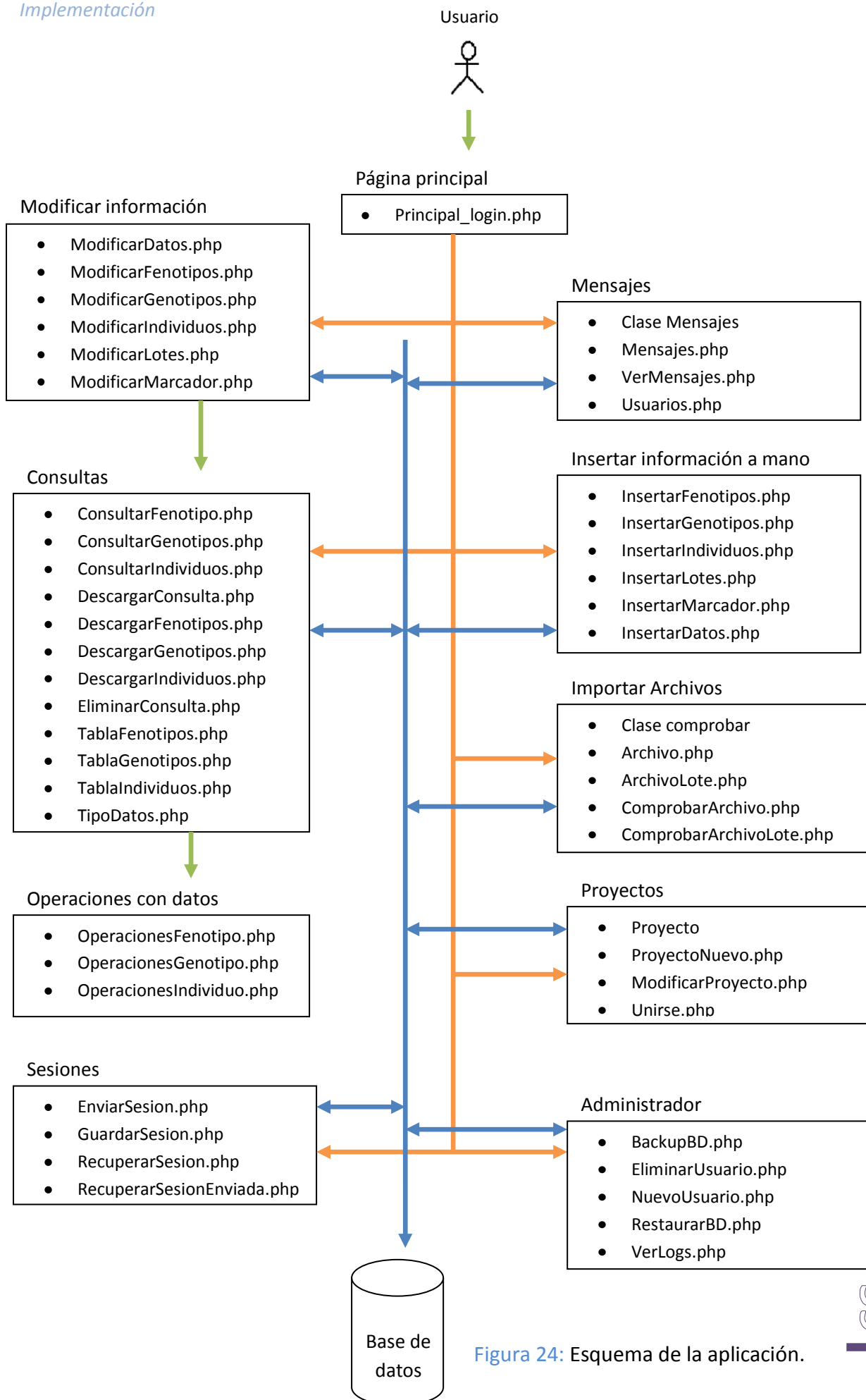


Figura 24: Esquema de la aplicación.

6.3 Codificación de las clases

A continuación se verá una breve descripción de las clases utilizadas en el proyecto y algunos ejemplos de su utilización.

Para realizar una conexión a la base de datos se utiliza la clase *Conectar*, que está situada en la carpeta Include. Esta clase se encarga de realizar la conexión y de realizar las consultas a la base de datos. También se encarga de cerrar las sesiones abiertas. A continuación se procederá a describir las funciones de esta clase:

- **conectar():** Esta función se encarga de establecer la conexión con la base de datos y de devolver el conector.
- **consultar():** Realiza las consultas a la base de datos.
- **cerrar_result():** Libera la memoria ocupada al realizar cierto tipo de consultas a la base de datos.
- **cerrar_sesion_bd():** Cierra la conexión con la base de datos.
- **getMydbname():** Devuelve el nombre de la base de datos utilizada.

En la Figura 25 se puede ver un ejemplo de uso de esta clase. Para comenzar se debe añadir la clase con un `require_once`, y a continuación crear una instancia a la clase. Una vez se ha instanciado, ya se pueden realizar las consultas oportunas. Por último se libera la memoria y se cierra la conexión con la base de datos.

```
1 <?php
2     require_once("conectar.php");
3     $BBDD = new BaseDatos();
4     $link = $BBDD->conectar();
5
6     $consulta = "SELECT * FROM Investigador";
7     $result = $BBDD->consultar($link, $consulta);
8     $numFilas = mysql_num_rows($result);
9     $BBDD->cerrar_result($result);
10    $BBDD->cerrar_sesion_bd($link);
11 ?>
```

Figura 25: Ejemplo de uso de la clase conectar.

También existe la clase *Mensajes*, que es la que será utilizada para enviar mensajes internos y emails entre los usuarios. Esta clase es importante ya que es la encargada de enviar a los nuevos usuarios el email comunicándoles que ya están registrados en la aplicación. A continuación se describirá las funciones que contiene esta clase:

- **MensajeInterno():** Esta función es la encargada de enviar los mensajes internos entre los usuarios.
- **Email():** Es la función encargada de enviar emails. En esta función se crea una instancia a la herramienta PHPMailer, que facilita el envío de correos.
- **emailBienvenida():** Esta función es la encargada de enviar los emails de bienvenida a los nuevos usuarios de la aplicación. Igual que en la función anterior es creada una instancia a la clase PHPMailer.

En la Figura 26 se puede ver un ejemplo de esta clase. Para poder utilizarla se debe declarar con un `require_once` y posteriormente crear una instancia de la misma. En la línea 4 se puede observar cómo se envía un mensaje interno entre dos usuarios y en la línea 5 un email entre los mismos usuarios.

```
1 <?php
2     require_once("../Include/Mensajes.php");
3     $sms = new Mensajes();
4     $sms->MensajeInterno("Esto es el mensaje", $usuarioEnvia, $usuarioRecibe);
5     $sms->Email("Esto es el mensaje", $usuarioEnvia, $usuarioRecibe);
6 ?>
```

Figura 26: Ejemplo de uso de la clase Mensajes.

Otra de las clases utilizadas, y posiblemente la más importante de ellas, es la clase *Comprobar*. Esta clase es la encargada de comprobar la integridad de los archivos subidos a la aplicación y también es la encargada de insertar la información contenida en estos ficheros en la base de datos. También se encarga de realizar la copia de seguridad de la base de datos antes de insertar la nueva información. Las funciones pertenecientes a esta clase son:

- **ComprobarDatos():** Es la encargada de comprobar que el archivo de *Tipos de datos* es correcto.
- **InsertarDatos():** Esta función es utilizada para insertar los tipos de datos en la base de datos.
- **ComprobarArchivo():** Esta función es la encargada de comprobar que tipo de archivo se está subiendo.
- **ComprobarDatosArchivo():** Comprueba que los tipos de datos utilizados en el fichero existen en la base de datos.
- **ComprobarIndividuo(), ComprobarFenotipo(), ComprobarGenotipo(), ComprobarMarcador(), ComprobarLote():** Se comprueba que la información de los archivos de individuos, fenotipos, genotipos, marcadores o lotes es correcta.

- **InsertarArchivo():** Esta función es la encargada de comprobar que tipo de archivo se está insertando.
- **InsertarIndividuo(), InsertarFenotipo(), InsertarGenotipo(), InsertarMarcador(), InsertarLote():** Estas funciones son las encargadas de insertar la información de los archivos en la base de datos.
- **buscarArray():** Busca coincidencias entre multiarrays.
- **comparaArray():** Compara dos arrays.
- **InsertarLog():** Es la encargada de insertar en la base de datos un nuevo registro de actividad dentro de la aplicación.
- **eliminarArchivo():** Esta función se encarga de eliminar el archivo del servidor una vez se ha comprobado e insertado.
- **tipoError():** Esta función contiene todos los posibles errores que pueden producirse al subir un nuevo archivo.
- **getNumRegistros():** Devuelve el número de filas que contiene el fichero subido.
- **__sleep(), __wakeup(), __construct():** Están definidos como métodos mágicos de PHP. **__sleep** es llamada antes de cualquier serialización, que es utilizada para enviar la clase por URL. Es la encargada de cerrar la conexión con la base de datos y de cerrar el apuntador con el archivo. Además devuelve una matriz con todas las variables que pueden ser serializadas. **__wakeup** es el inverso de **__sleep** y por tanto es llamada antes de cualquier unserialize. Esta función restablece la conexión con la base de datos y reabre el archivo. El método **__construct** es el constructor de la clase y se comporta como tal.
- **BackupBD():** Esta función crea una copia de seguridad de la base de datos antes de insertar cualquier información.

En la Figura 27 se puede ver un ejemplo de uso de la clase Comprobar. Como en casos anteriores lo primero es hacer un `require_once` y crear una instancia de la clase. En la línea 4 se puede observar como la función `ComprobarArchivo` devuelve si ha habido un error. En la línea 9 se insertan los datos con la función `InsertarArchivo`.

```
1 <?php
2     require_once("../Include/Comprobar.php");
3     $a = new Comprobar("archivo.csv");
4     $error = $a->ComprobarArchivo();
5     if($error == 0){
6         echo "Se ha producido un error.";
7     }
8     else{
9         if($a->InsertarArchivo()){
10            echo "Todo correcto";
11        }
12    }
13 ?>
```

Figura 27: Ejemplo de uso de la clase Comprobar.

6.4 Ejemplos de codificación

Los lenguajes utilizados para el desarrollo del proyecto son JavaScript, PHP y MySQL. En esta sección se comentará el formato que tiene la totalidad de las páginas de la aplicación y a continuación se verán algunos ejemplos de codificación de los lenguajes antes mencionados en estas páginas.

En primer lugar en todas las páginas se tiene una llamada al archivo de seguridad, que comprueba que se está identificado, que la sesión no ha caducado y además inicializa la variable de sesión. A continuación se tiene una llamada al archivo de configuración, para rellenar los campos que sean necesarios, como podría ser la ayuda. Después se realiza una llamada a la base de datos para consultar las preferencias del usuario y realizar los cambios pertinentes

En la cabecera de la página además de las declaraciones habituales, también se tiene un apartado donde se definen las características de JQuery dentro de la página. En la Figura 28 se tiene un ejemplo que muestra como se inicializa el menú, el tooltip y las tablas utilizadas en la página. En la línea 3 se inicia el menú, en la 4 las tablas y en la 21 se inicializa la información que aparecerá en la página.

```
1 <script type="text/javascript" charset="utf-8">
2   $(document).ready(function() {
3     $("#accordion").accordion({ active:4, autoHeight:false });
4     oTable = $('#example').dataTable({
5       "bJQueryUI": true,
6       "sPaginationType": "two_button",
7       "bAutoWidth": false,
8       "bSortClasses": false,
9       "bSort": false,
10      "bStateSave": true,
11      "bFilter": false,
12      "oLanguage": {
13        "sLengthMenu": "Muestra _MENU_ entradas por página",
14        "sZeroRecords": "No se han encontrado registros",
15        "sInfo": "Mostrando de la _START_ hasta la _END_ de _TOTAL_ entradas",
16        "sInfoEmpty": "Mostrando 0 a la 0 de 0 entradas",
17        "sInfoFiltered": "(filtered from _MAX_ total records)"
18      },
19      "iDisplayLength": <? if($res->pagina != null){ echo $res->pagina; } else{ echo "10"; } ?>
20    });
21    $('#informacion span').tooltip({
22      extraClass: "tabla",
23      delay: 0,
24      showURL: false,
25      showBody: " - ",
26    });
27  });
28 </script>
```

Figura 28: Inicialización de JQuery.

El cuerpo de la página está dividido en cabecera, que es una variable que hay en el archivo de configuración, Menú (izquierda), que es un archivo en el cual se define el menú para toda la aplicación, Contenido (derecha), en esta parte es donde se mostrarán todos los formularios, las consultas, etc., y por último está el pie de página.

Para el envío de valores entre páginas a través de formularios se ha utilizado los métodos `$_POST` Y `$_GET`. También se ha utilizado la variable `$_FILES` para subir los archivos a la aplicación. Además se ha usado la variable `$_SESSION`, en lugar de cookies. La razón es que las sesiones se almacenan en el servidor hasta que el usuario cierra el navegador o se destruye la sesión, en vez de almacenarse en el ordenador del usuario, como es el caso de las cookies. Las sesiones se han utilizado para almacenar la información del usuario y para guardar las acciones realizadas por éste.

7. Pruebas

7.1 Pruebas realizadas por el alumno

Tras haber finalizado la codificación de la aplicación y los pequeños ensayos que se han ido haciendo mientras se desarrollaba ha llegado el momento de realizar un conjunto de pruebas más exhaustivo para comprobar el buen funcionamiento de todas las partes de la aplicación.

Para la realización de las pruebas se ha decidido utilizar solo un usuario identificado como administrador, ya que este puede realizar las mismas actividades que un usuario normal, además de las actividades administrativas.

7.1.1 Módulo de Proyectos y Módulo importar archivos

Así tras la identificación del administrador el primer paso es crear un nuevo proyecto. Para ello se accede a la zona correspondiente y se crea uno nuevo.

Ahora que ya está el proyecto creado se deben insertar los *tipos de datos* que formarán parte de la aplicación. Para ello se insertará un archivo con la información correspondiente desde la zona de importar y se le dará a subir archivo. En este apartado se seleccionará el archivo a subir.

Hay que tener en cuenta que si existen errores en el fichero, o simplemente, el dato ya existe en la base de datos, se mostrará una página indicándolo y se dará al usuario la opción de descargar un fichero con las filas que han producido el error.

En la Figura 29 se puede observar el error que aparece cuando se intenta subir un archivo de tipos de datos que ya existen en la base de datos. Se puede ver que se muestran las filas que contienen el error y el tipo de error que se ha cometido.

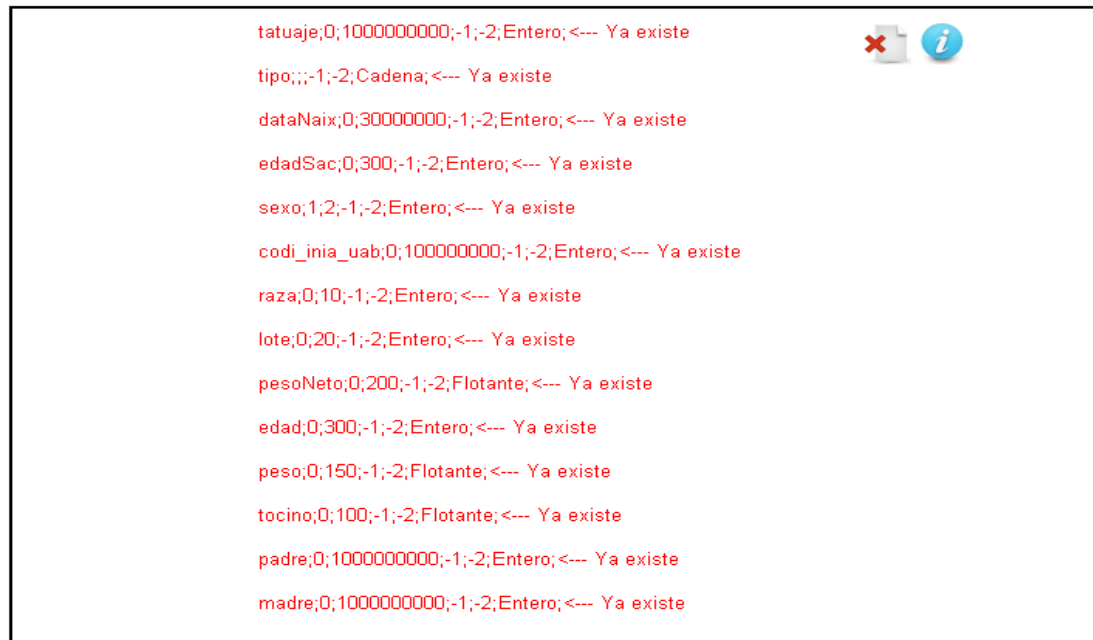


Figura 29: Página de error.

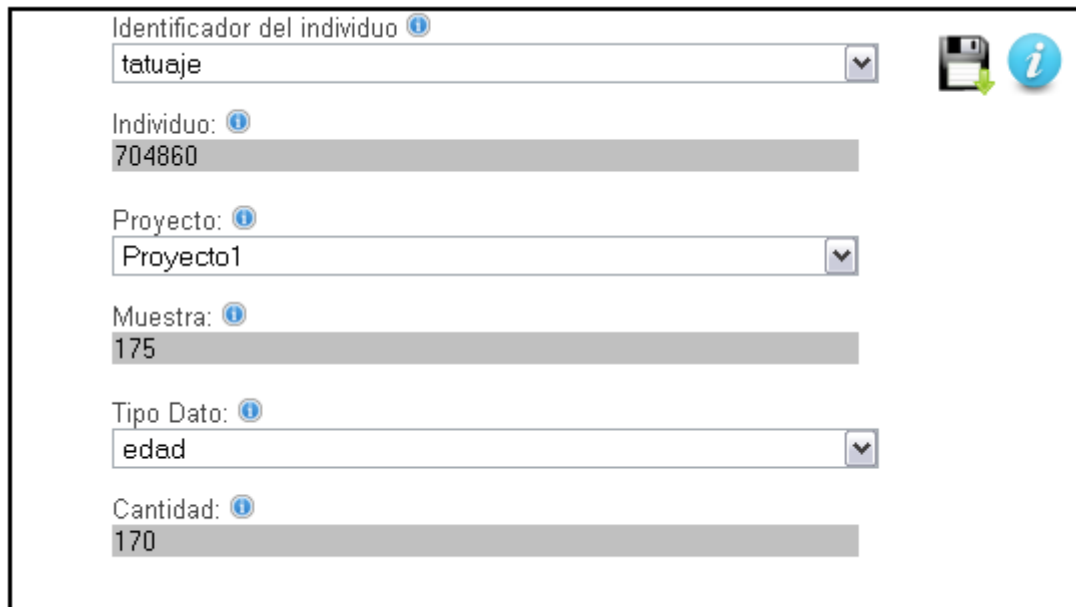
En este punto ya se ha comprobado que la creación de proyectos y la importación de archivos funcionan correctamente, así que a partir de este momento ya se pueden insertar los datos de los individuos. Esta información también se puede insertar desde ficheros pero como esta parte es similar a insertar los *tipos de datos* se procederá a insertar la información a mano para probar el resto de zonas de la aplicación.

7.1.2 Módulo insertar información

Para continuar se accede a la zona de *Gestión de datos* para insertar un nuevo individuo ya que si se intenta insertar un fenotipo o un genotipo sin que el individuo al que pertenecen exista en la base de datos se mostrará un mensaje de error indicando el problema que ha surgido.

Antes de insertar cualquier fenotipo se deben crear los lotes pertenecientes a un proyecto. Así cada fenotipo pertenecerá a un proyecto y a la vez a un lote dentro de ese proyecto.

Ahora que ya se tiene un lote y se ha insertado un individuo ya se puede añadir información sobre sus fenotipos y genotipos a la base de datos, para ello se debe acceder de nuevo a la zona de *Gestión de datos* y pulsar sobre la opción *Insertar Fenotipos* para incorporarlos a la base de datos de forma manual. El procedimiento a seguir es en primer lugar rellenar el formulario y a continuación pulsar el icono de Guardar Fenotipo. Aquí hay que tener en cuenta que si la cantidad introducida no está entre los límites asignados al tipo de dato se mostrará un mensaje de error.



Identificador del individuo ⓘ
tatuaje

Individuo: ⓘ
704860

Proyecto: ⓘ
Proyecto1

Muestra: ⓘ
175

Tipo Dato: ⓘ
edad

Cantidad: ⓘ
170

Figura 30: Insertar fenotipo.

Antes de insertar un genotipo se debe tener en cuenta que se ha de insertar un Marcador. Para ello en la zona de *Gestión de datos* se debe pulsar sobre *Insertar Marcador*, rellenar el formulario y guardar los cambios en la base de datos.

También en la zona de *Gestión de datos* se puede insertar un genotipo de forma manual. El procedimiento a seguir es el mismo que con el fenotipo, y se debe rellenar el formulario teniendo en cuenta que los *alelos* deben estar entre los límites asignados al tipo de dato.

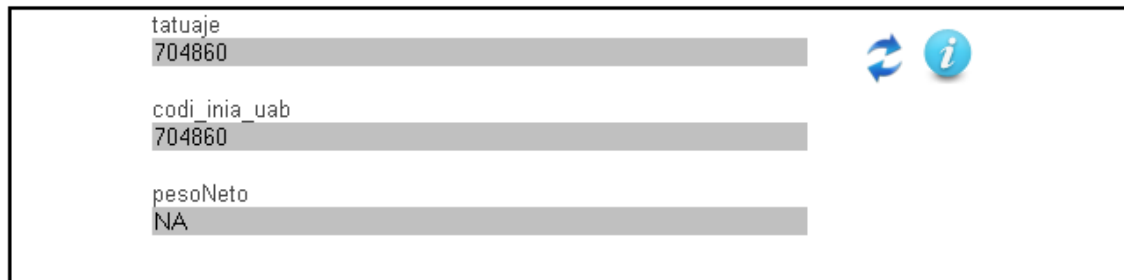
Hasta ahora se ha comprobado que los módulos de creación de proyectos, la importación de archivos y la inserción de datos a mano funciona correctamente así que se procederá a modificar los datos ya existentes.

7.1.3 Módulo modificar información

Asimismo en la zona de *Gestión de datos* se puede modificar o eliminar la información ya existente en la base de datos.

A continuación se modificará alguna de las características del individuo. Para realizar algún cambio en este apartado se tiene que acceder a la actividad de *Modificar Individuo*. Primero se tendrá que indicar el identificador del individuo y a continuación se tendrá que introducir en el apartado correspondiente el individuo que se desea modificar, después se deberá pulsar el icono de modificar individuo para que se muestre por pantalla las características asociadas a él que podrán ser cambiadas mediante el formulario. Por último para guardar los cambios se deberá pulsar el icono

Actualizar Individuo, a continuación se mostrará una pantalla con la información que se introducirá en la base de datos y se pedirá la confirmación para dichos cambios.





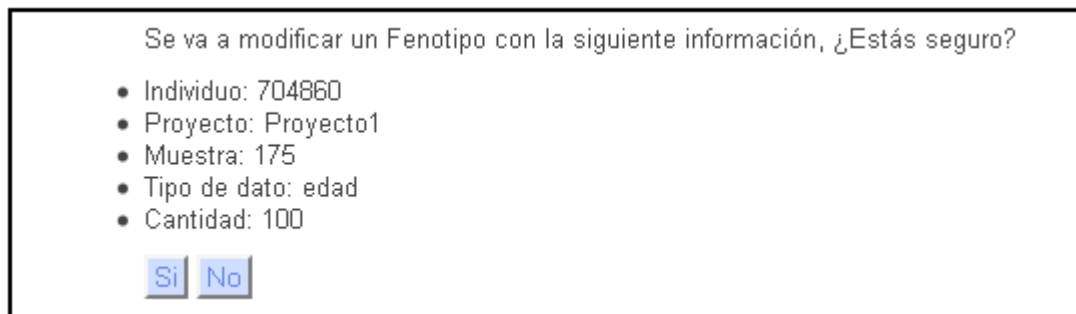
tatuaje	704860	 
codi_inia_uab	704860	
pesoNeto	NA	

Figura 31: Modificar individuo.

Para modificar los fenotipos de un individuo se tiene que acceder a la opción *Modificar Fenotipos* donde igual que en el caso anterior se tendrá que seleccionar primero el identificador individuo y a continuación escribir el individuo al que pertenece el fenotipo a modificar, el siguiente paso será seleccionar el proyecto y escribir el lote al que pertenece, por último se deberá seleccionar el tipo de dato y escribir la cantidad. Para conservar los cambios realizados se deberá pulsar el icono *Actualizar Fenotipos* y confirmarlos en la siguiente pantalla.



Se va a modificar un Fenotipo con la siguiente información, ¿Estás seguro?

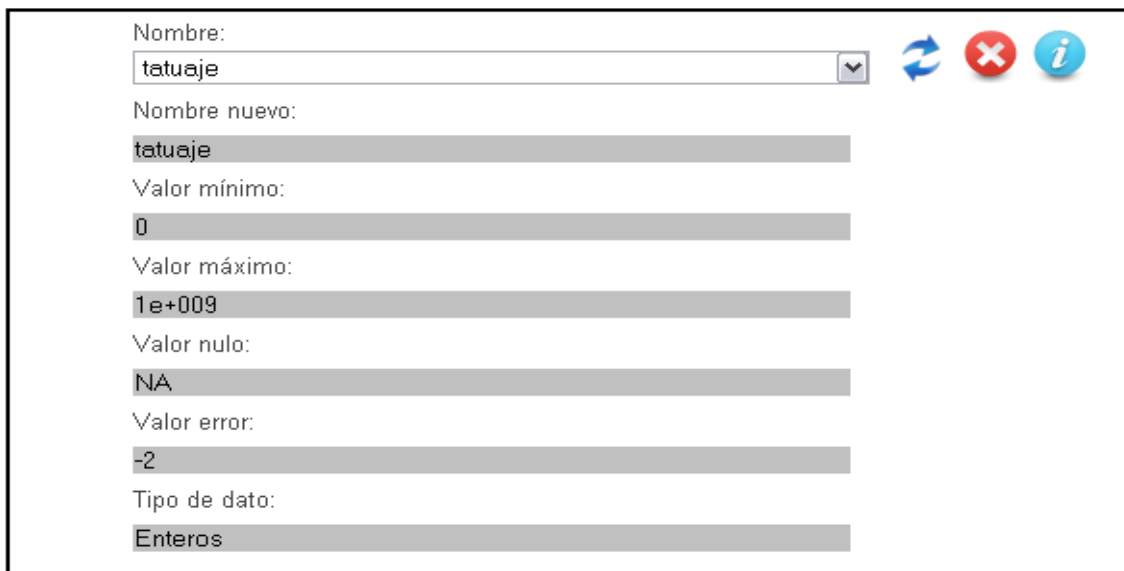
- Individuo: 704860
- Proyecto: Proyecto1
- Muestra: 175
- Tipo de dato: edad
- Cantidad: 100

Figura 32: Confirmación de los cambios.

En el caso de los genotipos el procedimiento es similar a los anteriores, en este caso se deberá acceder a la sección de *Modificar Genotipos* y rellenar el formulario con los nuevos datos que se desean introducir. Una vez modificados los campos del formulario para no perderlos se deberá pulsar el icono *Actualizar Genotipos* y confirmar los cambios en la siguiente pantalla.

Si se desea modificar un marcador ya existente en la base de datos se tendrá que acceder a la opción *Modificar Marcador* y rellenar el formulario que se muestra por pantalla, seleccionando en primer lugar el marcador que queremos actualizar. Por último para conservar los nuevos valores introducidos se deberá pulsar el icono *Actualizar Marcador* y en la siguiente pantalla que aparece confirmar esos cambios.

Igualmente se podrán cambiar los tipos de datos pulsando sobre *Modificar Tipo de Datos* y rellenando el formulario correspondiente.



The screenshot shows a web form for modifying data types. It contains the following fields and values:

- Nombre: tatuaje
- Nombre nuevo: tatuaje
- Valor mínimo: 0
- Valor máximo: 1e+009
- Valor nulo: NA
- Valor error: -2
- Tipo de dato: Enteros

At the top right of the form, there are three icons: a blue refresh icon, a red 'X' icon, and a blue information icon.

Figura 33: Modificar tipo de datos.

Hasta este punto se ha mostrado como se puede insertar y modificar información de la base de datos y se ha comprobado su buen funcionamiento. A continuación se mostrará la forma de consultar dichos datos, para ello se debe acceder a la zona de *Consultas* que contiene cinco opciones que se mostrarán en las siguientes páginas.

7.1.4 Módulo consultas y Módulo de operaciones con datos

En primer lugar se mostrará la opción *Consultar Tipo de Datos*, aquí se mostrará en forma de tabla los tipos de datos que se han insertado en la aplicación y se dará la opción al usuario de descargar la información seleccionada en un fichero con formato CSV con solo pulsar sobre el icono correspondiente.

Tipos de datos que existen en la aplicación

Muestra entradas por página

	id	nombre	Mínimo	Máximo	Nulo	Error	Tipo
<input type="checkbox"/>	1	tatuaje	0	1e+09	-1	-2	Enteros
<input type="checkbox"/>	2	tipo			-1	-2	Cadenas
<input type="checkbox"/>	3	dataNaix	0	3e+07	-1	-2	Enteros
<input type="checkbox"/>	4	edadSac	0	300	-1	-2	Enteros
<input type="checkbox"/>	5	sexo	1	2	-1	-2	Enteros
<input type="checkbox"/>	6	codi_inia_uab	0	1e+08	-1	-2	Enteros
<input type="checkbox"/>	7	raza	0	10	-1	-2	Enteros
<input type="checkbox"/>	8	lote	0	20	-1	-2	Enteros
<input type="checkbox"/>	9	pesoNeto	0	200	-1	-2	Flotantes
<input type="checkbox"/>	10	edad	0	300	-1	-2	Enteros
<input type="checkbox"/>	11	peso	0	150	-1	-2	Flotantes
<input type="checkbox"/>	12	padre	0	1e+09	-1	-2	Enteros
<input type="checkbox"/>	13	madre	0	1e+09	-1	-2	Enteros
<input type="checkbox"/>	14	tocino	0	100	-1	-2	Flotantes
<input type="checkbox"/>	15	microsatelites			NA	NB	Cadenas

Mostrando de la 1 hasta la 15 de 15 entradas

[Marcar toda la consulta](#) [Marcar toda la página](#)

Figura 34: Consulta de tipo de datos.

Las tres siguientes opciones, *Consultar Fenotipos*, *Consultar Genotipos* y *Consultar Individuo*, siguen el siguiente esquema:

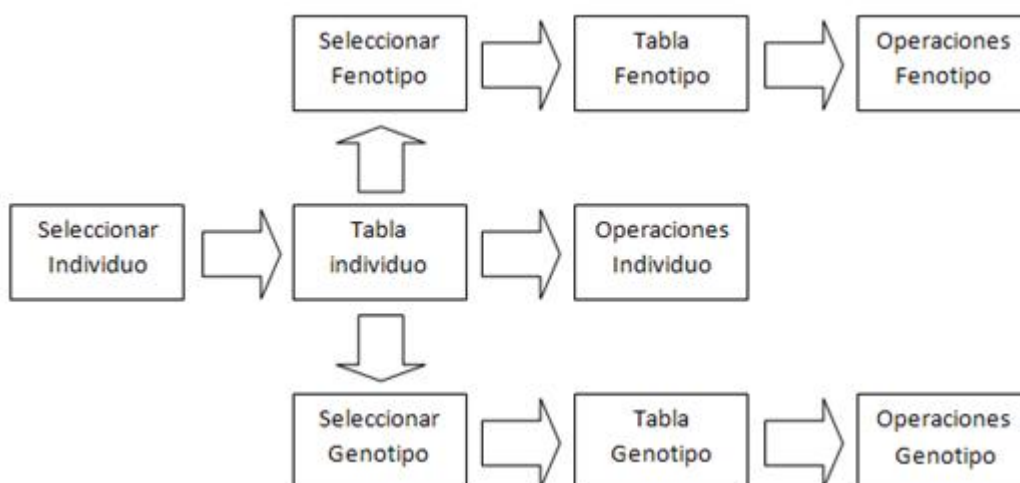


Figura 35: Esquema de las consultas.

Como se puede observar se puede seleccionar los fenotipos y genotipos de todos los individuos o únicamente los de los individuos que previamente se han escogido.

A continuación se mostrará la opción *Consultar Fenotipos*. En primer lugar se deberá seleccionar el experimento, los lotes y los rasgos que se mostrarán por pantalla dentro de una tabla para su mejor visualización.

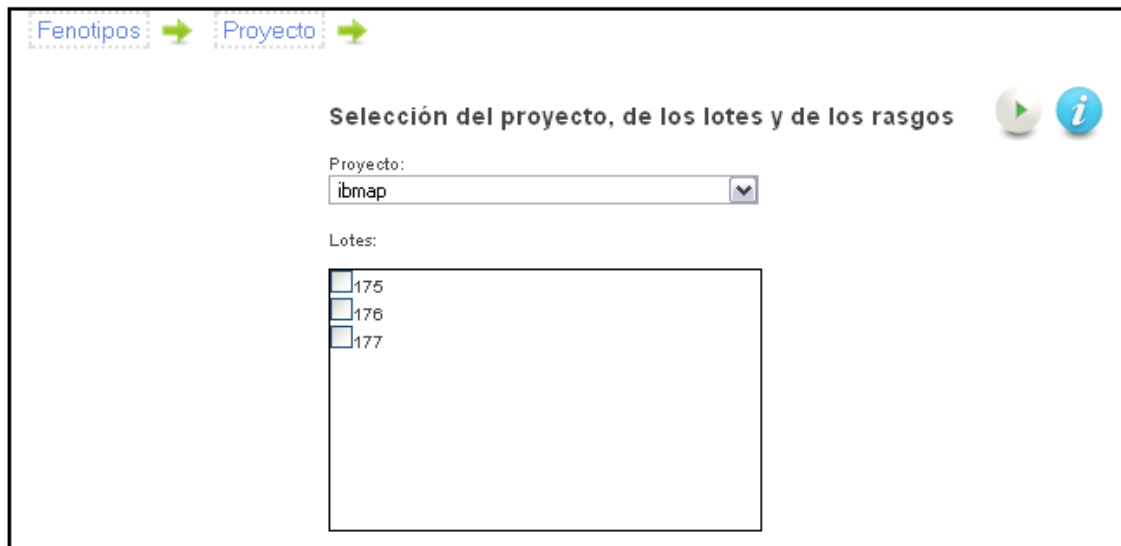


Figura 36: Selección de los datos.

En la pantalla que se muestra al realizar la selección de la información a ver el usuario podrá seleccionar las columnas y las filas en las cuales desea realizar alguna operación con R. En caso de no marcar ninguna columna simplemente se descargarán los datos seleccionados.

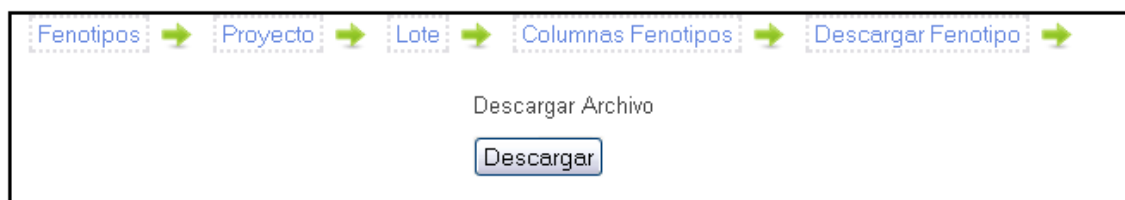


Figura 37: Descargar fenotipos.

En el caso de haber seleccionado alguna columna se accederá a la página de operaciones donde se mostrarán los datos consultados y un resumen con las operaciones realizadas. Por último, desde esta misma página se podrá descargar ambas opciones.

Fenotipos → Proyecto → Lote → Columnas Fenotipos → Descargar Fenotipo → Operaciones Fenotipos →

Resumen/Estadística

Muestra 100 entradas por página

Individuo	Lote	edad	peso	tocino
704852	175	172	113	16.5
704852	176	184	99.5	16.5

Mostrando de la 1 hasta la 2 de 2 entradas

Muestra 100 entradas por página

operacion(columna)	Resultado
max(edad):	172
hist(peso):	imagen

Mostrando de la 1 hasta la 2 de 2 entradas

Figura 38: Resumen fenotipo.

En la siguiente imagen se muestra el camino seguido para la consulta de fenotipos.

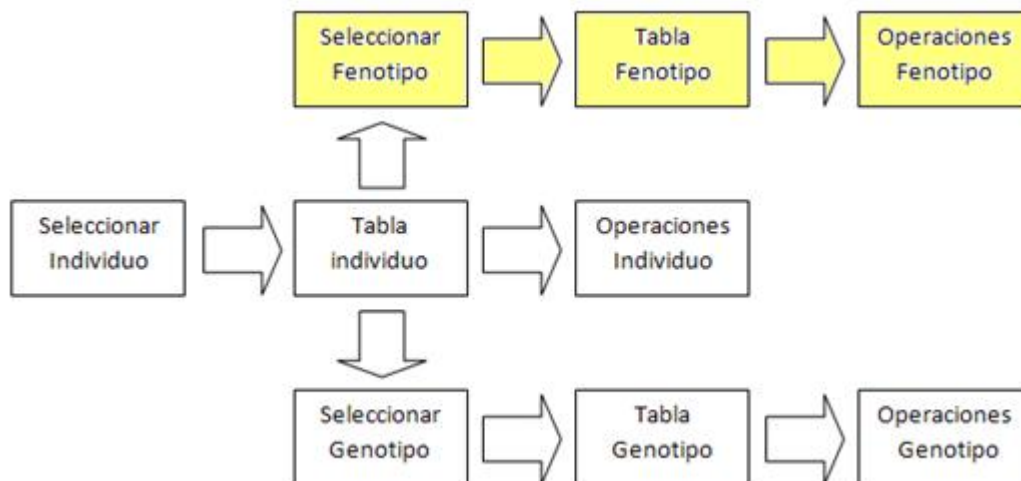


Figura 39: Camino seguido en la consulta del fenotipo.

En la opción de *Consultar Genotipos* se deberá seleccionar el tipo de dato que se mostrará por pantalla. El resto de funcionalidades de esta opción tienen el mismo mecanismo que en la opción de *Consultar Fenotipos*, para utilizar R se deberá seleccionar tanto las filas como las columnas involucradas en la operación que se desea realizar y en el caso de no seleccionar ninguna columna se descargarán los datos seleccionados de la tabla.

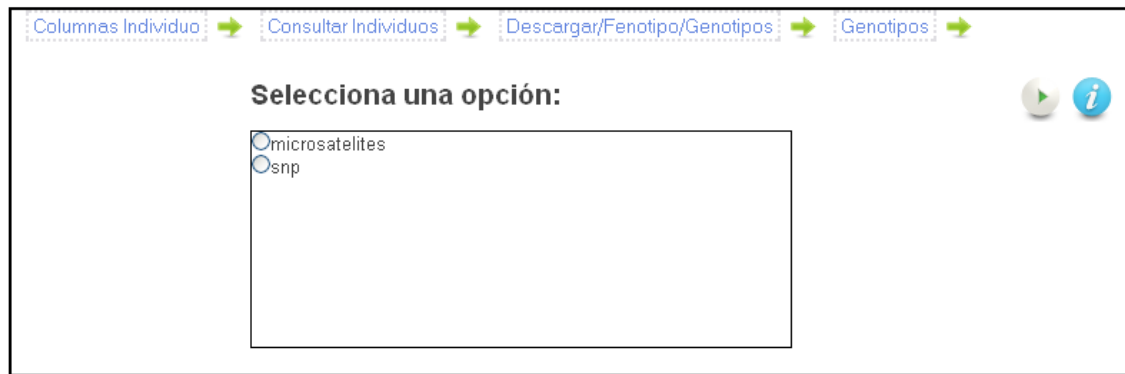


Figura 40: Selección de genotipos.

El camino seguido en esta zona ha sido el que se muestra en la siguiente imagen.

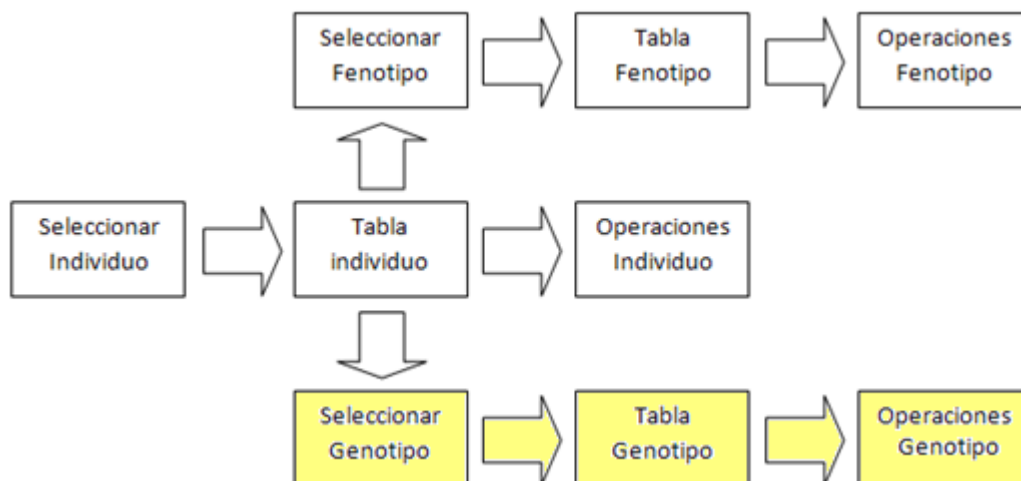


Figura 41: Camino seguido en los Genotipos.

En *Consultar Individuos* se deberá seleccionar los rasgos que se mostrarán en la consulta a realizar, a continuación se mostrará una tabla similar a la de las consultas anteriores con una metodología de trabajo idéntica a las opciones anteriormente comentadas.

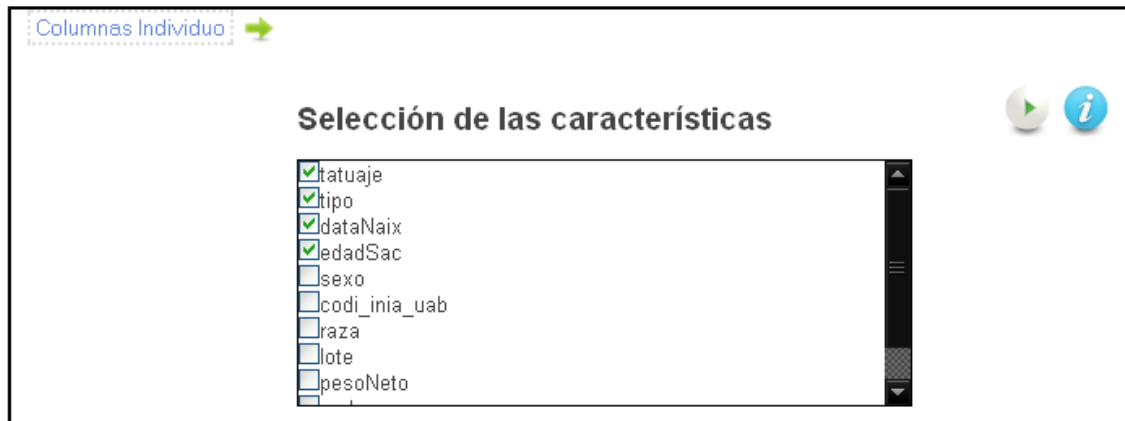


Figura 42: Selección de rasgos del individuo.

El camino seguido en esta ocasión ha sido el que se muestra en la imagen.

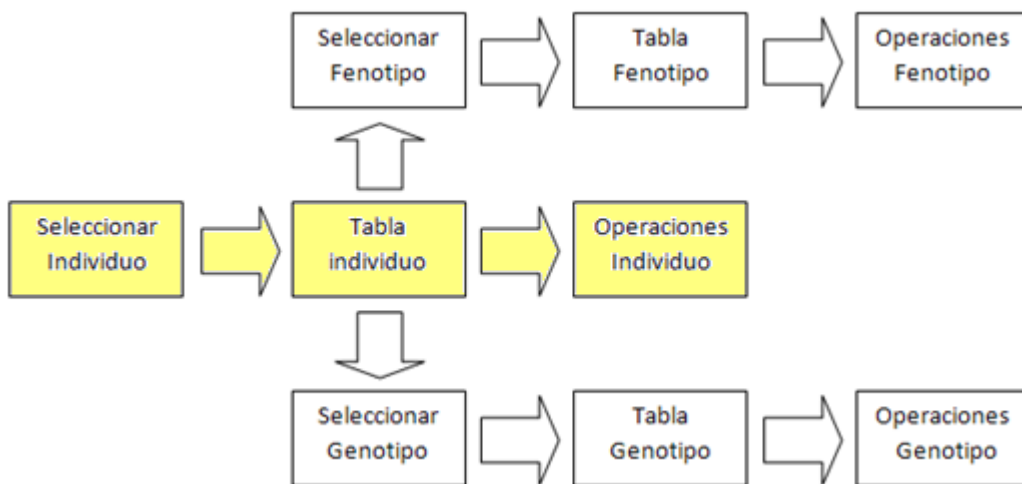


Figura 43: Camino seguido en la consulta de individuos.

La diferencia de esta consulta respecto a las anteriores, es que podemos acceder a los Fenotipos o Genotipos de los individuos seleccionados, para ello se mostrará una página intermedia donde se podrá seleccionar la acción a realizar.

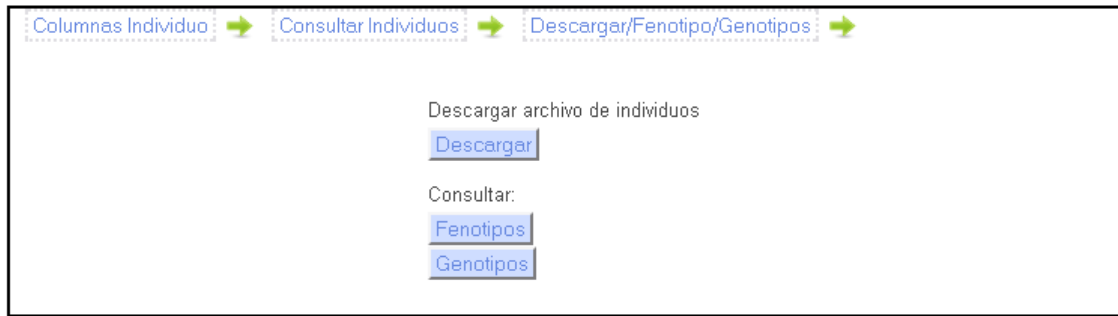


Figura 44: Selección de la consulta.

Si se accede a alguna de esas dos opciones la forma de proceder será como se ha explicado en los respectivos apartados anteriores.

Las tres últimas consultas comentadas se guardarán en la variable de sesión, y para retroceder se deberá seleccionar el paso anterior en el 'breadcrumb' o utilizar la quinta opción de la zona de consultas, que es *Eliminar Consulta*.

Una vez terminada la parte de *Consultas* del menú, y comprobado su buen funcionamiento, se procederá a comentar la zona de *Sesiones* donde se podrá guardar la consulta realizada con el fin de poder continuar trabajando en jornadas posteriores desde el último punto guardado. También se deberá tener en cuenta que solo se podrá guardar una sesión por usuario.

7.1.5 Módulo sesiones

Para recuperar una sesión guardada previamente se deberá acceder a la actividad correspondiente y recuperarla según corresponda. Una vez se ha contestado afirmativamente a la pregunta que se muestra se puede acceder a la zona de consultas y comprobar que efectivamente se ha recuperado la sesión.

Desde la zona de *Sesiones* también se podrá enviar una consulta realizada a otro usuario de la aplicación, para ello se deberá seleccionar el investigador que recibirá la sesión con la que se está trabajando.

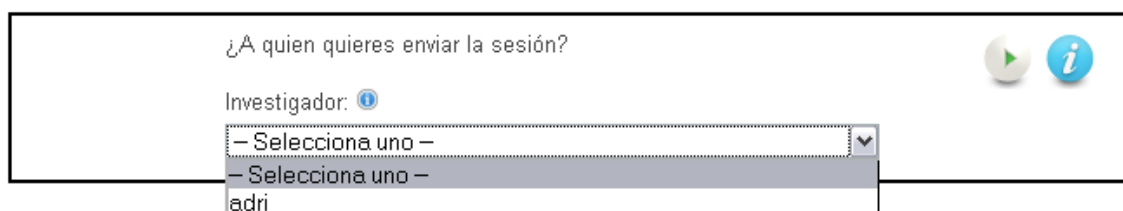


Figura 45: Envío de consulta.

De igual modo que un usuario podrá enviar consultas a otro, también las podrá recibir, y por ello habrá que tener en cuenta que como cada usuario solo podrá tener una sesión activa, en el momento de recuperar una consulta enviada por otro usuario se borrará la sesión actual que se esté utilizando.

7.1.6 Módulo administrador

Por último, cuando un usuario esta registrado como administrador tendrá una zona más que el resto de usuarios no administradores.

En esta sección se podrá descargar una copia de seguridad de la base de datos, para ello se deberá seleccionar el enlace *Backup base de datos* y se descargará el fichero con la información contenida en la ella en formato SQL.

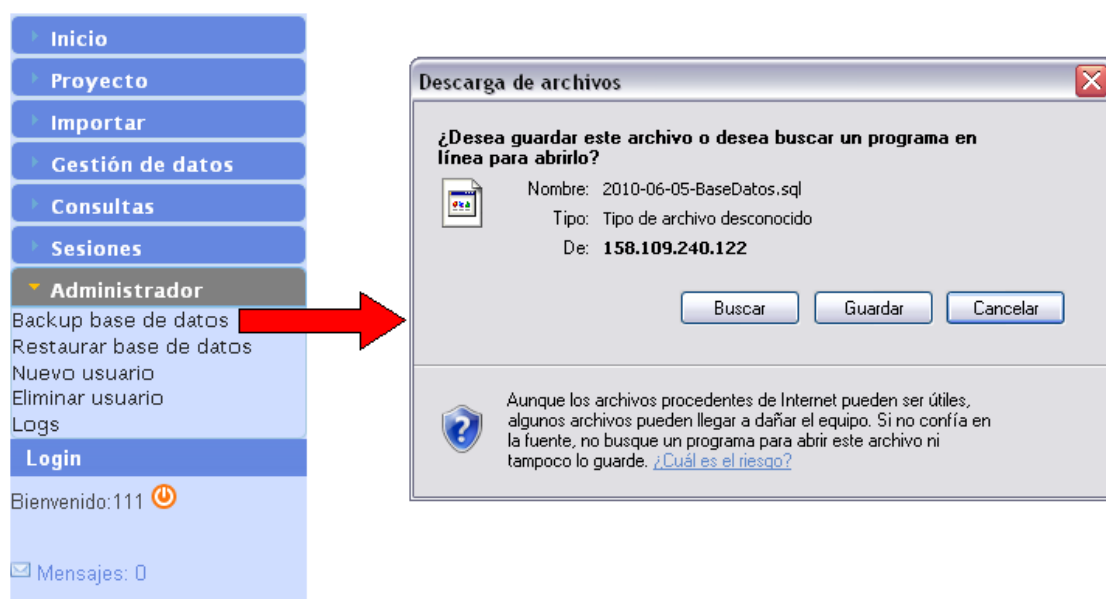


Figura 46: Backup de la base de datos.

Por otra parte se podrá restaurar la versión de la base de datos anterior, pero para que esta opción funcione se deberá haber insertado previamente información en la base de datos.

Otra de las opciones que podrá realizar el administrador es la de dar de alta a nuevos usuarios, para ello deberá rellenar el formulario pertinente. En caso de que el usuario ya existiera se mostraría un mensaje de error. En esta zona hay que prestar atención al campo de email ya que al nuevo usuario se le informará de que esta registrado enviándole uno.

Por otra parte el administrador también podrá dar de baja a un usuario, solamente deberá seleccionar un investigador y eliminarlo de la base de datos.

Otra de las opciones que tendrá el administrador de la aplicación es la de consultar los logs que se hayan producido durante el uso de la aplicación por el resto de usuarios. En esta opción también tiene la posibilidad de poder descargar un archivo con toda esa información.

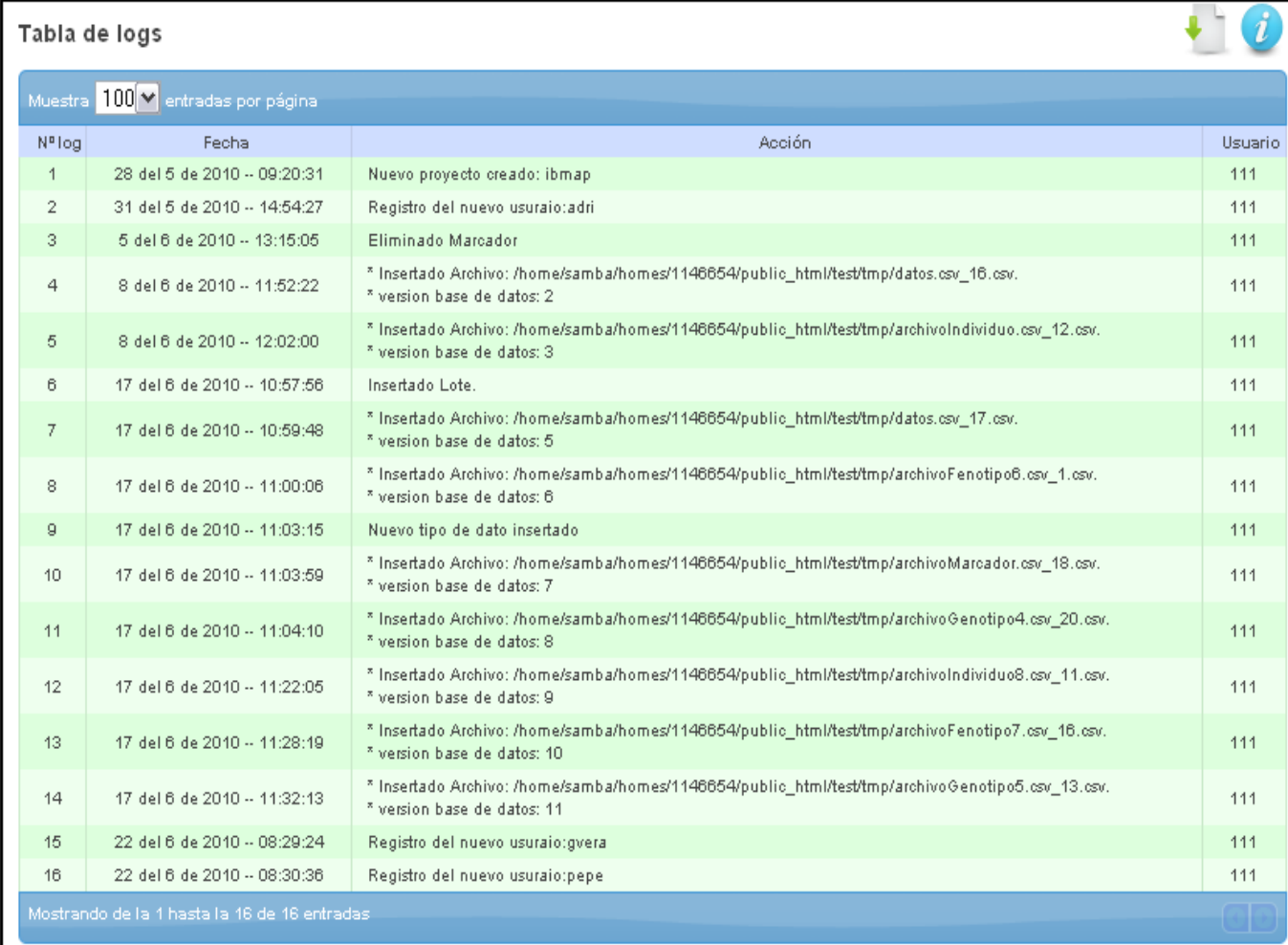


Tabla de logs

Muestra 100 entradas por página

Nº log	Fecha	Acción	Usuario
1	28 del 5 de 2010 -- 09:20:31	Nuevo proyecto creado: ibmap	111
2	31 del 5 de 2010 -- 14:54:27	Registro del nuevo usuraio:adri	111
3	5 del 6 de 2010 -- 13:15:05	Eliminado Marcador	111
4	8 del 6 de 2010 -- 11:52:22	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/datos.csv_16.csv. * version base de datos: 2	111
5	8 del 6 de 2010 -- 12:02:00	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/archivoIndividuo.csv_12.csv. * version base de datos: 3	111
6	17 del 6 de 2010 -- 10:57:56	Insertado Lote.	111
7	17 del 6 de 2010 -- 10:59:48	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/datos.csv_17.csv. * version base de datos: 5	111
8	17 del 6 de 2010 -- 11:00:06	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/archivoFenotipo6.csv_1.csv. * version base de datos: 6	111
9	17 del 6 de 2010 -- 11:03:15	Nuevo tipo de dato insertado	111
10	17 del 6 de 2010 -- 11:03:59	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/archivoMarcador.csv_18.csv. * version base de datos: 7	111
11	17 del 6 de 2010 -- 11:04:10	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/archivoGenotipo4.csv_20.csv. * version base de datos: 8	111
12	17 del 6 de 2010 -- 11:22:05	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/archivoIndividuo8.csv_11.csv. * version base de datos: 9	111
13	17 del 6 de 2010 -- 11:28:19	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/archivoFenotipo7.csv_16.csv. * version base de datos: 10	111
14	17 del 6 de 2010 -- 11:32:13	* Insertado Archivo: /home/samba/homes/1146654/public_html/test/tmp/archivoGenotipo5.csv_13.csv. * version base de datos: 11	111
15	22 del 6 de 2010 -- 08:29:24	Registro del nuevo usuraio:gvera	111
16	22 del 6 de 2010 -- 08:30:36	Registro del nuevo usuraio:pepe	111

Mostrando de la 1 hasta la 16 de 16 entradas

Figura 47: Tabla de log.

7.2 Validación del usuario final

Una vez se implementado un primer prototipo se le pasó al usuario final un cuestionario para validar el buen funcionamiento de la aplicación. Este cuestionario se presenta a continuación:

1.- Fecha de la realización de la prueba: 8 junio 2010.

2.- ¿Qué aspectos de la aplicación se consideran más valiosos?

El hecho de poder guardar los datos con seguridad y uniformidad, y que todos los investigadores implicados puedan consultarlos.

3.- La opciones van de completamente de acuerdo (5) a discrepo completamente (0).

3.1.- La aplicación permite encontrar la información de forma sencilla:

5..... **4...X** 3..... 2..... 1..... 0.....

3.2. -La aplicación cubre las necesidades de información que tengo:

5..... 4..... **3...X** 2..... 1..... 0.....

3.3.- Los instrumentos para la recuperación de información (formularios, instrucciones, etc.) son fáciles de utilizar y me permiten encontrar lo que busco:

5..... **4...X** 3..... 2..... 1..... 0.....

3.4.- El modo en el que se presenta la información de la aplicación es clara y comprensible:

5..... **4...X** 3..... 2..... 1..... 0.....

3.5.- El acceso a los datos almacenados es fácil e intuitivo:

5..... **4...X** 3..... 2..... 1..... 0.....

3.6.- Pensé que había demasiada inconsistencia en la aplicación:

5..... 4..... 3..... 2..... **1...X** 0.....

3.7.- Imagino que la mayoría de las personas aprenderían rápidamente a utilizar la aplicación:

5..... 4..... **3...X** 2..... 1..... 0.....

3.8.- Necesito aprender muchas cosas antes de manejar esta aplicación:

5..... 4..... 3..... 2..... 1..... **0...X**

3.9.- La aplicación me parece demasiado lenta:

5..... 4..... 3..... 2..... 1..... **0...X**

4.- En comparación con otras alternativas, la aplicación le parece:

Mucho mejor..... Mejor..... Igual..... Peor..... Mucho peor.....

5.-Cuál es su grado de satisfacción general con la aplicación:

Muy satisfecho..... **Satisfecho...X** Insatisfecho..... Muy insatisfecho.....

6.- ¿Qué se podría hacer para mejorar la aplicación?

Algunos aspectos, como la subida de tipos y de genotipos, no son del todo intuitivos.

7.- ¿Cree que la aplicación a cumplido con las expectativas? ¿Por qué?

Yo diría que en un 80%. Aunque el estudiante ha hecho un trabajo muy meritorio y estaba siempre dispuesto a escuchar y modificar lo que se comentaba, creo que es difícil entender la complejidad de los datos de genética en tan poco tiempo, y viceversa, es difícil para el biólogo entender los requisitos y peculiaridades de las bases de datos. No siempre es fácil prever el tipo de dato que se va a generar en los próximos experimentos.

7.3 Valoración de los resultados de validación del usuario final

Viendo el grado de satisfacción del usuario final y estudiando sus respuestas, se puede considerar que la aplicación tiene un buen nivel de cumplimiento de las expectativas, a pesar de que se tiene la certeza de que la aplicación tiene ciertas limitaciones ocasionadas por el hecho de que el proyecto se ha de finalizar en un año académico.

8. Conclusiones

Llegado a este punto de la memoria, se valorará si los objetivos descritos al principio se han cumplido con la finalización del proyecto. Este último capítulo es una valoración sobre el trabajo realizado, así como la declaración de posibles ampliaciones.

8.1 Consecución de objetivos

A lo largo del desarrollo del proyecto se han ido alcanzando los objetivos explicados en el capítulo 1 de esta memoria. Se ha obtenido una aplicación multiusuario online que facilita el intercambio de información entre los usuarios.

Por otra parte se ha conseguido crear una base de datos que permite trabajar con distintos tipos de individuos, como podrían ser animales o vegetales, gracias a que los tipos de datos son guardados en la aplicación y son definidos por el usuario.

Otro de los objetivos que se buscaba con este proyecto era la de realizar análisis estadísticos sencillos con la información contenida en la base de datos. Esta parte se ha logrado hacer gracias a la integración que se ha podido realizar entre PHP y R.

En definitiva, se han logrado en mayor o menor medida todos los objetivos planteados al inicio del proyecto.

8.2 Planificación real

A continuación se describe la planificación real del proyecto una vez se ha finalizado. Como podemos ver han surgido nuevas tareas y la duración de otras a cambiado.

Fases del proyecto

Descripción de las tareas	Duración
Estudio previo del proyecto	6 h
Entrevistas con el tutor	1 h
Estudio previo del proyecto	5 h
Estudio de viabilidad	40 h
Entrevistas con el tutor	5 h
Estudio de viabilidad	35 h
Diseño de la base de datos	40 h
Estudio de la información proporcionada	10 h
Aportaciones del profesor de la unidad genética	10 h
Diseño de la base de datos	20 h
Diseño de la interfaz de usuario	33 h
Entrevistas con tutor y con profesor de unidad de genética	3 h
Diseño de la interfaz de usuario	30 h



Programación de la aplicación	145 h
Entrevistas con tutor y con profesor de unidad de genética	5 h
Programación de la base de datos	2 h
Programación de la interfaz de usuario	70 h
Programación de parser para la subida de archivos Excel	30 h
Programación de un script para la comunicación entre la aplicación y R	15 h
Programación de hojas de estilo	13 h
Comprobación de cumplimiento de estándares	10 h
Pruebas	40 h
Creación de instalador	5 h
Elaboración de la documentación	61 h
Entrevistas con el tutor	6 h
Memoria del proyecto	50 h
Manual de creación de archivos	2 h
Manual de usuario	2 h
Manual de instalación	1 h
Total	370 h

La duración final ha sido de 370 horas con lo que ha habido una desviación de 40 horas con la planificación que se hizo en el estudio de viabilidad. Esta desviación ha surgido por las dificultades que presenta la comunicación con usuarios finales y el no tener claro desde el principio las necesidades reales que se querían cumplir, pero a pesar de todos estos obstáculos el proyecto ha podido ser finalizado en el plazo previsto y la viabilidad no se ha visto alterada.

Alguna de las variaciones que se han producido ha sido la creación de manuales de usuario, de instalación y de creación de archivos. También han aumentado las horas dedicadas a algunas tareas como pueden ser la programación de la interfaz de usuario o la creación de un parser para la subida de archivos.

En la Figura 48 se puede observar el diagrama de Gantt del proyecto, con las nuevas estimaciones de la duración de cada una de las tareas una vez asignados los recursos. Se puede observar que el proyecto se ha retrasado hasta el 25 de junio, un retraso de más de un mes respecto a la planificación real.

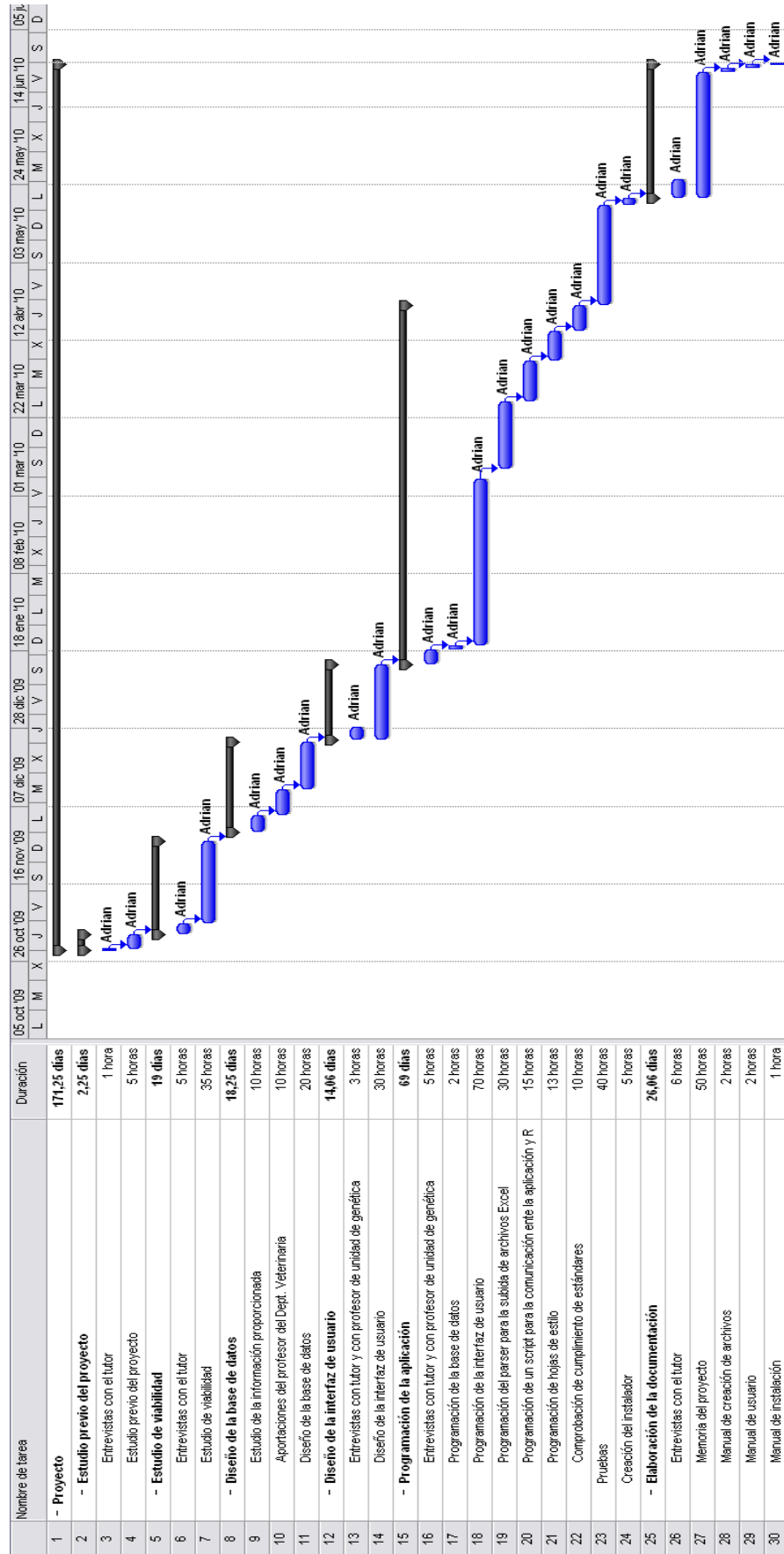


Figura 48: Diagrama de Gantt

8.3 Líneas de ampliación

Como en todo proyecto existen posibles ampliaciones que pueden mejorar o añadir funcionalidades o calidad a la aplicación. Así algunas de las posibles mejoras que se podrían realizar son:

- **Internacionalización:** Con esta ampliación se permitiría ver la web en varios idiomas y así adaptarse a las preferencias del usuario que está trabajando con la aplicación.
- **Consultas más completas:** Permitiría la unión de varias consultas realizadas por el usuario, y por consiguiente no tener que realizarlas por separado.
- **Análisis estadísticos definidos por el usuario:** En vez de definir el programador los análisis que se pueden realizar en la aplicación, se le daría al usuario la opción de añadir sus propios análisis.
- **Crear permisos para las partes del menú:** El administrador tendría la opción de decidir qué partes del menú podría ver cada usuario.

8.4 Valoración personal

Este proyecto me ha servido para poder profundizar en los conocimientos que ya tenía, pero que nunca antes había utilizado para la realización de un proyecto de estas magnitudes.

Gracias a las equivocaciones que ha habido y a causa de las cuales se ha tenido que insistir en la búsqueda de posibles soluciones y caminos por donde seguir he adquirido experiencia a la hora de programar y de buscar soluciones diversas.

También he aprendido que un proyecto informático no se basa solo en la programación, sino que hay otros muchos aspectos que hay que tener en cuenta, como podría ser el diseño, que es una parte importante de todo proyecto.

También ha surgido la oportunidad de trabajar con un usuario final con lo que he aprendido las dificultades que se presentan en este tipo de proyectos. Una de estas dificultades ha sido que el usuario hablaba un lenguaje distinto al del desarrollador de la aplicación con lo que esto conllevó a que se tardara un poco más en definir claramente los objetivos que se perseguían en el proyecto. Con el paso del tiempo esta distancia en el lenguaje se fue acortando hasta un punto en el que se aproximaron bastante y la comunicación entonces fue más fluida.



9. Bibliografía

9.1 Fuentes bibliográficas

- [1] Andy Harris, *Manual imprescindible de programación con PHP 6 y MySQL*, 2009 Anaya Multimedia.
- [2] Lee Babin, *Introducción a Ajax con PHP*, 2007 Anaya Multimedia.
- [3] Rasmus Lerdorf, Kevin Tatroe, Peter MacIntyre, *Programming PHP*, 2006, O'Reilly
- [4] Francisco Charte Ojeda. *La biblia de HTML*, 2005 Ed. Anaya Multimedia.

9.2 Fuentes electrónicas

9.2.1 Fuentes para el desarrollo de la aplicación

- [5] PHP [Online]. Disponible en: <http://php.net/>
- [6] Desarrollo web [Online]. Disponible en: <http://www.desarrolloweb.com/>
- [7] Foros del web [Online]. Disponible en: <http://www.forosdelweb.com/>
- [8] JQuery [Online]. Disponible en: <http://jquery.com/>
- [9] DataTables [Online]. Disponible en: <http://www.datatables.net/>
- [10] MySQL [Online]. Disponible en: <http://www.mysql.com/>
- [11] Wikipedia [Online]. Disponible en: <http://es.wikipedia.org/>
- [12] mkInstaller [Online]. Disponible en: <http://mkinstaller.nisu.org/index.php.es/>
- [13] R [Online]. Disponible en: <http://www.r-project.org/>

9.2.2 Fuentes relacionadas con la temática de la aplicación

- [14] Scientific Electronic Library Online [Online]. Disponible en: <http://www.scielo.org.pe/scielo.php>
- [15] Fermentek: biotechnology [Online]. Disponible en: <http://www.fermentek.co.il/>
- [16] Degesa JSR [Online]. Disponible en: <http://www.degesa.com/calidad.htm>
- [17] Miguel Calvo. Milk Science [Online]. Disponible en: <http://milksci.unizar.es/>
- [18] División celular: Meiosis y reproducción sexual [Online]. Disponible en: <http://mail.efn.uncor.edu/dep/biologia/intrbiol/meiosis.htm>
- [19] Genotipo y Fenotipo [Online]. Disponible en: <http://www.educaplus.org/play-48-Genotipo-y-fenotipo.html>
- [20] Morris A. Swertz. Molgenis. Disponible en: <http://www.molgenis.org/>
- [21] Jari Häkkinen, Nicklas Nordborg, Martin Svensson, Johan Vallon-Christersson. BASE. Disponible en: <http://base.thep.lu.se/>
- [22] BioMart. Disponible en: <http://www.biomart.org/>

Anexo I: Creación de ficheros

Creación de los archivos

En este anexo se describe a los usuarios el formato que deben tener los archivos, que se podrán subir a la aplicación. Todo archivo que no se ajuste a estos formatos producirá un error.

Creación del archivo de Tipo de datos

En esta sección se describirá el archivo de tipo de datos. Éste contendrá los tipos de datos que serán utilizados en la aplicación. Así tenemos que los datos deben estar descritos de la siguiente forma:

1. Nombre del tipo de dato.
2. Valor mínimo que puede tomar el dato (Si el tipo de dato va a ser una cadena este dato se quedara vacío).
3. Valor máximo que puede tomar el dato (Si el tipo de dato va a ser una cadena este dato se quedara vacío).
4. Este número será el valor que tomará el dato si no se tiene información de él.
5. Aquí se pondrá el valor que tendrá la información si se ha producido un error.
6. En esta sección se pondrá la clase a la que pertenece el Tipo de dato, que puede ser *Entero*, *Flotante* o *Cadena*. Esta información deberá estar escrita tal cual porque sino la aplicación mostrará un mensaje de error.

Una vez se tiene el archivo creado habrá que tener en cuenta, que si se produce un error, no se insertará ninguna fila ya que se comprobará la integridad de los datos antes de subirlos. También se tendrá que tener en cuenta que la información deberá estar separada por punto y coma (Archivos CSV).

Además se tendrá que tener presente a la hora de crear el archivo que existen ciertos tipos de datos que estarán asignados previamente a las palabras siguientes:

- *padre y madre*: Estos datos se utilizarán para identificar a los progenitores de los especímenes en el archivo de Características del Individuo. Estas palabras deberán ser del tipo *Enteros* y con el mismo rango que el identificador del sujeto.

- *experimento y tiempo*: Estos datos se manejarán para identificar el experimento al que pertenecen los fenotipos y el lote en el cual ha sido tomada la muestra. Las cifras que identifican al lote deben ser del tipo *Enteros*. Además no hace falta que estén definidos como tipo de datos ya que existen campos en la base de datos que los identifican.
- *posicionFisicaRelativa, PosicionGenetica, nombreMarcador, cromosoma, clase, nivel, ploidia y comentario*: Estos datos son utilizados en el archivo de Marcadores e indican las características que tendrá el marcador. Al igual que el experimento y el tiempo estas palabras no hace falta definirlos en tipo de datos ya que existe un campo en la base de datos asignados para ellos.

En la Figura 1 se puede observar un ejemplo de archivo de Tipo de datos. Como se puede comprobar en la línea 1 y 9 hay dos datos numéricos (Enteros y Flotantes) que tienen todos los datos anteriormente descritos. En cambio en la línea 2 se puede observar que el dato es una Cadena y se puede ver que la cantidad de mínimo y máximo no están definidos, pero sí que se preserva un hueco para cada uno de ellos.

También se tiene que en las líneas 13 y 15 están definidos los parámetros del padre y la madre, que tienen las mismas propiedades que el identificador del individuo que en este caso es el tatuaje.

```
1  datos
2  tatuaje;0;1000000000;-1;-2;Entero
3  tipo;;;-1;-2;Cadena
4  dataNaix;0;30000000;-1;-2;Entero
5  edadSac;0;300;-1;-2;Entero
6  sexo;1;2;-1;-2;Entero
7  codi_inia_uab;0;1000000000;-1;-2;Entero
8  raza;0;10;-1;-2;Entero
9  lote;0;20;-1;-2;Entero
10 pesoNeto;0;200;-1;-2;Flotante
11 edad;0;300;-1;-2;Entero
12 peso;0;150;-1;-2;Flotante
13 tocino;0;100;-1;-2;Flotante
14 padre;0;1000000000;-1;-2;Entero
15 madre;0;1000000000;-1;-2;Entero
```

Figura 1: Ejemplo de archivo de Tipo de datos.

Creación del archivo Características del individuo

En este punto se describirá el archivo de Características del individuo. Éste contendrá las características de los individuos que no sean genotipos ni fenotipos. La información en éste archivo debe estar distribuida de la siguiente forma:

1. En la primera línea debe estar escrito “Individuo” que será el que identifique que se va a subir un archivo con las características de éste.
2. En la segunda línea tiene que haber el número de Tipo de datos que hay en la siguiente fila.
3. En esta línea están los Tipos de datos que van a pertenecer a las características de los individuos. Se tiene que tener en cuenta que el primer dato debe ser el identificador del individuo, que además debe ser numérico. También se debe recordar que los datos padre y madre son palabras reservadas y que en este archivo se utilizarán para identificarlos, por tanto deben existir en la base de datos.
4. Una vez se ha puesto la información anterior se puede empezar a colocar los datos referentes a cada individuo.

Como en el archivo de Tipos de datos, si se produce un error no se subirá ninguna fila. Además hay que tener en cuenta que la información debe estar separada por punto y coma.

También se tiene que recordar que los Tipos de datos que se utilicen en este fichero deben estar ya definidos en la aplicación.

En la Figura 2 se puede observar un ejemplo de archivo de Características del individuo. Como se puede comprobar se mantiene la estructura antes descrita y se tiene que el primer dato es el identificador que dice el tipo de archivo que se va a subir. En la segunda línea se puede ver que está el número de tipos de datos que hay en la siguiente línea. En la tercera línea están los tipos de datos, que como se puede ver, son los que se han definido anteriormente en el archivo de *Tipos de datos*. También se puede comprobar que el primer dato es el identificador del individuo, en este caso es el tatuaje, y que a partir de éste se definen las características que pertenecen al individuo.

```
1 Individuo
2 11
3 tatuaje;tipo;dataNaix;edadSac;sexo;codi_inia_uab;raza;lote;pesoNeto;padre;madre
4 12603177;animal;19960623;0;1;12603177;3;1;85.3;-1;-1
5 12603062;animal;19960623;0;2;12603062;3;1;84.6;-1;-1
6 704851;animal;19970623;175;1;704851;3;2;80.16;12603177;12603062
7 704852;animal;19970623;175;1;704852;3;2;84.77;12603177;12603062
8 704853;animal;19970623;175;1;704853;3;2;79.87;12603177;12603062
9 704854;animal;19970623;175;1;704854;3;2;89.95;12603177;12603062
10 704855;animal;19970623;175;2;704855;3;2;73.34;12603177;12603062
```

Figura 2: Ejemplo de archivo de Características del individuo.

Creación del archivo de Fenotipos

En esta sección se describirá el archivo de Fenotipos. La información en éste archivo debe estar distribuida de la siguiente forma:

1. En la primera línea debe estar escrito la palabra “Fenotipo”, que será la que diga que el archivo que se va a subir es de fenotipos.
2. En esta línea, como en el archivo de *Características del individuo*, se tiene que poner el número de tipos de datos que se va a utilizar en la siguiente línea.
3. En esta línea van los tipos de datos que se utilizarán en este fichero. Hay que tener en cuenta que el primer dato debe ser el identificador del individuo, el segundo dato debe ser el número de experimento y el tercer dato el lote al cual pertenece la muestra. Una vez se ha puesto esta información se puede poner los tipos de datos que se utilizarán.
4. A partir de esta línea ya se pueden poner todos los datos referentes a los fenotipos.

Como en los archivos anteriores, si se produce un error no se subirá ninguna fila. Además hay que tener en cuenta que la información debe estar separada por punto y coma y que todos los tipos de datos deben estar definidos en la aplicación, menos el experimento y el tiempo que como señalamos anteriormente ya están definidos en la base de datos.

En la Figura 3 se puede observar un ejemplo de archivo de Fenotipos. Como se puede ver se mantiene la estructura que se ha descrito al principio de este punto. Así se puede ver que en la primera línea está la palabra Fenotipo que es la que indica que se va a subir un archivo de fenotipos. En la siguiente línea se tiene la cantidad de datos que se va a utilizar. En la tercera línea se tienen los tipos de datos. Se debe prestar

atención a los tres primeros datos que, como se ha mencionado anteriormente, deben ser el identificador del individuo, el experimento y el tiempo.

```
1 Fenotipo
2 6
3 tatuaje;experimento;tiempo;edad;peso;tocino
4 704851;1;175;164;103.4;18.5
5 704851;1;176;172;111;19
6 704851;1;177;164;108.5;17
7 704852;1;175;172;113;16.5
8 704852;1;176;164;99.5;16.5
9 704852;1;177;172;110.5;17
10 704853;1;176;164;113.5;20
11 704853;1;177;172;119;20.5
12 704854;1;176;164;91.5;16
13 704854;1;177;172;97.5;16.5
14 704855;1;177;173;98.5;17.5
```

Figura 3: Ejemplo de archivo de Fenotipos.

Creación del archivo de Genotipos

En este punto se describirá el archivo de Genotipos. La información en éste archivo debe estar distribuida de la siguiente forma:

1. En la primera línea debe estar escrita la palabra “Genotipo”, que será la que diga que el archivo que se va a subir es de genotipos.
2. En esta línea, como en los archivos precedentes, se debe poner el número de tipos de datos que se va a utilizar en la siguiente línea.
3. En esta línea van los tipos de datos que se utilizarán en este fichero. Hay que tener en cuenta que el primer dato debe ser el identificador del individuo y a continuación los tipos de datos que se van a utilizar.
4. En esta línea va el número de marcadores que se va a utilizar en el archivo. Estos marcadores deben estar ya definidos en la base de datos.
5. En esta línea van los nombres de los marcadores que se utilizarán.
6. A partir de esta línea ya se pueden poner todos los datos referentes a los genotipos.

Como en los archivos anteriores, si se produce un error no se subirá ninguna fila. Además hay que tener en cuenta que la información debe estar separada por punto y coma y que el primer tipo de dato debe ser el identificador del individuo.

En la Figura 4 se puede observar un ejemplo de archivo de Genotipos. Como se puede ver se mantiene la estructura que se ha descrito al principio de este punto. Así se tiene que en la primera línea esta la palabra Genotipo que es la que indica que se va a subir un archivo de genotipos. En la siguiente línea se tiene la cantidad de datos que se van a utilizar. En la tercera línea se tienen los tipos de datos, y como se puede observar el primer tipo de dato es el identificador del individuo. En la cuarta línea está el número de marcadores que se van a utilizar. En la quinta línea están los nombres de los marcadores utilizados. A partir de esta línea están los datos que se insertarán en la base de datos.

```
1 Genotipo
2 3
3 tatuaje;microsatelites;microsatelites
4 11
5 ASGA00000005;ALGA00000009;ALGA00000014;ASGA00000014;ASGA00000021;H3GA0
6 704851;C;NA;A;G;T;C;A;C;G;T;A;G;A;C;G;G;C;C;G
7 704852;C;C;A;G;T;C;A;C;G;T;A;G;A;C;G;G;C;C;G
8 704853;C;C;A;G;T;C;A;C;G;T;A;G;A;C;G;G;G;C;G
9 704854;C;C;A;G;T;C;A;C;G;NB;NA;G;A;C;G;G;C;C
10 704855;A;C;A;G;T;C;A;C;G;T;A;G;A;C;G;G;T;C;G
```

Figura 4: Ejemplo de archivo de Genotipos.

Creación del archivo de Marcadores

En este punto se describirá el archivo de Marcadores. La información en éste archivo debe estar distribuida de la siguiente forma:

1. En la primera línea debe estar escrito la palabra “Marcador”, que será la que diga que el archivo que se va a subir es de marcadores.
2. En esta línea, como en los archivos precedentes, se tiene que poner el número de tipos de datos que se van a utilizar en la siguiente línea.
3. En esta línea van los tipos de datos que se utilizarán en este fichero. Hay que tener en cuenta que el primer dato debe ser el Nombre del marcador, el segundo el cromosoma, el siguiente la posición física relativa, el cuarto la posición genética, el quinto dato debe ser la clase, el sexto el nivel y el último dato debe ser el comentario del marcador.
4. A partir de esta línea ya se pueden poner todos los datos referentes a los marcadores.

Como en los archivos anteriores, si se produce un error no se subirá ninguna fila. Además se tiene que tener en cuenta que la información debe estar separada por punto y coma.

En la Figura 5 se puede observar un ejemplo de archivo de Marcadores. Como se puede ver se mantiene la estructura que se ha descrito al principio de este punto. Así se tiene que en la primera línea esta la palabra Marcador que es la que nos indica que se va a subir un archivo de marcadores. En la siguiente línea se tiene la cantidad de datos que se van a utilizar. En la tercera línea se tienen los tipos de datos y a partir de la cuarta línea se tienen los datos referentes a los marcadores que se insertarán en la base de datos.

```
1 marcador
2 7
3 NombreMarcador;chr;bp;cM;clase;nivel;comentario
4 ASGA0000005;1;27137;0;snp;5;Comentario
5 ALGA0000009;1;52297;0;snp;5;Comentario
6 ALGA0000014;1;79763;0;snp;5;Comentario
7 ASGA0000014;1;126385;0;snp;5;Comentario
8 ASGA0000021;1;152975;0;snp;5;Comentario
9 H3GA0000026;1;173251;0;snp;5;Comentario
10 ALGA0000021;1;210181;0;snp;5;Comentario
11 INRA0000015;1;250695;0;snp;5;Comentario
12 H3GA0000032;1;268032;0;snp;5;Comentario
13 ALGA0000022;1;286414;0;snp;5;Comentario
14 M1GA0000060;1;310320;0;snp;5;Comentario
```

Figura 5: Ejemplo de archivo de Marcadores.

Creación del archivo de Lotes

En este punto se describirá el archivo de Lotes. La información en éste archivo debe estar distribuida de la siguiente forma:

1. En la primera línea debe estar escrito la palabra “lote”, que será la que diga que el archivo que se va a subir es de lotes.
2. A partir de esta línea se insertarán los datos referentes a los lotes. El primer dato será el identificador del proyecto, el segundo será el lote (debe ser numérico), y el tercer dato será el comentario del lote.

Como en los archivos anteriores, si se produce un error no se subirá ninguna fila. Además se tiene que tener en cuenta que la información debe estar separada por punto y coma.

En la Figura 6 se puede observar un ejemplo de archivo de Lotes. Como se puede ver se mantiene la estructura que se ha descrito al principio de este punto. Así se tiene que en la primera línea esta la palabra lote que es la que indica que se va a subir un archivo de lotes. En las siguientes líneas se tienen los datos que se insertarán en la base de datos, y que pertenecen a cada lote.

```
1 lote
2 Proyecto1;175;Esto es el comentario
3 Proyecto1;176;Esto es el comentario
4 Proyecto1;177;Esto es el comentario
```

Figura 6: Ejemplo de archivo de Lotes.

Anexo II: Manual de usuario

Introducción

En este manual se intentará dar una idea general de cómo empezar a trabajar con la aplicación. Este anexo no pretende ser un manual muy extenso, ya que todas las partes se han ido explicando durante toda la memoria, y por lo tanto solo se centrará en los puntos más importantes de la aplicación.

Inicio

Una vez el usuario se ha identificado en la aplicación entra a la página principal o a la página de perfil, dependiendo de si se han rellenado o no los campos obligatorios. Aquí se supondrá que ya se han rellenado todos los campos y que se accederá por tanto a la página principal.



Figura 1: Página principal.

Una vez identificado aparecerá el menú en el cual están todas las actividades que se pueden realizar con la aplicación. Como se puede apreciar, en el menú del administrador aparece una sección más, esta es exclusiva del administrador. A continuación se describirán todas ellas.

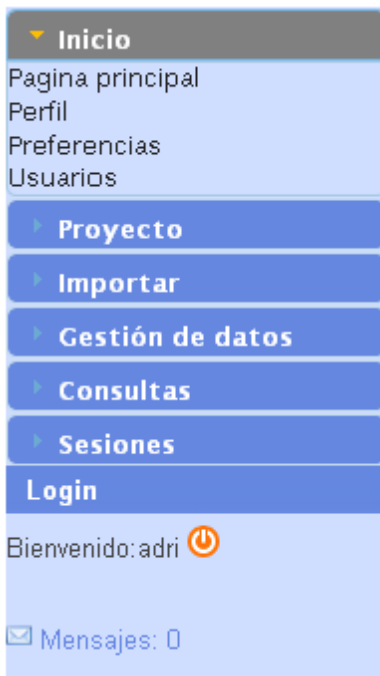


Figura2: Menú del usuario.

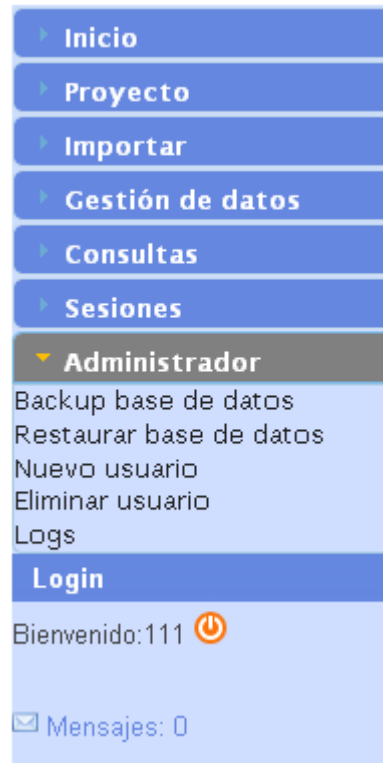


Figura 3: Menú del administrador.


Zona de inicio:

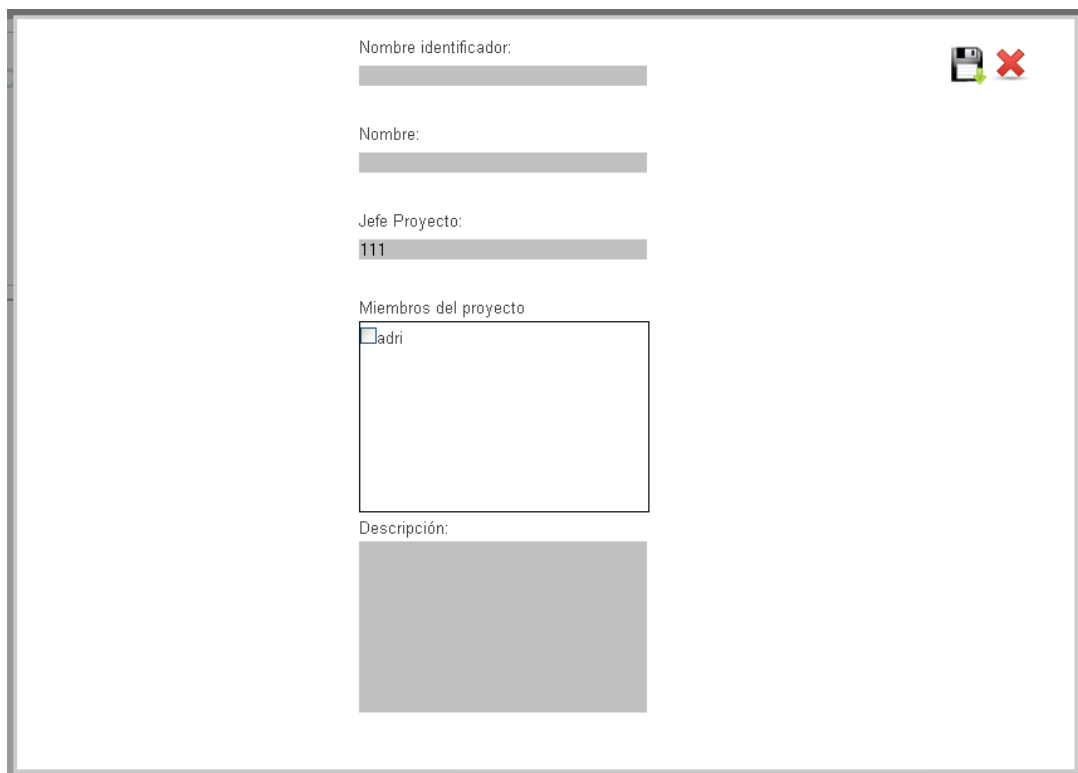
- **Página principal:** Pulsando sobre este enlace se irá a la página inicial de la aplicación.
- **Perfil:** En esta opción se podrá modificar el perfil de usuario.
- **Preferencias:** Aquí se podrá cambiar las preferencias que tiene el usuario.
- **Usuarios:** En esta zona se podrán enviar mensajes entre los usuarios, tanto mensajes internos como emails.

Zona de proyecto:

Esta zona es importante ya que es donde se crearán los proyectos con los cuales se trabajará posteriormente. Los puntos de esta zona son:

- **Proyectos:** En esta sección se podrán crear o eliminar proyectos, además también se podrán ver los proyectos a los que pertenece el usuario.
- **Usuarios:** Aquí se podrán añadir o quitar usuarios de los proyectos.
- **Modificar Proyecto:** En esta zona se podrán modificar las características de los proyectos.

Para crear un nuevo proyecto solo se tendrá que pulsar sobre el icono  y rellenar el formulario que se presenta en la página. Si el nombre identificador del proyecto no existe se creará un proyecto con éxito, por el contrario si ya existe se mostrará un mensaje de error.



Formulario de creación de proyecto con los siguientes campos:


- Nombre identificador:
- Nombre:
- Jefe Proyecto:
- Miembros del proyecto: adri
- Descripción:

En la esquina superior derecha del formulario hay un icono de documento con un signo de plus (+) y un icono de documento con un signo de X (rojo).

Figura 4: Formulario a rellenar.

Zona de Importar:

- **Subir archivo:** En esta actividad se podrán elegir los archivos que se quieren subir a la aplicación.
- **Subir Lote:** Se podrá seleccionar el archivo de lotes que se va a insertar en la base de datos.

Esta es otra de las zonas importantes. Aquí se podrán subir los archivos con el formato descrito en el Anexo I. Para ello solo se tendrá que seleccionar el archivo que se quiere subir y pulsar sobre el icono . A partir de aquí tan solo hay que seguir las instrucciones que se presentan en pantalla.

Zona de Gestión de datos:

- **Insertar Tipo de datos, Lote, Individuo, Fenotipo, Genotipo, Marcador:** En estas opciones se podrán insertar a mano todos estos datos.
- **Modificar o eliminar Tipo de datos, Lote, Individuo, Fenotipo, Genotipo, Marcador:** En esta sección se podrán modificar o eliminar todos estos datos.

Esta zona realiza las mismas acciones que el punto anterior, con la única diferencia que en esta habrá que poner la información a mano.

Zona de Consultas:

- **Consultar Tipo de datos, Individuos, Fenotipos, Genotipos:** Aquí se podrán consultar todos los datos referentes a estos.
- **Eliminar Consulta:** Se eliminará la consulta actual.

Esta es otra de las zonas importantes ya que es aquí donde se presenta la información que se ha introducido anteriormente. Lo más destacable que hay que tener en cuenta de esta zona es que mientras se va avanzando la consulta se va guardando en la sesión, y por lo tanto para retroceder hay que utilizar el Breadcrumb o se tiene que eliminar la consulta.



Figura 5: Breadcrumb.

Zona de Sesiones:

- **Guardar Sesión:** Aquí se podrá guardar la sesión con la que se está trabajando.
- **Recuperar Sesión:** Se recuperará la sesión anteriormente guardada.
- **Enviar sesión:** Se podrán enviar las sesiones al resto de usuarios de la aplicación.
- **Recuperar sesiones recibidas:** Se recuperarán las sesiones que se han enviado entre los usuarios.

A continuación se describirá la zona exclusiva del administrador.

Zona de Administrador:

- **Backup base de datos:** Se podrá descargar una copia de la base de datos a un archivo sql.
- **Restaurar base de datos:** Se restaurará la versión de la base de datos anterior.
- **Nuevo usuario:** Aquí el administrador podrá dar de alta a nuevos usuarios.
- **Eliminar usuario:** En esta zona el administrador podrá dar de baja a los usuarios.
- **Logs:** Aquí se presentarán las acciones que han realizado los usuarios en la aplicación.

Anexo III: Manual de instalación

Introducción

En este Anexo se explicará cómo instalar la aplicación. El instalador ha sido preparado con *mkInstaller*, que permite crear un instalador de un sitio web y además permite definir variables que serán utilizadas para la configuración del mismo. Esta aplicación tiene unos manuales muy completos en su página web oficial.

Requerimientos del software

Para que el software funcione correctamente se debe ejecutar en un servidor que tenga los siguientes requerimientos:

- Servidor con PHP instalado y habilitado el modulo de MySQL.
- Tener R instalado con el modulo de gráficos.
- Tener un administrador de MySQL, como PHPMyAdmin, para gestionar la base de datos.
- Tener instalado Perl para el uso de CGI.

Instalación de la aplicación

Con esta memoria se entregará un archivo llamado *instalador.php* con el que se podrá instalar la aplicación en cualquier servidor que soporte PHP.

Una vez se ha puesto el archivo en la ubicación en la que se quiere instalar la aplicación se debe ejecutar en cualquier navegador. Entonces aparecerán unos campos que hay que rellenar.

Instalador de la aplicación

En esta página se preguntan las variables que serán utilizadas en la aplicación, como podrían ser el usuario y contraseña de la base de datos.

Dirección de la instalación de R <i>R_command</i> :	<input type="text" value="/home/samba/homes/1"/>
Dirección en la que se esta haciendo la instalación <i>direccion</i> :	<input type="text" value="/home/samba/homes/1"/>
Dirección de web de la aplicación <i>direccionweb</i> :	<input data-bbox="821 1751 1050 1780" type="text" value="https://158.109.240.122/"/>
Host del email utilizado para la aplicación <i>Host</i> :	<input type="text" value="ssl://smtp.gmail.com"/>
Puerto que utiliza el servidor de emails <i>Port</i> :	<input type="text" value="465"/>
Usuario del email <i>Username</i> :	<input type="text" value="deptveterinaria@gmail."/>

Figura 1: Campos a rellenar.



Estos campos se describen a continuación:

- **Dirección de la instalación de R:** Aquí se deberá poner la ruta en la cual está instalado el programa R. Por ejemplo `home/samba/homes/opt/R/bin/R`
- **Dirección en la que se está haciendo la instalación:** Hay que indicar la dirección en la cual se encuentra ubicado el archivo de instalación, y que por tanto albergará la aplicación final. Por ejemplo `/home/samba/homes/public_html/`
- **Dirección web de la aplicación:** Aquí se debe escribir la dirección web que tendrá la aplicación. Por ejemplo `https://158.109.240.122/`
- **Host del email utilizado para la aplicación:** Se debe poner el host que se utilizará para el envío de emails. Por ejemplo `ssl://smtp.gmail.com`.
- **Puerto que utiliza el servidor de emails:** El puerto que utiliza el host anterior. Por ejemplo 465.
- **Usuario del email:** Usuario que se utilizaría para entrar al email. Por ejemplo `deptveterinaria@gmail.com`.
- **Contraseña del email:** Contraseña del email asociada al usuario anterior.
- **Dirección del email:** Dirección que se utilizará, esta puede ser igual que el usuario del email. Por ejemplo `deptveterinaria@gmail.com`
- **Email al cual se enviarán las peticiones de registro:** Aquí se debe poner el email del administrador que se encargará de dar de alta a nuevos usuarios.
- **Nombre de la base de datos:** Nombre que tendrá la base de datos dentro del sistema.
- **Host de la base de datos:** Host en el cual está ubicada la base de datos.
- **Usuario de la base de datos:** Usuario que se utilizará para entrar en la base de datos.
- **Contraseña de la base de datos:** Contraseña que se utilizará para entrar en la base de datos conjuntamente con el usuario anterior.

Una vez se han rellenado todos los campos, se tiene que pulsar el botón de instalar y a continuación se podrá ver cómo se van creando los diferentes archivos y carpetas, además de ver cómo se va generando la base de datos.

Instalador de la aplicación

En esta página se preguntan las variables que serán utilizadas en la aplicación, como podrían ser el usuario y contraseña de la base de datos.

```
Operando ALTER TABLE `Lote` ADD CONSTRAINT `Foreign_m_experimento` FOREIGN KEY ...
Operando ALTER TABLE `LoteFenotipo` ADD CONSTRAINT `Foreign_if_lote` FOREIGN KEY ...
Operando ALTER TABLE `LoteIndividuo` ADD CONSTRAINT `Foreign_loteIndiv_lote` FOREIGN KEY ...
Operando ALTER TABLE `Preferencias` ADD CONSTRAINT `Foreign_pref_user` FOREIGN KEY ...
Operando ALTER TABLE `TablaBD` ADD CONSTRAINT `Foreign_tablaBD_usuario` FOREIGN KEY ...
Instalación de ficheros y bdd completa
Tiempo total: 22 segundos
```

mKinstaller

Figura 2: Final de la instalación.

Una vez ha terminado de crear todos los archivos y carpetas y generada la base de datos, se debe ejecutar el archivo `parse.php` que terminará de configurar la aplicación con la información que se ha proporcionado anteriormente.

Configurador de la aplicación

```
Insertando datos de R...
Insertando Menu...
Insertando datos conexión a la base de datos...
Insertando datos email de la aplicación...
Insertando Menús de ayuda...
Insertando Cabecera de la aplicación...
Insertando Mensaje inicial...
Insertando Restaurar base de datos...

Configuración finalizada.
```

Figura 3: Configurador de la aplicación.

Una vez se han hecho estos pasos ya se tiene instalada la aplicación. Si se desea se puede borrar todos los archivos a excepción de la carpeta `public_html`.

El abajo firmante, **Adrián Forte Romera**, alumno de la Escuela de Ingeniería de la UAB,

CERTIFICA:

Que ha realizado la presente memoria y el proyecto “Desarrollo de una plataforma bioinformática para la gestión de datos fenotípicos y moleculares”.

Y para que conste firma la presente.

Sabadell, Junio del 2010

Firmado: Adrián Forte Romera

NIU: 1146654

