GenLab

# Resultado de imagen de ucm logo

**Facultad de Informática**

**Trabajo Fin de Grado 2017/2018**

**Tutor:** Rubén Fuentes Fernández (Dpto. de Ingeniería del

Software e Inteligencia Artificial de la Facultad de Informática)

**Autores**

Pablo Arranz Ropero (GIS)

Juan Alberto Camino Sáez (GIS)

Carlos López Martínez (GII)

Autorización de difusión y utilización

Dedicatorias

Agradecimientos

Índice

1. Índice de imágenes
2. Índice de diagramas
3. Resumen
4. Abstract
5. Introducción
6. Manuales
   1. Manual de usuario
   2. Manuales del desarrollador
      1. Manual de extensión de la aplicación móvil
      2. Manual de extensión de la aplicación de administración
7. Apéndices
8. Resultados
9. Conclusiones
10. Bibliografía
11. Anexos
12. Glosario

Índice de imágenes

Índice de diagramas

Resumen

Abstract

Introducción

Manuales

## Manual de usuario

En esta sección se van a explicar las distintas acciones que debe hacer un usuario para el correcto manejo de la aplicación móvil.

Al abrir la aplicación se nos mostrará la pantalla de bienvenida donde deberemos introducir nuestro nombre para que la aplicación recoja cierta información asociada a nuestro perfil (por ejemplo, para dar feedback al administrador dentro de la aplicación del servidor).

[imagen login]

Una vez dentro de la aplicación, se nos mostrará una serie de secciones (herramientas de cálculo, problemas, tests, teoría y libros recomendados) adscritas a una aplicación en concreto y que podemos consultar pinchando en ellas. En esta pantalla además se nos muestra al igual que durante toda la navegación, un menú desplegable para seleccionar una aplicación (One Locus, Two Loci, Polyhybrid, Linkage o Epistasia) en concreto, dentro de la cual están las mismas secciones nombradas anteriormente.

[imagen main view]

[imagen main view con menú desplegado]

Ahora trataremos los distintos apartados disponibles a consultar dentro de una aplicación específica:

Herramientas de cálculo (calculation tools):

En este apartado veremos un listado de las diferentes herramientas de cálculo disponibles para la aplicación seleccionada.

[imagen lista CT]

Al pinchar sobre una de ellas se nos mostrará una pantalla con la herramienta de cálculo, dentro de la cual estarán nos encontramos con distintos inputs donde introduciremos los datos y una vez le demos a calcular se nos mostrarán los distintos valores de los resultados hallados.

[imagen de una CT antes y después de darle a calcular]

Dentro de esta pantalla también hay un botón para borrar los resultados.

Problemas:

En esta pantalla se nos muestran listados todos los problemas propuestos dentro de la aplicación escogida, dentro de cada problema se plantearán distintas cuestiones que el usuario si lo desea deberá solucionar (no a través de la aplicación móvil).

[imagen de problemas]

Tests:

Al igual que con las herramientas de cálculo, aparecerá una lista de los distintos tests disponibles para la aplicación escogida. Al entrar dentro de uno de estos test se presentarán las distintas preguntas con sus respuestas a responder. El usuario deberá responder la respuesta que crea que es correcta pinchando sobre ella, dicha respuesta se mostrará en rojo si es incorrecta o en verde si es correcta.

[imagen listado tests]

Dentro de los tests puede haber la posibilidad de cuestiones multirrespuesta si el administrador decide que así sea, en ese caso el comportamiento será parecido a si fuese de una única respuesta.

[imagen de un test con alguna pregunta respondida]

Teoría:

En este apartado se mostrará la teoría correspondiente a la aplicación elegida junto con imágenes explicativas si el administrador así lo decide. Esta teoría explicará los distintos conceptos incorporados dentro de la aplicación.

[imagen de ejemplo de teoría]

Libros recomendados:

Aquí se presentarán los distintos libros recomendados para poder entender la aplicación elegida, ya que tienen relación con los temas tratados dentro de ella; la información mostrada es el título del libro, el autor, la editorial, el código ISBN y el enlace a dicho libro.

[imagen libros recomendados]

En definitiva, estos son los distintos apartados que podemos encontrar dentro de cada aplicación.

A parte de lo tratado anteriormente, dentro del menú desplegable encontramos un botón de logout para cerrar la sesión de la app y un botón para volver atrás entre acciones.

[imagen conjunta logout y botón de atrás]

## Manual del desarrollador

Manual de extensión de la aplicación móvil

En este apartado se procede a explicar todos los detalles de programación y la estructura que se ha seguido para programar la aplicación móvil, así como las instrucciones necesarias para llevar a cabo una ampliación o extensión de la aplicación.

Para llevar a cabo esta aplicación, se ha utilizado Apache Cordova. Esto es debido a que permite diseñar una aplicación móvil mediante tecnologías web, por lo que se tiene acceso a amplios recursos que HTML, CSS y JavaScript permiten (como el diseño Responsive, para permitir la correcta visualización independientemente del tamaño de la pantalla, fundamental para una aplicación móvil, donde existen múltiples tamaños del dispositivo) y, además, porque permite generar a partir del código los distintos ejecutables para cada sistema operativo (como Android o IOS).

INSTALACIÓN DE PROGRAMAS NECESARIOS

Para instalar Apache Cordova en el ordenador, hay que seguir los siguientes pasos:

* Instalar Node.js, ya que esta aplicación se basa en su uso. Se puede descargar a través del siguiente enlace <https://nodejs.org/en/>
* Una vez instalado, es necesario tener un cliente de Git, ya que, aunque no es obligatorio su uso, el programa descarga algunos paquetes de esta plataforma al crear un nuevo proyecto. Puedes descargarlo a través del siguiente enlace <https://git-scm.com/downloads>
* Por último, para descargar el Cordova, hay que hacerlo mediante el instalador de paquetes de Node.js npm. Se realiza mediante línea de comandos, dependiendo del sistema operativo que use el ordenador. Por ejemplo, en el caso de Windows, el comando sería el siguiente: **npm install -g cordova**

Tras esto, ya estará instalado Apache Cordova en el ordenador, y ya podremos ejecutar los comandos para crear los ejecutables de la aplicación para el dispositivo móvil.

Para hacer la depuración y la codificación más sencilla, también será necesario algún editor de texto especializado en tecnologías web (como Visual Studio Code) y un navegador actualizado para poder probar la aplicación (como Google Chrome). Esto, aunque muy recomendable, no es obligatorio.

ESTRUCTURA DE LA APLICACIÓN

En este apartado se procede a explicar la estructura que se ha llevado a cabo para desarrollar la aplicación.

Como se ha citado anteriormente, para desarrollar la aplicación se ha usado Apache Cordova, por lo que la estructura inicial es la de un proyecto Cordova. En la imagen x.x se muestra la estructura del proyecto. Sin embargo, la carpeta donde se encuentra el código de la aplicación se encuentra en la carpeta www, el resto lo autogenera Cordova al crear el proyecto y al hacer el build de la aplicación.

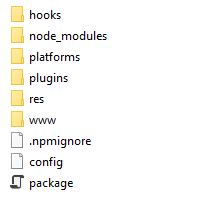


Imagen x.x. Estructura del proyecto Cordova.

Dentro de la carpeta www, se encuentra el código HTML, CSS y JavaScript con el que se ha diseñado la aplicación, y cuya estructura de carpetas, representada en la imagen X.X, se procede a explicar a continuación:

Por un lado, tenemos un único fichero HTML llamado *index*.*html*, el cuál contiene todos los elementos de la aplicación. Solo hay un único archivo debido a que se ha decidido usar el modelo SPA (Single Page Application) para que de esta manera la carga de la página y de sus recursos se realice una única vez al iniciar la aplicación, y de esta manera se reduce los tiempos de carga para favorecer la interacción con el usuario. A medida que el usuario navega por las distintas secciones, se muestran u ocultan los elementos correspondientes.

Por otro lado, tenemos dos carpetas. Una de ellas, *resources*, contiene todos los recursos que usa la aplicación (dentro de *css* se encuentran las hojas de estilo propias, dentro de *bootstrap* las hojas de estilo de Bootstrap descargadas de su página web, dentro de *img* las imágenes y, por último, dentro de *js* tanto los scripts propios como scripts de JQuery, que también se ha utilizado para manejar distintos aspectos en la aplicación. La otra carpeta llamada *ctools* contiene, a su vez, otras carpetas correspondientes a cada sección de la aplicación, y, en su interior, ficheros HTML donde se encuentran definidas las calculation tools de cada aplicación en concreto (se explican mas detalles sobre esto más adelante).

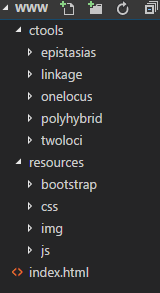


Imagen X.X. Estructura de la carpeta www que contiene el código de la app

EXTENSIÓN DE LA APLICACIÓN

Es posible implementar mejoras o añadir nuevos elementos en un futuro a la aplicación. Para ello, es necesario diferenciar si se quiere añadir elementos generales, o, sin embargo, se desea añadir una nueva calculation tool, ya que es necesario seguir una serie de pasos adicionales para facilitar la mantenibilidad y la modularización de las mismas.

Para añadir nuevos elementos, simplemente hay que añadirlos en el archivo index.html, identificados mediante la etiqueta *id* de HTML, ya que como se explica anteriormente, se usa el modelo SPA y se controla la visibilidad de los elementos mediante los scripts a medida que el usuario navega a través de la aplicación. Es posible, si ese es su deseo, el añadir nuevos ficheros HTML y llamarlos mediante un *href*, aunque no es recomendable.

Para manejar su estilo, se pueden crear nuevas hojas de estilo CSS y añadirlas mediante la etiqueta *link* al HTML principal, y poner los estilos que se deseen, aunque es mejor si se utilizan algunas de las hojas de estilo ya creadas anteriormente. Hay tres hojas de estilo creadas: *styles.css* tiene todos los estilos correspondientes a toda la aplicación en general, *form-elements.css* contiene estilos propios del formulario de acceso a la aplicación y *ctools.css* tiene todos los estilos referentes a las calculation tools que existen en la aplicación.

Para los scripts, sucede igual que en caso anterior. Es posible añadir nuevos scripts mediante la etiqueta link al HTML principal, aunque es recomendable usar los scripts ya creados anteriormente. En este caso, tenemos home.js, donde se encuentra el comportamiento de la aplicación en general, y ctools.js donde se encuentra el comportamiento de las calculation tools, así como las peticiones AJAX para pedir los resultados al servidor (donde se encuentran alojadas las fórmulas de cálculo).

En estos casos, es posible añadir, eliminar o editar los elementos sin ningún tipo de problema y sin ningún tipo de limitación, respetando la estructura de diseño proporcionada para permitir que nuevos desarrolladores puedan modificar la aplicación.

En el caso de las calculation tools, el proceso a seguir es distinto, ya que al usar un modelo SPA y, debido a las múltiples calculation tools de las que se dispone actualmente o incluso de las que se puede disponer en un futuro, y, además, cada una tiene sus propios elementos y estilos diferentes, introducir todo el código de las distintas calculation tools dentro del archivo principal index.html haría que resulte en un fichero extremadamente grande y complicado de mantener y controlar, por lo que se ha decidido cargar su contenido desde ficheros HTML externos que se insertan en el HTML principal (de esta manera se evita la carga de recursos adicionales, ya que solo se inserta el código HTML de las calculation tools , ya que su estilo se ha cargado anteriormente).

Estos ficheros externos, cada uno correspondiente a una única calculation tools, se encuentran en la carpeta ctools

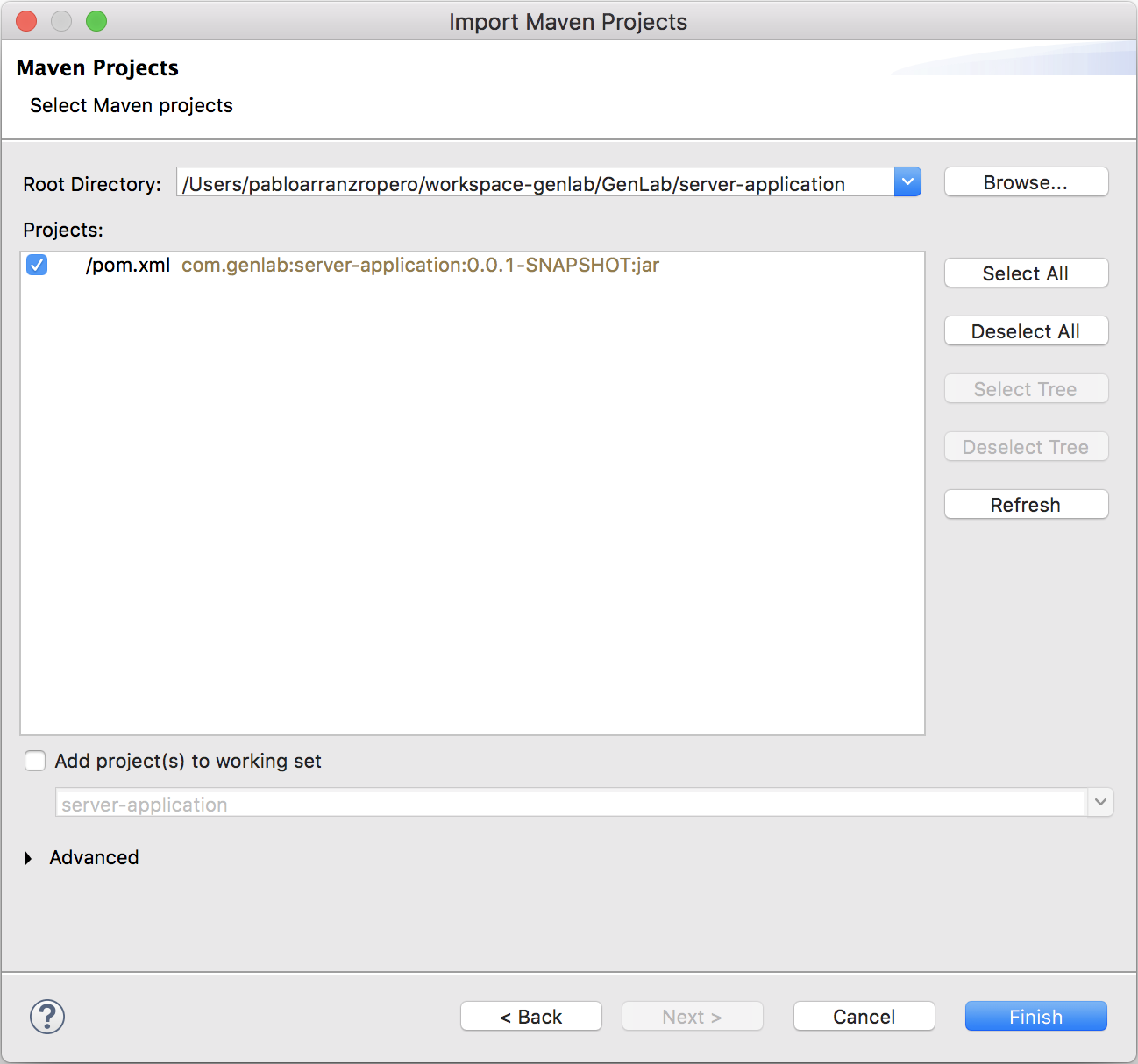
Manual de extensión de la aplicación de administración

A continuación, se procederá a explicar el procedimiento a seguir para la extensión de la parte servidor de la aplicación. Los detalles de su implementación (véase lenguajes de implementación, frameworks y patrones de diseño utilizados) están recogidos en la sección <link y referencia>.

Para comenzar a desarrollar este proyecto, debemos instalar en nuestro ordenador una versión de Java igual o superior a la 1.8 (referencia a <https://www.java.com/es/download/>) y Maven (referencia a <https://maven.apache.org/>). Una vez instalado Maven, mediante la consola, nos dirigiremos a la carpeta donde esté nuestro proyecto (donde deben estar los archivos *mvnw* y *pom*) y ejecutaremos el comando *mvn clean install* para instalar todas las dependencias necesarias.

Por último, debemos instalar Lombok, una librería de Java que permite simplificar al máximo los objetos que contienen los datos de nuestra aplicación. (referencia a <https://projectlombok.org/download>)

Seguidamente, importaremos el proyecto al IDE del que estemos haciendo uso.

*Ilustración x.x: Importación de proyecto a STS*

El proyecto está desarrollado siguiendo el patrón MVC y una arquitectura multicapa (referencia a donde expliquemos patrones usados) por lo que encontraremos el proyecto distribuido, a grandes rasgos, de la siguiente manera:

* Controllers: Contiene todos los manejadores de rutas de nuestra aplicación web y de la API REST. Realiza la parte de Controlador, como su nombre indica, en el patrón MVC y forma parte de la capa de presentación.
* Models: Contiene todas las entidades y objetos de datos de la aplicación. Realiza la parte de Modelo en el patrón MVC y forma parte de la capa de datos y de la capa de vista.
* Repositories: Contiene los repositorios de la aplicación, es decir, son las clases que accederán a la base de datos cuando sea necesario. Forma parte del Modelo en el patrón MVC y forma parte de la capa de integración en nuestra arquitectura.
* Services: Contiene los servicios de aplicación donde se realizan las operaciones necesarias sobre los datos que llegan desde el controlador. Hacen de intermediarios entre la capa de presentación y la de integración. Forma parte del Modelo en el patrón MVC y de la capa de negocio en nuestra arquitectura.
* Config, Interceptor y Utils: Estos paquetes contienen utilidades necesarias para la configuración o para la intercepción de peticiones HTTP.
* Resources: Contiene las plantillas creadas en HTML, utilizando Thymeleaf (referencia). También contiene los archivos CSS y JavaScript necesarios para la construcción de la vista. Forma parte de la Vista en el patrón MVC y de la capa de presentación en nuestra arquitectura.

Añadiendo una nueva sección a nuestra aplicación

En el momento de la publicación de este proyecto, las secciones existentes son las descritas en (sección y referencia). Para añadir una nueva sección en nuestra aplicación será tan sencillo como añadir la sección en el fichero *application.yml* encontrado en el directorio *src/main/resources* y añadir el nombre de la sección, su número identificador y su prioridad en la tabla sections de la base de datos. Por último hará falta añadir la sección con su id en el la enumerado llamado *SectionsMapping* encontrado en el paquete *Models.* No hará falta añadir más información acerca de la sección y ya existirá en la aplicación y será visible para los alumnos (aunque vacía por el momento).

Añadiendo un nuevo módulo a nuestra aplicación

En el momento de la publicación de este proyecto, los módulos existentes son los descritos en (sección y referencia). Para añadir un nuevo módulo y mantener la modularidad y arquitectura de nuestra aplicación deberemos seguir el siguiente procedimiento:

1. Crearemos, en el paquete *controllers* una nueva clase que mapee todos los endpoints necesarios para gestionar ese módulo. Para mantener el estilo del código, la propia clase mapeara las URLs que empiecen por “/<nombre-modulo>”, por ejemplo, en el módulo tests, la clase TestsController mapeará todas las URLs que empiecen por “/tests” y cada método interno mapeará una URL que haga referencia a las acciones que realiza dicho método.
2. Crearemos, en el paquete *models*, los objetos necesarios para este nuevo módulo. Estos objetos podrán ser entidades que se volcarán a la base de datos u objetos que utilice la vista (También se podría implementar el patrón Transfer si fuera necesario, aunque nosotros no hayamos hecho uso de este). Para la creación de estos objetos de manera simple, recomendamos el uso de las anotaciones proporcionadas por lombok que se han utilizado en el resto de los objetos.
3. En el paquete *services*, añadiremos un paquete con el nombre “<nombre-modulo>Service” que contendrá la interfaz y la implementación o implementaciones necesarias de esa interfaz.
4. Añadiremos en el paquete *repositories* una nueva interfaz con el nombre “<nombre-modulo>Repository” que extenderá la clase *CrudRepository* parametrizada con el tipo de datos que manejará y el tipo de dato que sea su id.
5. Crearemos una tabla en la BD que represente al nuevo módulo a la que accederemos mediante el repositorio creado en el paso anterior.
6. Por último deberemos crear la interfaz gráfica de este nuevo módulo (en caso de ser accesible gráficamente y no formar solo parte de la API) creando las plantillas necesarias en la carpeta resources.

Añadiendo una nueva Calculation Tool a nuestra aplicación

Al publicar nuestra aplicación las Calculation tools de las que dispone la misma son las enumeradas en (sección). Para añadir una nueva Calculation Tool y mantener la modularidad y arquitectura de nuestra aplicación seguiremos los siguientes pasos:

1. En el paquete *Models* añadiremos la Calculation Tool deseada indicando su nombre, id de sección e id de la propia Calculation Tool.
2. En el paquete *Services/ctservice,* en la sección correspondiente añadiremos, en la implementación deseada de la interfaz, un método que reciba los parámetros de cálculo de entrada y devuelva un objeto del tipo CTResult (así como todos los métodos auxiliares necesarios) realizando en su interior todos los cálculos pertinentes.
3. En el paquete *Controllers,* en la clase *CalculationToolsController* *añadiremos*, en el método *getCalcResult* la llamada al método de cálculo creado en el paso anterior. Lo añadiremos teniendo en cuenta que el *switch* más externo indica el id de sección y el *switch* interno indica el id de la Calculation Tool dentro de esa sección.

Añadiendo un nuevo usuario administrador

A fecha de entrega de este proyecto el único usuario existente en la base de datos con el rol de administrador es (comosellame) ¿cuya contraseña queda en conocimiento de (quiensea)?. Para añadir nuevos usuarios con dicho rol a la base de datos. Deberemos acceder a esta y crear una nueva entrada en la tabla users e insertar:

* En el campo email, el identificador del usuario.
* En el campo role, el rol, que en este caso será ADMIN
* En el campo password, la contraseña encriptada según el algoritmo BCrypt. Para calcular de manera sencilla el hash a partir de una cadena de texto recomendamos el uso de <https://bcrypt-generator.com/>

Apéndices

Resultados

Conclusiones

Bibliografía

Anexos

Glosario, siglas y acrónimos

API: Application Programming Interface.

IDE: Integrated Development Environment.

JDK: Java Development Kit.

MVC: Modelo Vista Controlador.

REST: Representational State Transfer.