



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

ÁREA DE INGENIERÍA TELEMÁTICA

TRABAJO FIN DE GRADO N° 17010523

“Aplicación web para la gestión de un repositorio”

Autor:
Raúl García Fernández

Tutor:
Raquel Blanco Aguirre

Junio de 2017

Manual Del Usuario

Índice

Índice	2
Ilustraciones.....	4
Tablas	6
1. Introducción:	7
2. Obtención del código:	8
3. Instalación del repositorio:	10
3.1. Generaciones ejecutables:	10
3.1.1. Eclipse: Conversión Maven.....	10
3.1.2. Eclipse: Incluir e Ejecutar.....	13
3.2. Instalación de base de datos:.....	18
3.2.1. Generación de estado inicial:.....	19
3.3. Ejecución del repositorio:	21
4. Instalación del servicio web:	22
4.1. Instalación de Node.js:.....	22
4.2. Arranque servidor web:	22
5. Arranque rápido en Ubuntu server:	24
5.1. Instrucciones de arranque:	24
5.2. Listado de programas:.....	24
6. Manual de creación de proyectos:	25
7. Manual de utilización del repositorio:.....	26
7.1. Iteración habitual:	26
7.2. UniApi (JS):.....	26
8. Manual de utilización de la aplicación web:	28
8.1. Autenticarse y crear una cuenta:	28
8.2. Navegar por los grupos:	30

8.3.	Crear un grupo:	31
8.4.	Crear un proyecto:	33
8.5.	Crear una ejecución:	35
8.6.	Gestionar un proyecto propio:	36
8.7.	Gestionar un grupo propio:	38
8.8.	Información básica:.....	39
8.9.	Miembros:.....	40
8.9.1.	Subgrupos:.....	40
8.9.2.	Permisos:	40
8.9.3.	Gestionar una ejecución propia:	40

Ilustraciones

Ilustración 2-1 Página del repositorio.....	8
Ilustración 2-2. Botón de descarga GIT.....	9
Ilustración 3-1. Seleccin import eclipse	11
Ilustración 3-2. Multiples opciones de import.....	11
Ilustración 3-3. Import projects final	12
Ilustración 3-4. Convert project maven	13
Ilustración 3-5. Maven run as.....	14
Ilustración 3-6. Opciones maven	14
Ilustración 3-7. Carpeta target	15
Ilustración 3-8. Estado final externalJar del proyecto REST	16
Ilustración 3-9. Comprobación JDK.....	17
Ilustración 3-10. Comprobación JavaSE	17
Ilustración 3-11. Ventana neo4j.....	18
Ilustración 3-12. Ventana neo4j Iniciada.....	19
Ilustración 4-1. Jerarquía de uniapi (prompt)	23
Ilustración 4-2. Ejecución del servidor web	23
Ilustración 5-1 Ventana Login.....	28
Ilustración 5-2 Página Principal	29
Ilustración 5-3 Botón sign up	29
Ilustración 5-4 Ventana nuevo usuario	30
Ilustración 5-5 Menú Navegación	30
Ilustración 5-6 Nuevo grupo.....	32
Ilustración 5-7 Nuevo grupo: Tipo grupo.....	32
Ilustración 5-8 Nuevo grupo: Permisos	33
Ilustración 5-9 Nuevo proyecto: Integración o creación	33
Ilustración 5-10 Nuevo proyecto: principal.....	34
Ilustración 5-11 Nuevo grupo: tipo de proyecto.....	35
Ilustración 5-12 Nuevo grupo: Entradas por defecto	35
Ilustración 5-13 Nueva ejecución	36
Ilustración 5-14 Proyectos	37
Ilustración 5-15 Gestión proyecto	37

Ilustración 5-16 Mis Grupos.....	38
Ilustración 5-17 Gestión grupo	39
Ilustración 5-18 Gestión de ejecución: En ejecución	41
Ilustración 5-19 Gestión de ejecución: Finalizado	42

Tablas

Tabla 4-10 UniApi.js: Nombre de funciones	27
---	----

1. Introducción:

Este documento denominado “Manual del usuario”, es un documento de un conjunto de documentos cuya finalidad es el definir y diseñar una “Aplicación web que gestione un repositorio”.

Para poder describir mejor el problema se deberá desarrollar el significado de lo que se va a definir, es decir, “Aplicación web que gestione un repositorio”. Un repositorio es según la Wikipedia: Un repositorio de software es un lugar de almacenamiento del cual pueden ser recuperados e instalados los paquetes de software en un ordenador.

Nuestro repositorio mantendrá un lugar de almacenamiento dentro del ordenador que lo instale, con el fin de contener diversos proyectos que albergarán programas de diversos lenguajes que podrán ser ejecutados. Nuestro repositorio devolverá no es programa almacenado, sino, la información de interactuar con él.

Nuestra aplicación web, gestionara la comunicación con nuestro repositorio. Pidiéndole información y mostrándosela al usuario de una manera sencilla para su entendimiento. Se insta que si se quiere saber más sobre el objetivo de la aplicación se lea el documento general de la memoria.

El objetivo de este documento es el realizar un manual del usuario, para que todos los lectores e revisores de este artículo puedan utilizar el software a desarrollar. Para ello se listará un conjunto de instrucciones y descripciones del sistema que creemos que serán de utilidad para poder utilizar el sistema.

2. Obtención del código:

Para poder obtener los códigos de los diversos programas usaremos la plataforma de repositorios GIT conocida como **Github**. Github provee un repositorio permanente para nuestro programa con un seguidor de modificaciones muy notorio que facilitara a cualquier persona el poder descargar versiones modificadas o mejoradas.

<https://github.com/RaulGF92/UniApi>

La dirección previamente mostrada se referencia al espacio donde el repositorio se alberga, ahí se podrá ver todas las modificaciones, paso por paso que el desarrollador a usado. Cualquier persona con conocimientos básicos sobre GIT puede descargárselo, no obstante, mostraremos las dos formas de realizarlo:

- **Descargando a través de la página web:** En este caso el usuario deberá introducirse en la página indicada por el documento. Se le mostrara una página web como la indicada en la figura de a continuación. Ahí deberá pulsar en el botón “Clone or download”.

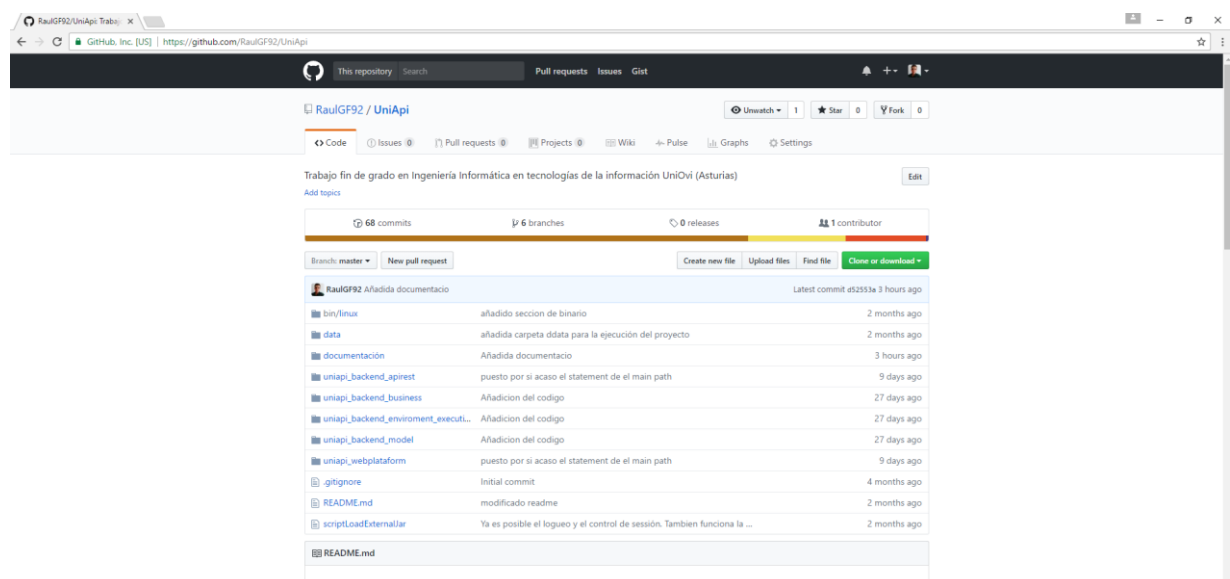


Ilustración 2-1 Página del repositorio

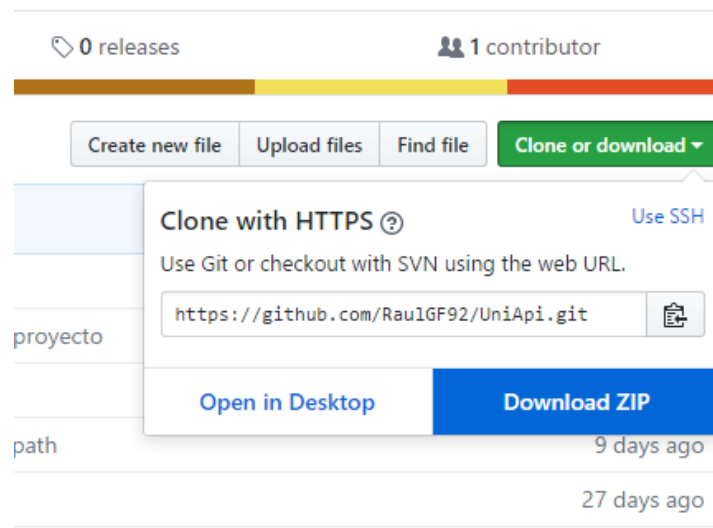


Ilustración 2-2. Botón de descarga GIT

Tras pulsar el botón de download Zip. Se descargará un archivo Zip donde se contendrá el proyecto UniApi. Es necesario descomprimirlo y depositarlo en el sistema de archivos del servidor que albergará el servicio.

- **Mediante consola de comandos GIT:** Para ello debemos conocer la dinámica del protocolo GIT y sus comportamientos. Primero realizaremos un clonado del repositorio usando la URL que se nos indica posteriormente. Tras eso el protocolo se ocupará automáticamente de la descarga.

3. Instalación del repositorio:

Para instalar el repositorio necesitaremos los datos del repositorio, además de un IDE denominado Eclipse. Usaremos este IDE por que gestiona Maven un gestor de paquetes Java de forma transparente para el desarrollador, aunque es posible el uso de maven usando el prompt del sistema operativo.

3.1. Generaciones ejecutables:

En esta parte realizaremos un ejecutable que se utilizará para la ejecución del repositorio a partir de todos los datos de la jerarquía UniApi. Hay ejecutables de prueba en la carpeta **/bin/Linux** a modo de ejemplo.

3.1.1. Eclipse: Conversión Maven

Para empezar con la conversión habrá que inicializar el eclipse. Se nos preguntará donde albergar el workspace y es recomendable indicarle al eclipse que debe de ser en la jerarquía UniApi. El eclipse se empezará a cargar y proporcionará un espacio de trabajo vacío.

Después se entrará en las opciones File y se elegirá la opción importar. Ahí se desplegará un sinfín de opciones y se elegirá la opción “**Existing projects in the workspace**”.

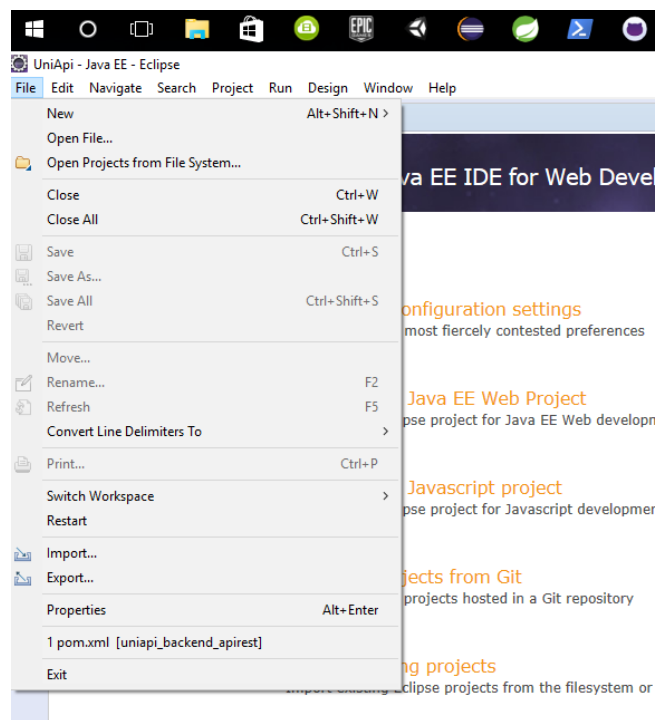


Ilustración 3-1. Selección import eclipse

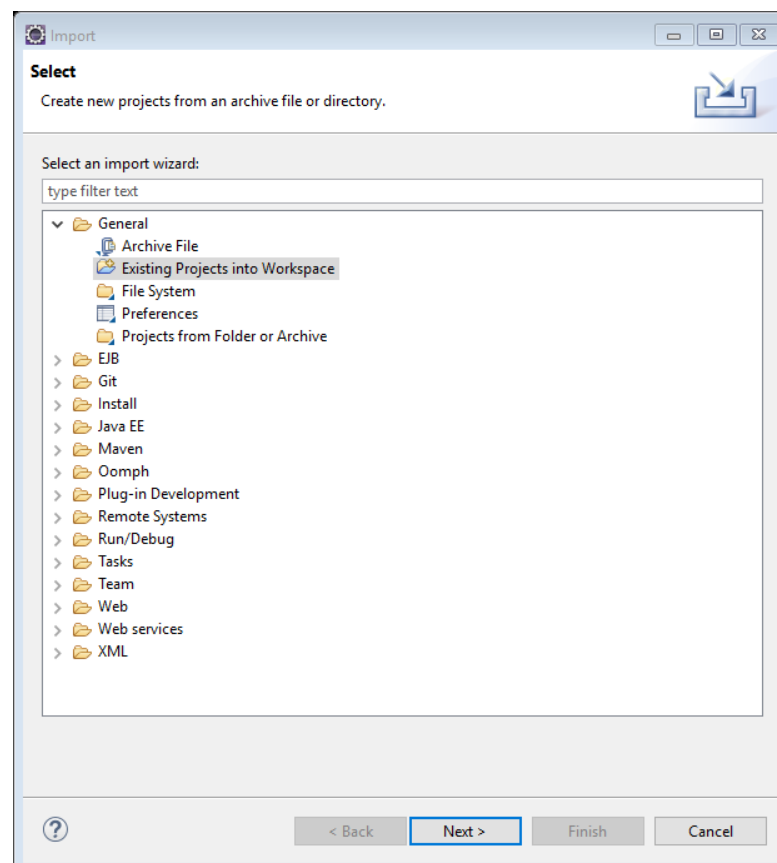


Ilustración 3-2. Múltiples opciones de import

Tras seleccionar la opción se nos mostrará otra ventana donde deberemos seleccionar el check box “**root directive**”. Habrá que pulsar el botón “Browse”, tras eso, se desplegará una ventana que nos ayudará a navegar por la jerarquía de archivos y se tendrá que seleccionar la carpeta uniapi. Tras la selección, la ventana nos indicará la posibilidad de importar los tres proyectos de uniapi, habrá que seleccionar los tres y por ultimo darle a finish.

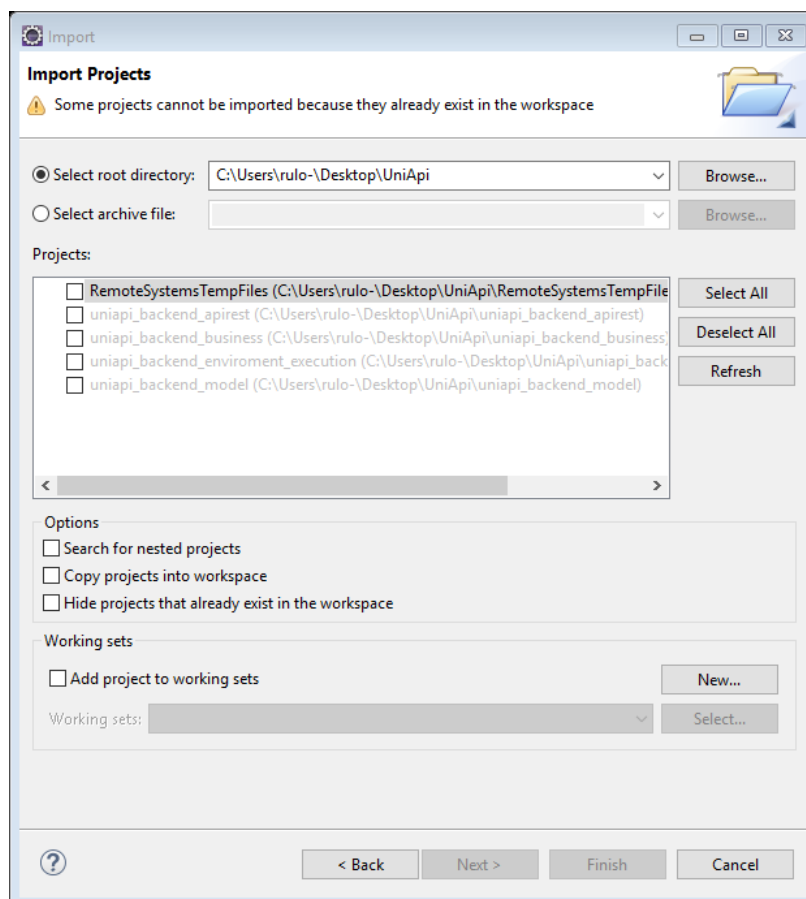


Ilustración 3-3. Import projects final

Tras la importación tendremos cuatro proyectos java. Para terminar la conversión deberemos ir uno por uno pulsando el botón derecho. Accediendo a la pestaña “configure” y por ultimo pulsando “convert in maven project”.

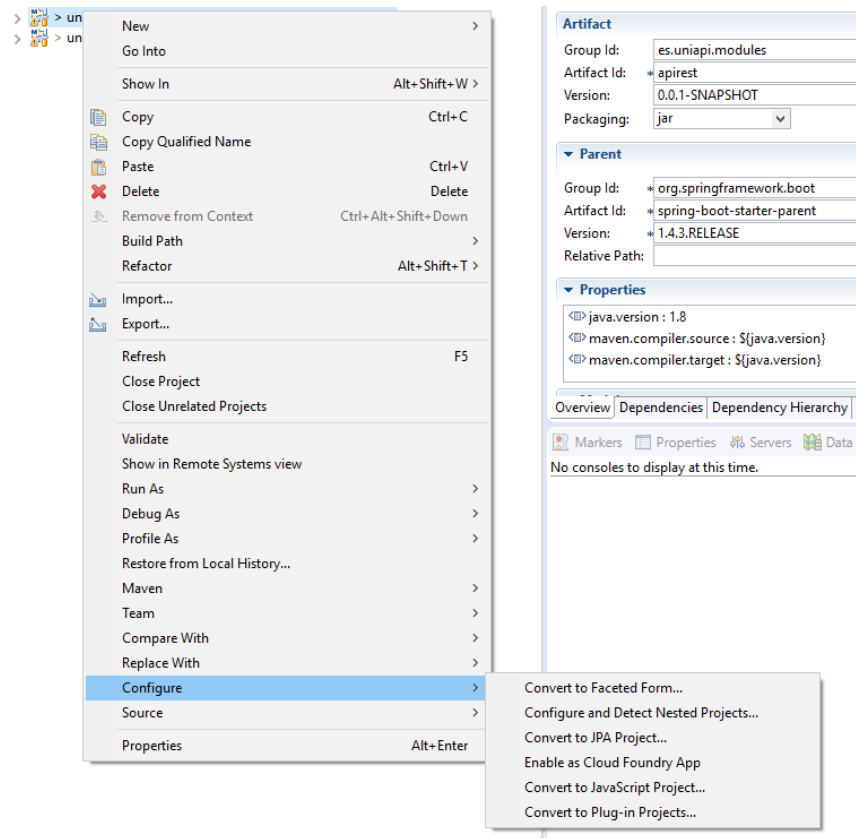


Ilustración 3-4. Convert project maven

Con esto todos los proyectos del workspace serán maven. Por último, habrá que seleccionar cada proyecto de nuevo. Pulsa botón derecho, deberás clicar la opción maven y después clicar update. Esto actualizará los proyectos. Esta acción generará muchos errores, pero luego serán arreglados.

3.1.2. Eclipse: Incluir e Ejecutar

Tras contener todos los proyectos maven instalados e actualizados en el workspace vamos a compilarlos y ejecutarlos. La aplicación del repositorio es una aplicación en pila, es decir, las capas superiores utilizan datos de la capa inferior. Es necesario compilar los datos para que las capas puedan utilizarlos. Para realizar esta operación se irá de la capa más inferior a la más superior construyendo sus proyectos, compilándolos y entregándolos a las capas superiores.

1. **Modelo de datos:** nos colocamos en el proyecto “**uniapi_backend_model**” y lo seleccionamos, pulsamos el botón derecho y nos dirigimos a run, donde pincharemos maven build...

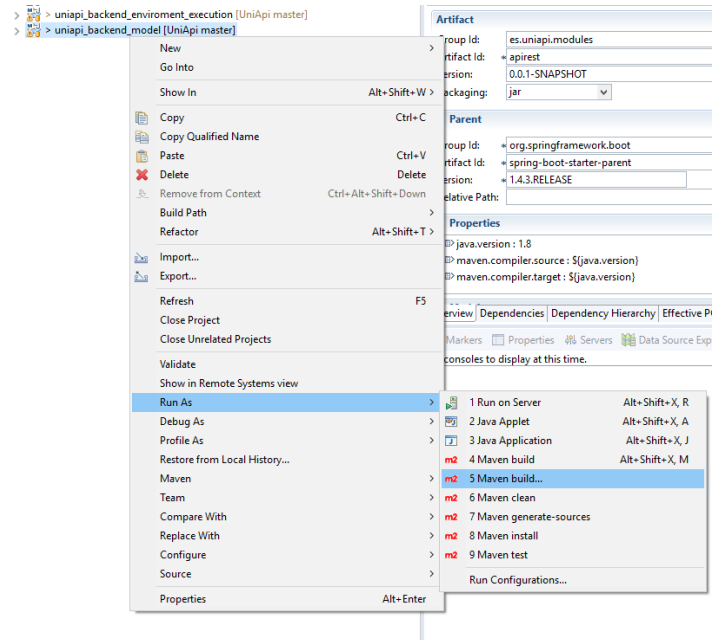


Ilustración 3-5. Maven run as...

Tras pinchar en la opción nos aparecerá una ventana de opción de maven en la pestaña “Goals” se introducirá el siguiente código:

clean build package

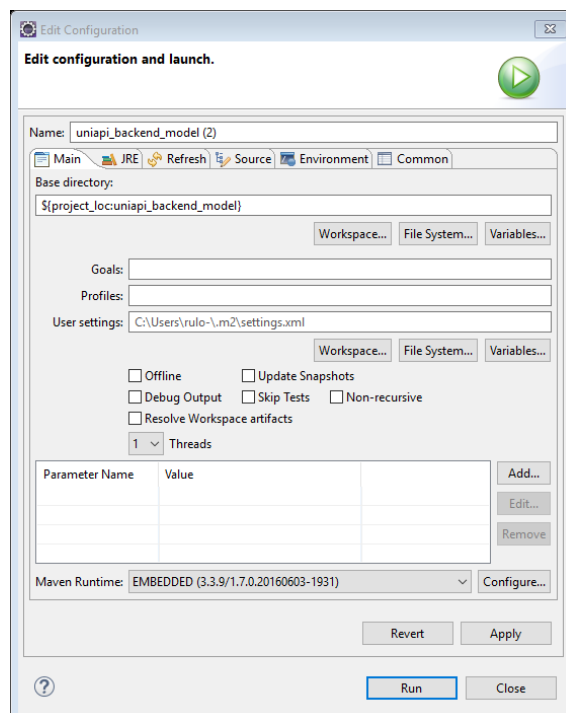


Ilustración 3-6. Opciones maven

Tras realizar esto tendremos un ejecutable dentro de la carpeta target del proyecto justo tal que en la figura siguiente:

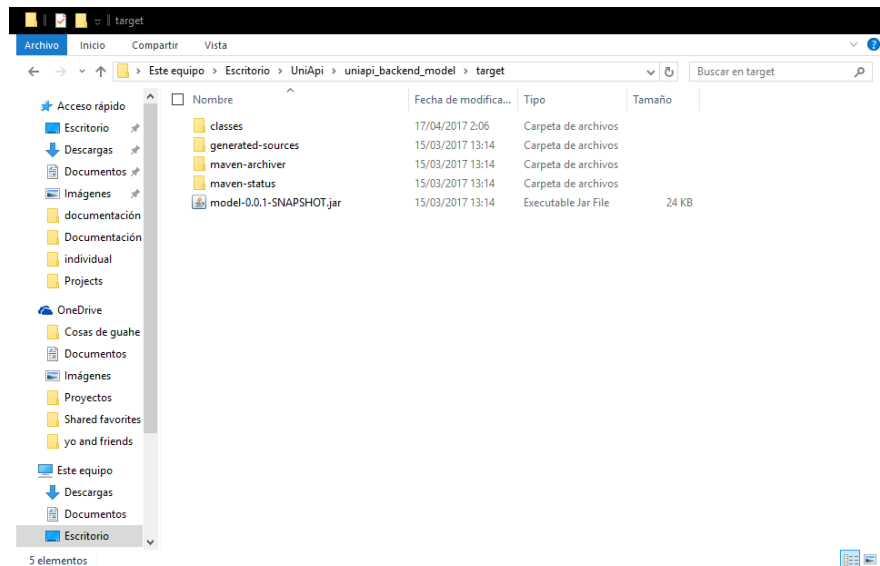


Ilustración 3-7. Carpeta target

En todos los proyectos menos en “**uniapi_backend_model**” se creará una carpeta denominada “externalJar”, si no el proceso no funcionará. Tras la creación de la carpeta se introducirá el “**model-0.0.1-SNAPSHOT.jar**” en todas las “externalJar” de los proyectos “**uniapi_backend_enviroment_execution**”, “**uniapi_backend_business**” y “**uniapi_backend_apirest**”.

2. Capa de ejecución: Para esta capa se realizará el mismo procedimiento que en el proyecto “**uniapi_backend_enviroment_execution**”. Se utilizará maven con la instrucción “clean install package” consiguiendo un ejecutable denominado “**enviroment_execution-0.0.1-SNAPSHOT.jar**”. El ejecutable deberá ser trasladado a las capas superiores (“**uniapi_backend_business**”, “**uniapi_backend_apirest**”) a sus carpetas externalJar.

3. Capa de negocio: Realizaremos las mismas instrucciones en el proyecto “**uniapi_backend_business**” que en los casos anteriores y conseguiremos un archivo denominado “**business-0.0.1-SNAPSHOT.jar**” que estará dentro la carpeta target.

Después el archivo .jar será necesario trasladarlo al externalJar del proyecto “uniapi_backend_apirest”.

4. **Capa REST:** Esta capa es la última. Para que todo funcione correctamente deberá albergar todos los ejecutables como sigue en la siguiente figura:

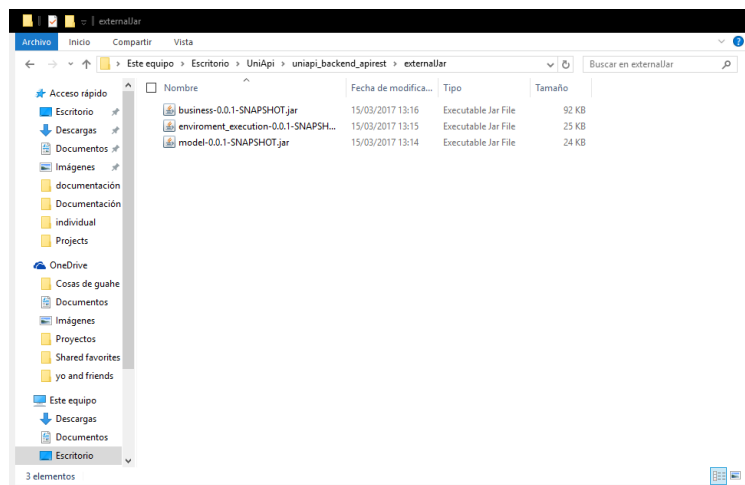


Ilustración 3-8. Estado final externalJar del proyecto REST

No es necesario aplicar la compilación a esta capa, ya que no se necesita su ejecutable como dependencia. No obstante, puede hacerse para asegurarse que no hay errores si la compilación es correcta. Entonces existirá en la carpeta target de proyecto, un archivo denominado “apirest-0.0.1-SNAPSHOT.jar”.

Posibles errores:

1. **Fallo en las dependencias:** Asegúrate que los archivos que pasas a las carpetas “externalJar” contienen los mismos nombres que los descritos en este documento.
2. **JRE Than JDK:** Este fallo se da en los proyectos maven que no tienen un JRE igual que el del compilador que enlaza eclipse. Para eso deberás entrar en las propiedades del proyecto (botón derecho en el proyecto). Buscar la pestaña Java y la subpestaña Java Compiler. También deberías observar el Java Buid Path, y comprobar que es la librería JavaSE es el mismo que el JDK.

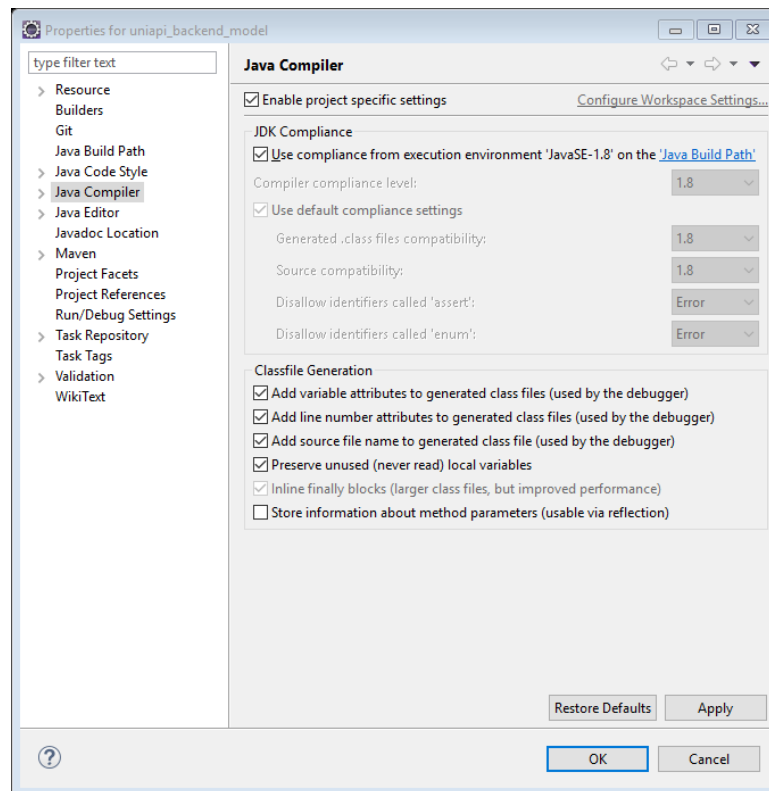


Ilustración 3-9. Comprobación JDK

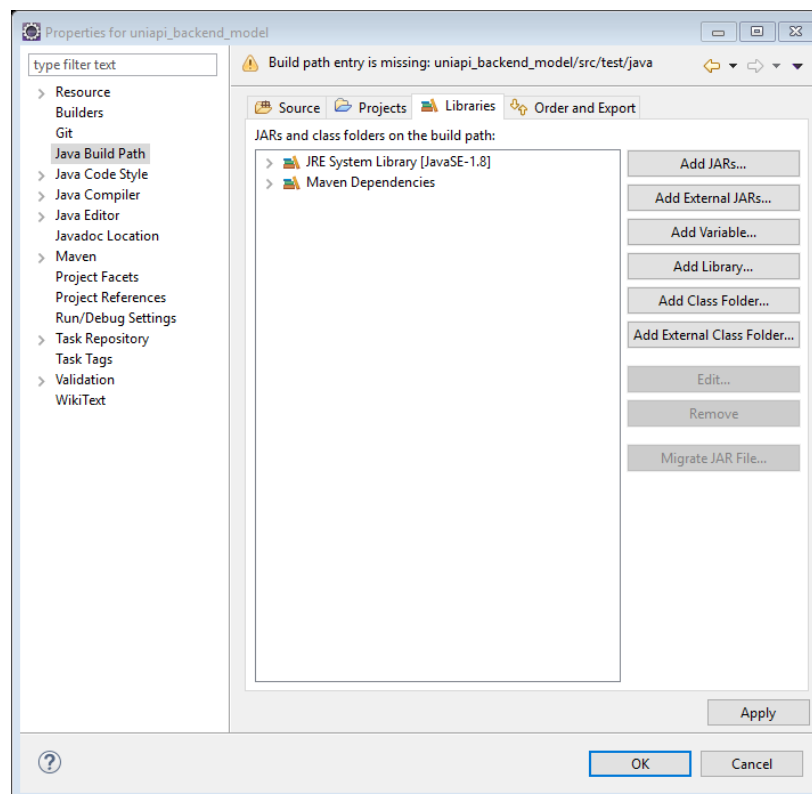


Ilustración 3-10. Comprobación JavaSE

3.2. Instalación de base de datos:

Para realizar este repositorio se ha utilizado como base de persistencia la base de datos Neo4j. Para la realización de la instalación de Neo4j se describirán las dos instalaciones en función de su núcleo Windows o Linux:

- **Windows:** Para instalar la base de datos en Windows se deberá acceder a la siguiente dirección URL.

<https://neo4j.com/download/community-edition/>

en la URL se nos descargará un .exe. y se deberá instalar el ejecutable, tras el final de la instalación se mostrará un programa denominado neo4j Community edition.

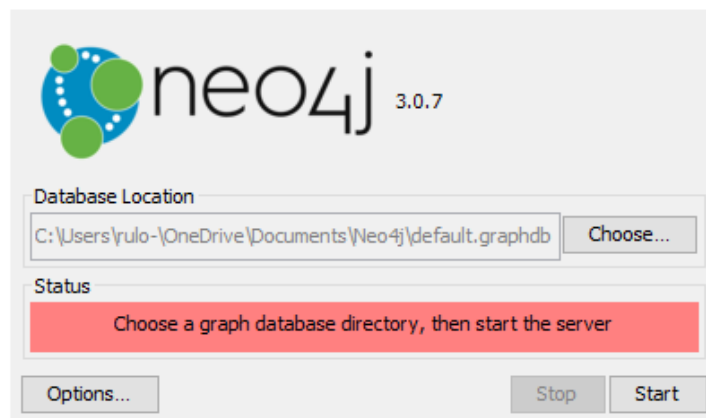


Ilustración 3-11. Ventana neo4j

En este programa se podrá instalar la base de datos en el lugar que queramos del sistema de archivos, no obstante, la carpeta uniapi contiene una subcarpeta denominada data. Esta subcarpeta tiene como objetivo albergar toda la información referente a la aplicación, así que sería recomendable albergar la base de datos ahí. Cuando obtengamos la dirección que queramos, solo se pulsará en el botón “Start” y se visualizará en el cuadro “Status” una dirección donde se podrá acceder a través del navegador web para realizar gestiones por consola.

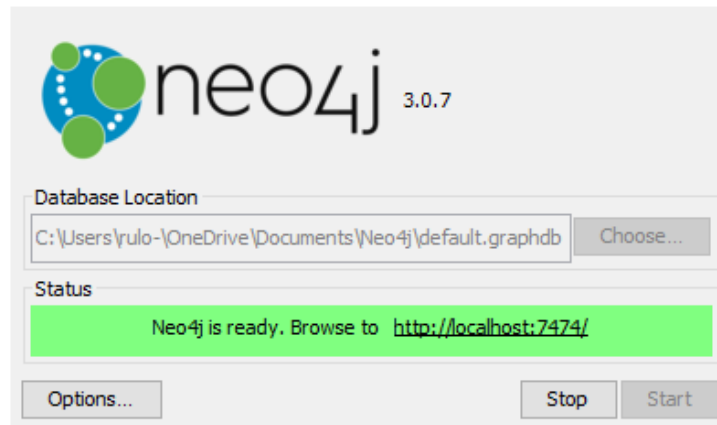


Ilustración 3-12. Ventana neo4j Iniciada

- **Linux:** Para instalar la base de datos en Linux iremos a la URL listada a continuación y se descargará la opción de Linux.

<https://neo4j.com/download/community-edition/>

Tras la descarga se obtendrá un archivo comprimido donde se encontrará una carpeta denominada “neo4j-versionDeSoftware” (la recomendación del desarrollador es que se descomprima la carpeta y se albergue en la carpeta uniapi). Después para iniciar la base de datos, será necesario introducirse en la carpeta descomprimida y buscar la carpeta “bin” y por ultimo ejecutar el siguiente comando.

```
./neo4j-community-3.1.1/bin/neo4j console
```

3.2.1. Generación de estado inicial:

Para poder realizar una correcta ejecución del repositorio será necesario que la base de datos este en un estado específico. Es necesario que el repositorio contenga unas entidades mínimas para garantizar un correcto funcionamiento del mismo y actualmente el desarrollador no tiene automatizado la generación del caso inicial.

Para empezar, se necesitará verificar si en la carpeta uniapi existe la carpeta **data/**, la cual deberá contener tres hijos **conf/**, **projects/**, **tmpExecution/** y si no es así, se requiere la creación de estas carpetas.

Después de eso, se inicializará la base de datos siguiendo las instrucciones del punto anterior y se arrancará el navegador web introduciendo la siguiente dirección URL en el navegador:

<http://localhost:7474/>

La URL dará acceso en la base de datos y se solicitará que se modifique la contraseña, tras realizar la modificación se deberá ir al archivo de la carpeta **data/conf/uniapiConfig.properties** ahí se cambiará las credenciales por las creadas en la página web.

Después de configurar la base de datos se creará dos entidades necesarias para el funcionamiento de la aplicación: La entidad administradora original y la entidad del grupo público.

1. `CREATE (a:UserLogin {user:"admin@admin.com",pass:admin,creationTime:0,rol:ADMIN,hashcode:""})`
2. `CREATE(g:Group{name:"public",creationDate:0,type:"MAIN_GROUP",hashcode:"publicHASHCODE",sharingGroupPermissions:"[YES,YES,YES]",projectPropertiesPermissions:"[YES,YES,YES]",memberGestionPermissions:"[YES,YES,YES]",groupCreationPermissions:"[YES,YES,YES]",description:"PUBLIC GROUP"})`
3. `MATCH (u:UserLogin),(g:Group) WHERE u.user="admin@admin.com" AND g.hashcode="publicHASHCODE" CREATE (u)-[k:KNOWS {dateFrom:0}]->(g);`

Después de esto, se podrá inicializar el repositorio sin fallo alguno.

3.3. Ejecución del repositorio:

Si se ha realizado correctamente los pasos previos. Se tendrá todos los proyectos compilados e inyectados en los diversos proyectos hasta llegar al proyecto final. Aquí tenemos tres opciones para inicializar la ejecución del repositorio:

- **Ejecución mediante eclipse:** Se seleccionará el proyecto de “**uniapi_backend_apirest**” y se aplicará la misma metodología que el punto 3.1 introduciendo los siguientes comandos para generar la compilación y la ejecución de programa:

```
clean build package  
spring-boot:run
```

- **Ejecución mediante consola y Maven:** Utilizando la consola se deberá ir por la jerarquía del sistema hasta la carpeta uniapi. Luego se deberá de entrar en el proyecto “**uniapi_backend_apirest**” y después se deberán introducir los siguientes comandos para generar la compilación y ejecución del programa:

```
maven clean build package  
maven spring-boot:run
```

- **Ejecución mediante .jar:** Utilizando el comando “maven clean build package” utilizado en las otras dos opciones anteriores, se compila un .jar en la carpeta target del proyecto “**uniapi_backend_apirest**” y puede ser ejecutado utilizando el siguiente comando:

```
java -jar apirest-0.0.1-SNAPSHOT.jar
```

4. Instalación del servicio web:

4.1. Instalación de Node.js:

Para poder iniciar el servicio web del proyecto necesitaremos instalar el motor Node.js de JavaScript. Este motor puede ser instalado desde su página web principal donde puede ser instalado para diversos sistemas operativos. A continuación, se proporciona unas instrucciones básicas para su instalación en diversos sistemas operativos:

- **Windows:** Debe instalarse siguiendo esta dirección web. En él se descargará un instalador, hay que seguir las instrucciones e instalarlo. Para poder manejarlo Node.js suele venir con un acceso directo a un prompt que viene desplegado la variable de entorno por defecto.

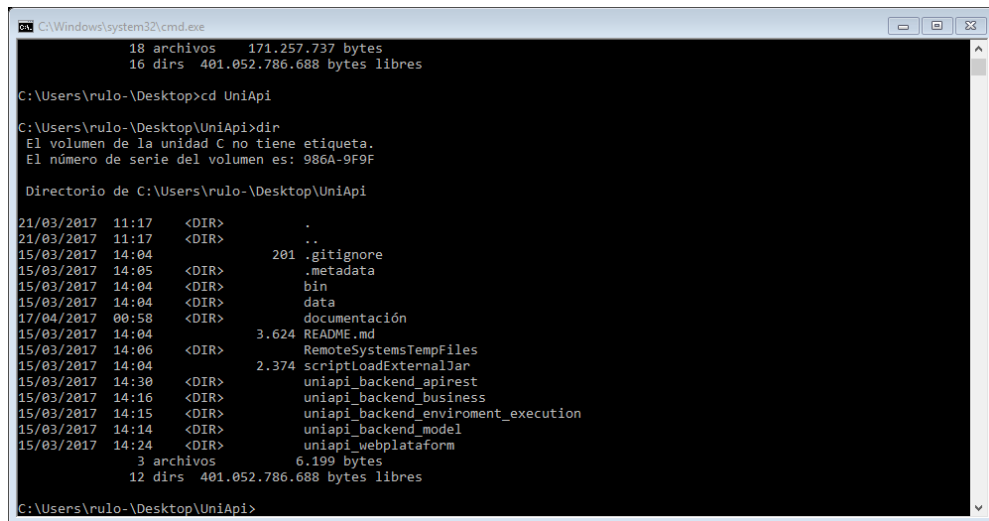
<https://nodejs.org/es/>

- **Linux:** Para Linux la cosa es más sencilla solo deberán ejecutar la siguiente instrucción y se instalara automáticamente. (la instrucción descrita es para Ubuntu y si se quiere para demás sistemas operativos se debe usar el gestor de paquetes por defecto del mismo)

```
sudo apt-get install node
```

4.2. Arranque servidor web:

Para poder realizar el arranque del servidor web habrá que introducirse en el prompt o consola del sistema operativo que utilizemos y acceder a la jerarquía de la carpeta de UniApi.



```
C:\Windows\system32\cmd.exe
18 archivos    171.257.737 bytes
16 dirs    401.052.786.688 bytes libres

C:\Users\rulo-\Desktop>cd UniApi

C:\Users\rulo-\Desktop\UniApi>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 986A-9F9F

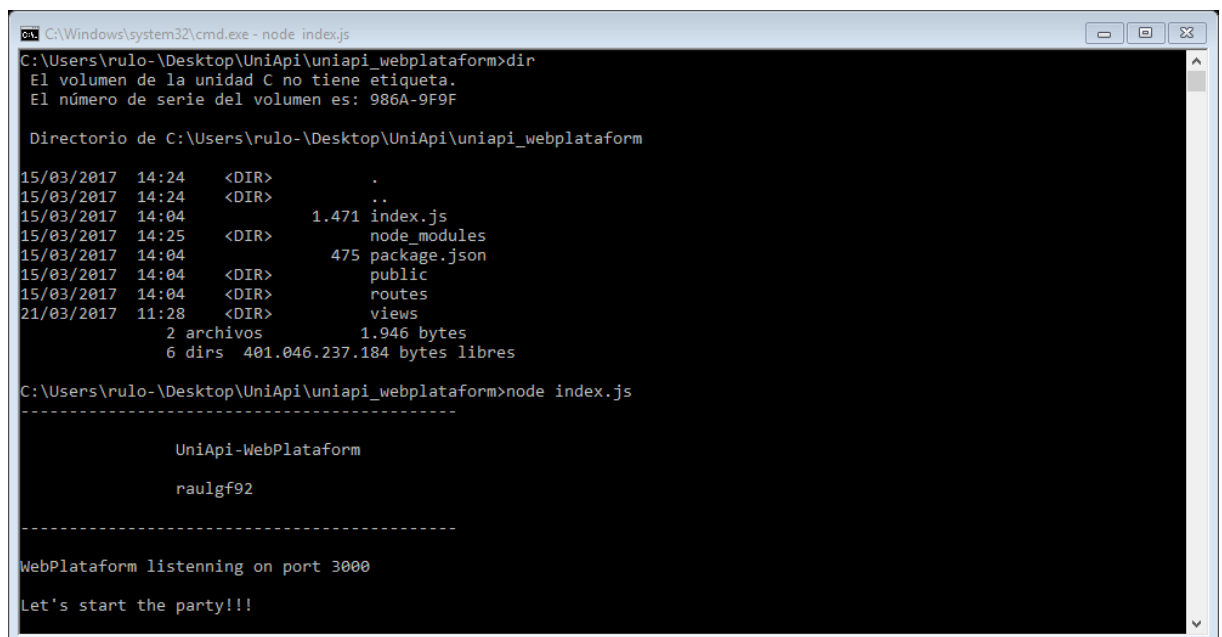
Directorio de C:\Users\rulo-\Desktop\UniApi

21/03/2017  11:17    <DIR>        .
21/03/2017  11:17    <DIR>        ..
15/03/2017  14:04                201 .gitignore
15/03/2017  14:05    <DIR>        .metadata
15/03/2017  14:04    <DIR>        bin
15/03/2017  14:04    <DIR>        data
17/04/2017   00:58    <DIR>        documentación
15/03/2017  14:04                3.624 README.md
15/03/2017  14:06    <DIR>        RemoteSystemsTempFiles
15/03/2017  14:04                2.374 scriptLoadExternalJar
15/03/2017  14:30    <DIR>        uniapi_backend_apirest
15/03/2017  14:16    <DIR>        uniapi_backend_business
15/03/2017  14:15    <DIR>        uniapi_backend_enviroment_execution
15/03/2017  14:14    <DIR>        uniapi_backend_model
15/03/2017  14:24    <DIR>        uniapi_webplataform
                        3 archivos    6.199 bytes
                        12 dirs    401.052.786.688 bytes libres

C:\Users\rulo-\Desktop\UniApi>
```

Ilustración 4-1. Jerarquía de uniapi (prompt)

Tras acceder a la jerarquía de UniApi deberemos introducirnos en la carpeta “uniapi_webplataform” y tras eso ordenar a Node.js que ejecute el programa.



```
C:\Windows\system32\cmd.exe - node index.js

C:\Users\rulo-\Desktop\UniApi\uniapi_webplataform>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 986A-9F9F

Directorio de C:\Users\rulo-\Desktop\UniApi\uniapi_webplataform

15/03/2017  14:24    <DIR>        .
15/03/2017  14:24    <DIR>        ..
15/03/2017  14:04                1.471 index.js
15/03/2017  14:25    <DIR>        node_modules
15/03/2017  14:04                475 package.json
15/03/2017  14:04    <DIR>        public
15/03/2017  14:04    <DIR>        routes
21/03/2017  11:28    <DIR>        views
                        2 archivos    1.946 bytes
                        6 dirs    401.046.237.184 bytes libres

C:\Users\rulo-\Desktop\UniApi\uniapi_webplataform>node index.js
-----
UniApi-WebPlataform
raulgf92
-----
WebPlataform listening on port 3000
Let's start the party!!!
```

Ilustración 4-2. Ejecución del servidor web

5. Arranque rápido en Ubuntu server:

Para facilitar el desarrollo, el desarrollador ha producido un arranque rápido del sistema. Este arranque automatiza todas las tareas que están descritas en este documento. Cabe destacar que para que el arranque funcione, el sistema deberá estar en un estado específico descrito a continuación.

5.1. Instrucciones de arranque:

Para arrancar de manera rápida deberemos ubicarnos en la carpeta de la jerarquía principal (carpeta uniapi) y mediante el Bash de Ubuntu (conocido como terminal) se deberá realizar la ejecución de las siguientes instrucciones:

```
cd bin/Linux  
./uniapi
```

5.2. Listado de programas:

Para que funcione el arranque será necesario que los siguientes programas sean instalados correctamente e incluso así, no se cree que funcione correctamente.

- Maven
- Node.js
- Neo4j (ubicado en la carpeta uniapi)

6. Manual de creación de proyectos:

Para poder ejecutar un proyecto en UniApi son necesarios ciertos requisitos para la realización de los proyectos. Hay que decir que actualmente el desarrollador no ha tenido en cuenta el análisis y monitorización del diseño de los proyectos. Es por eso que será parte del usuario el saber realizarlo.

- **Main:** Todos los programas tienen que tener un main. Un Main es un archivo donde se inicializa todos los proyectos y es necesario que este archivo esté en la carpeta superior del proyecto. En el caso que no se pueda introducir el main en la carpeta superior, se concatenará por delante del nombre un path relativo desde la carpeta superior del proyecto.
- **Argumentos:** Los argumentos que se entregarán al programa Main serán por consola, es decir, qué es obligatorio controlar en el proyecto los argumentos de las variables de entorno y es recomendable que se trate las variables de entorno por parte del usuario, ya que el repositorio no lo hace. Hay que tener en cuenta que los argumentos siempre serán +1, eso quiere decir que al final de los argumentos se añadirá un argumento final, el cual es el path donde se espera que se devuelva la respuesta.
- **Respuesta:** Es obligatorio para el proyecto que la respuesta se de en el path que se explicó en el punto anterior y tendrá que ser del conocimiento del usuario el saber escribir un fichero en el lenguaje que maneja. El contenido de la respuesta debe estar escrito obligatorio en el formato JSON. Existen muchas librerías que facilitan esto y es obligación del usuario conocerlas además de saber manejarlas.

7. Manual de utilización del repositorio:

El repositorio utiliza como método de comunicación el protocolo HTTP. Este protocolo contiene unas directrices especiales que hay que conocer. No obstante si se quiere trabajar fácilmente con él, cualquier navegador, librerías de Python o programa de ordenador facilitan el uso de peticiones HTTP.

7.1. Iteración habitual:

Antes de empezar a utilizar el repositorio deberá haber un proceso de autenticación usando la parte del protocolo de comunicación 6.2.1. Si existe autenticación correcta se le otorgará una ID de acceso (token de acceso). Después de eso, cualquier acción que se ordene al sistema se deberá introducir el ID o token. Todas las acciones son sin estado, es decir, terminado la acción el sistema olvidará la acción realizada.

7.2. UniApi (JS):

Para facilitar el desarrollo de aplicaciones web utilizando el repositorio, se ha diseñado una librería para angular.js extrapolable a cualquier tecnología que use JavaScript (utilizando JQuery). Esta API está hecha mediante funciones que utilizan el protocolo explicado en el punto 6.2. La API facilita a los desarrolladores de aplicaciones web que no quieren entender el protocolo de comunicación del repositorio el usarlo.

Para poder usarlo. Debemos instalarlo en la página web incluyéndolo con el siguiente trozo de código HTML. Cabe destacar que deberá poder ser accedido por el servidor web que sustenta la aplicación web. Las funciones son:

Tabla 7-1 UniApi.js: Nombre de funciones

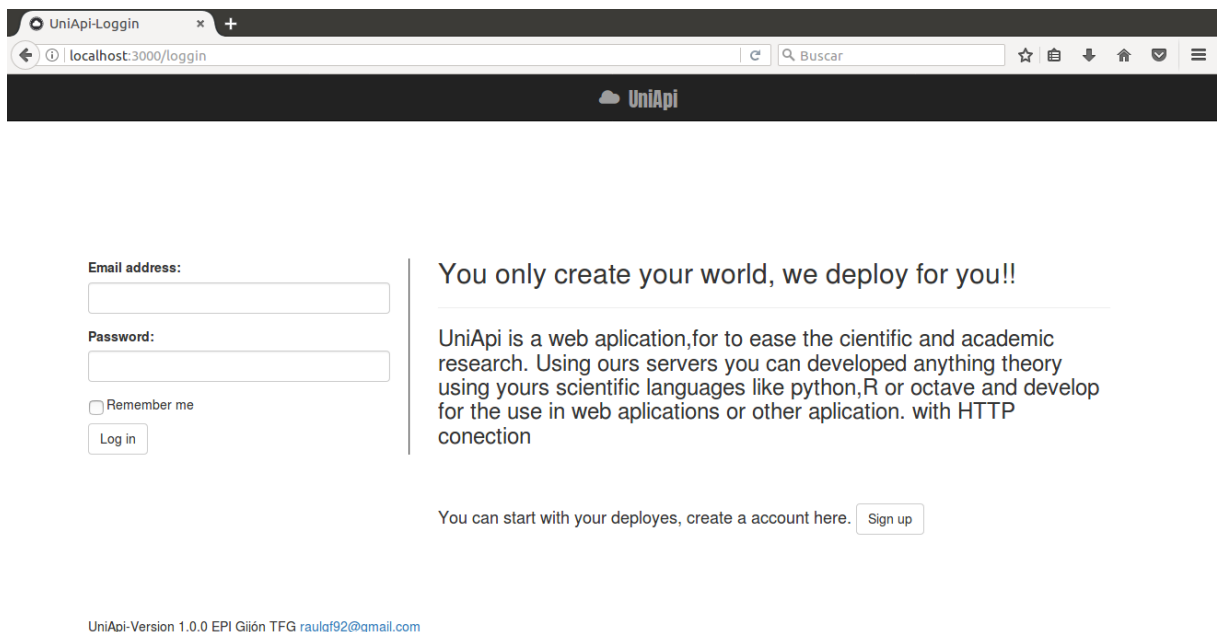
Nombre de las funciones		
Whoami()	getMyProjects()	deleteSubgroup()
changePass()	getMyGroups()	getMainPathGroups()
changeBio()	getProject()	getPathGroups()
createAccount()	updateProject()	projectsInsideGroup()
getAllUsers()	deleteProject()	getProjectsInsideGroup()
getAllProjects()	createProject()	getExecutionsRunning()
getAllGroups()	createGroup()	getAllExecutionsRunning()
getGroup()	updateGroup()	getExecution()
deleteGroup()	inviteToGroup()	getProjectOfExecution()
getAllGroupMembers()	removeMemberOfTheGroup()	getExecutionFinish()
getSubgroups()	getPathProjects()	getAllExecutionsFinish()
createSubgroup()	putGroupProject()	executedProject()
getMainPathProjects()	deleteProjectInsideGroup()	

8. Manual de utilización de la aplicación web:

La aplicación web utilizará para comunicarse con el usuario una serie de formularios y listados que facilitarán al usuario la interacción y gestión con el repositorio. A continuación, se desplegarán unos tutoriales para realizar diferentes tareas.

8.1. Autenticarse y crear una cuenta:

Cuando se acceda a la aplicación web se nos direccionará a un espacio donde tendremos que autenticarnos. En este espacio se nos dispondrá de un formulario para meter el email del usuario y la contraseña.



UniApi-Login

localhost:3000/login

UniApi

Email address:

Password:

☐ Remember me

Log in

You only create your world, we deploy for you!!

UniApi is a web application,for to ease the cientific and academic research. Using ours servers you can developed anything theory using yours scientific languages like python,R or octave and develop for the use in web applications or other aplicacion. with HTTP conection

You can start with your deploys, create a account here. Sign up

UniApi-Version 1.0.0 EPI Gijón TFG raulgf92@gmail.com

Ilustración 8-1 Ventana Login

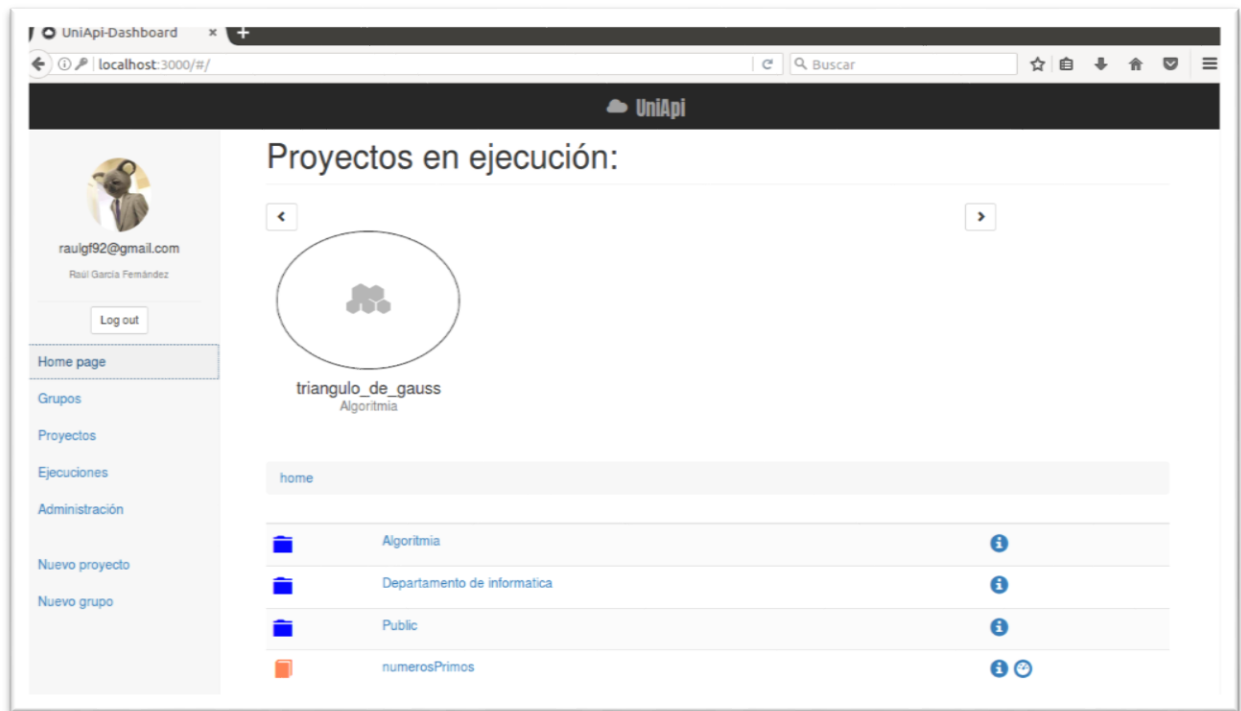


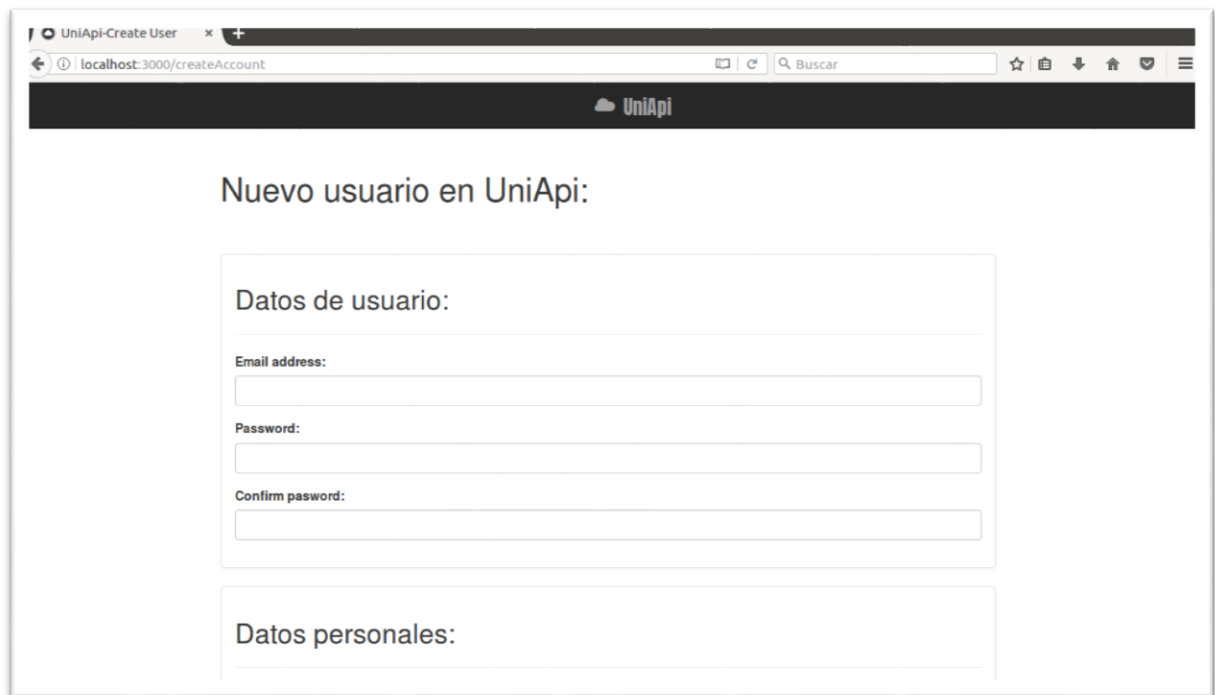
Ilustración 8-2 Página Principal

Si la autenticación ha sido correcta se le pasará la página principal de la aplicación. En caso de que la contraseña sea incorrecta se desplegará un cartel informativo declarando que la identidad es incorrecta y en caso de que no se disponga de una identidad se podrá pulsar el botón “Sign up”.

, create a account here. [Sign up](#)

Ilustración 8-3 Botón sign up

Si pulsamos el botón “Sign up”, automáticamente se nos redirigirá a una página con un formulario para poder crear una nueva cuenta.



The screenshot shows a web browser window titled 'UniApi-Create User' with the address bar showing 'localhost:3000/createAccount'. The page has a dark header with the 'UniApi' logo. The main content area is titled 'Nuevo usuario en UniApi:'. Below this title, there are two main sections: 'Datos de usuario:' and 'Datos personales:'. The 'Datos de usuario:' section contains three input fields: 'Email address:', 'Password:', and 'Confirm password:'. The 'Datos personales:' section is currently empty.

Ilustración 8-4 Ventana nuevo usuario

Si no hay fallos de validación en los datos del formulario “Nuevo usuario en UniApi”, se le devolverá al usuario a la ventana de login donde se podrá introducir el nuevo nombre de usuario y la nueva contraseña.

8.2. Navegar por los grupos:

Cuando se acceda a la página principal de la aplicación, se nos mostrara un menú lateral con todas las opciones de la aplicación. Dentro de este menú se tendrá que identificar los dos elementos del final del menú “Nuevo proyecto” y “Nuevo grupo”, ya que más tarde los usaremos en este tutorial. En el centro de la página principal se despliega un menú de navegación con todos los grupos y proyectos que podemos visualizar, algo como se muestra en la figura siguiente.



Ilustración 8-5 Menú Navegación

En el menú de la navegación tenemos 3 elementos principales:

- **Menú superior:** El menú de navegación superior, es un menú que ira registrando la navegación por la jerarquía de los grupos. Si pulsamos en los nombres de los grupos nos redirigirá a ellos.
- **Grupos:** Son los elementos de la carpeta azul, dentro del cuadro que delimita el elemento existe un elemento con la letra i, la cual, nos llevará a la información del grupo y si pulsamos en el nombre del grupo se activará la navegación hacia ese grupo. Los movimientos de grupos son monitorizados por el menú superior de navegación.
- **Proyectos:** Son los elementos del libro naranja y en el cuadrado que delimita el elemento contiene dos elementos una i y un símbolo de un indicador. El símbolo de la i referencia a la información del proyecto y el símbolo del indicador nos lleva a realizar una ejecución sobre el proyecto.

Si en el proceso de navegación se pulsa “Nuevo grupo” o “Nuevo proyecto” (los cuales están localizados en el menú lateral izquierdo) significará que se quiere crear un proyecto o un subgrupo en el grupo ubicado mediante la navegación. El sistema comprobara los permisos del grupo y el rol del usuario, para ver si es posible la realización de la creación.

8.3. Crear un grupo:

Tras situarnos en un grupo en el que queramos crear un subgrupo, se pulsará el elemento del menú lateral izquierda denominado “Nuevo grupo” y automáticamente la aplicación se movilizará para disponer un formulario para crear un grupo.

En el formulario se encontrará todos los datos que se deberá introducir para crear un grupo. Hay que tener en cuenta que el nombre de un grupo es inmutable y que no se podrá cambiarse. La descripción sirve para introducir lo que quiera el usuario y los distintos tipos de grupo identifican la privacidad del grupo:

- **Público:** Cualquiera que este navegando por un grupo y que contenga un grupo público podrá visualizar este grupo.

- **Privado:** Solo puede ser visualizado por los miembros que sean invitados por el dueño. El administrador también podrá acceder.

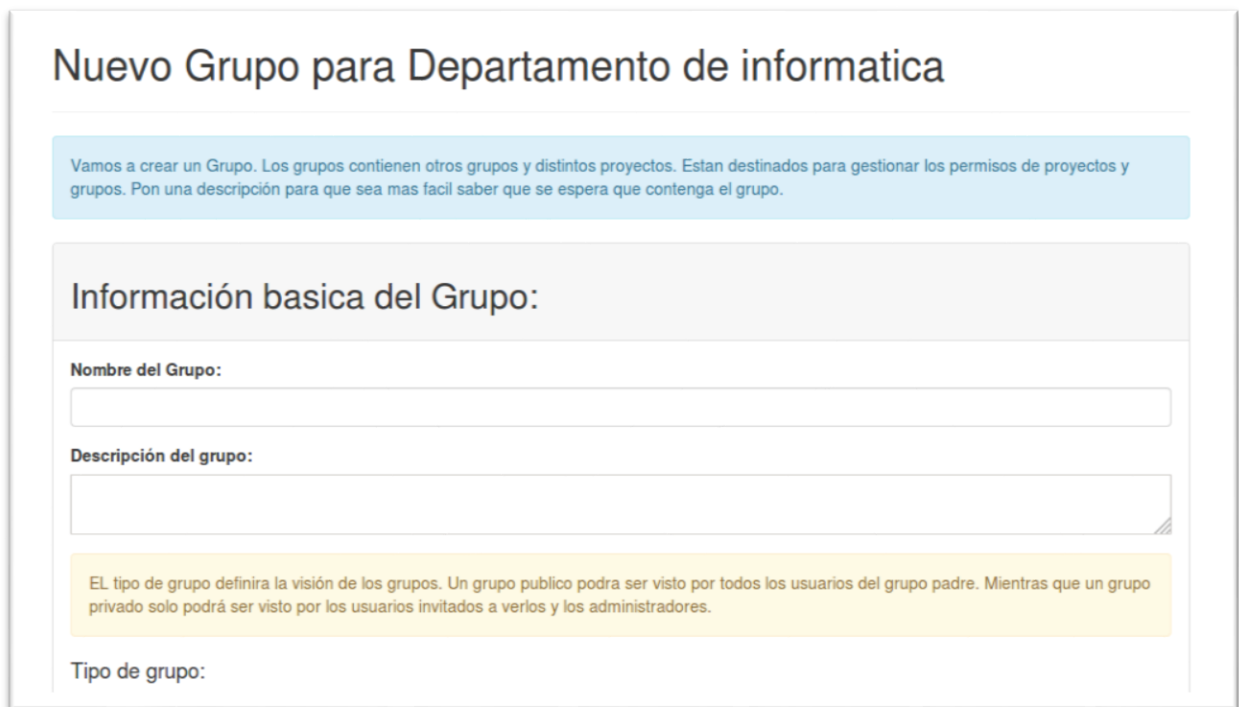


Ilustración 8-6 Nuevo grupo

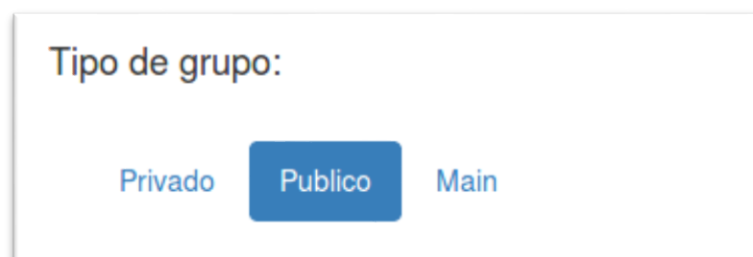


Ilustración 8-7 Nuevo grupo: Tipo grupo

Tras crear los datos básicos del grupo como su nombre y tipo se deberá configurar los permisos del grupo. Los permisos dan autorización a acciones sobre ejecuciones, proyectos, otros grupos y miembros del grupo. Los permisos son heredados por el grupo del grupo padre y cabe destacar que los permisos son de SI o NO. Si un permiso es negado por el padre, el grupo a crear no podrá poder modificar la negación.



Ilustración 8-8 Nuevo grupo: Permisos

Para terminar, solo tendremos que comprobar los datos y darle al botón crear. Saldrá un mensaje explicativo si la creación se ha realizado.

8.4. Crear un proyecto:

Para crear un proyecto habrá que navegar por el grupo que deseemos y después se deberá pulsar en el elemento del menú lateral izquierdo denominado “Nuevo proyecto”. Habrá que tener en cuenta los permisos para poder realizar la creación o la asociación. No obstante, si no tenemos los permisos la aplicación nos avisara de que no tenemos esos permisos.

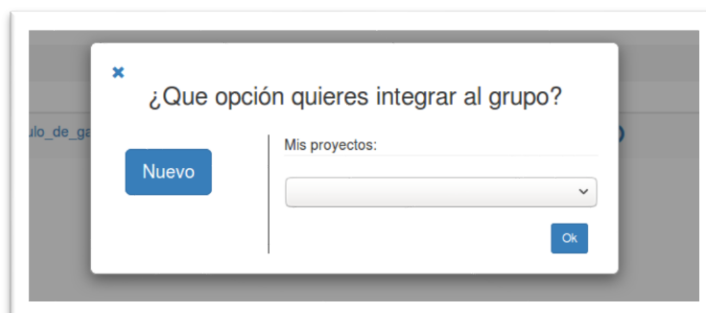


Ilustración 8-9 Nuevo proyecto: Integración o creación

Cuando se pulse el elemento del menú, se desplegará una ventana emergente como el de la figura 5-9. En la ventana se preguntará si se quiere agregar un proyecto de los realizados por el usuario. Para la agregación de proyectos existentes, el sistema nos dispondrá un listado de los proyectos propios del usuario y después el usuario solo tendrá que seleccionar un proyecto y pulsar el botón OK.

La aplicación también permite realizar la creación de un proyecto desde cero. Primero, la aplicación se movilizará para desplegar un formulario que proporcione los medios para introducir toda la información de un proyecto nuevo.



Ilustración 8-10 Nuevo proyecto: principal

Cabe destacar que el nombre del proyecto es inmutable, así como su tipo, pero eso será explicado junto con los demás datos:

- **Datos básicos:** Son el nombre, descripción, descripción de la entrada y descripción de la salida. Salvo el nombre estos datos son mutables.
- **Datos de repositorio:** Estos datos sirven para conectar al repositorio GIT, cabe destacar que es obligatorio su introducción y son la URL, el usuario y la contraseña.
- **Tipo de proyecto:** Los tipos de proyecto determinan en que lenguaje se encuentra el proyecto y cuál va a ser su ejecución.

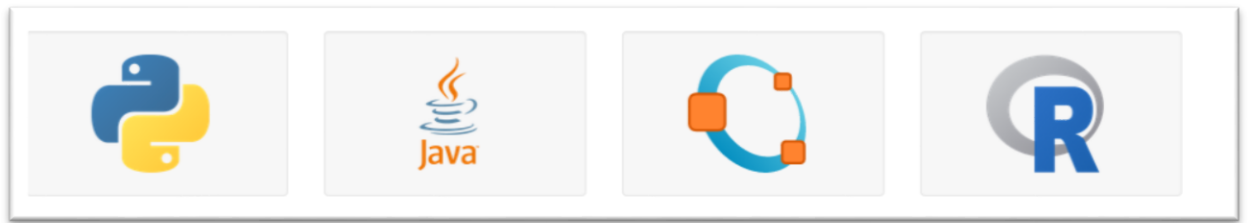


Ilustración 8-11 Nuevo grupo: tipo de proyecto

- **Datos de ejecución:** Los datos de ejecución determinan toda la configuración de la ejecución del proyecto. Estos datos son el programa principal, la respuesta y los inputs. El programa principal es la ubicación del fichero donde se va a ejecutar el programa, la respuesta es el nombre que va a tener el archivo con la respuesta de la ejecución y los inputs por defecto son las entradas que tendrá por defecto el proyecto.

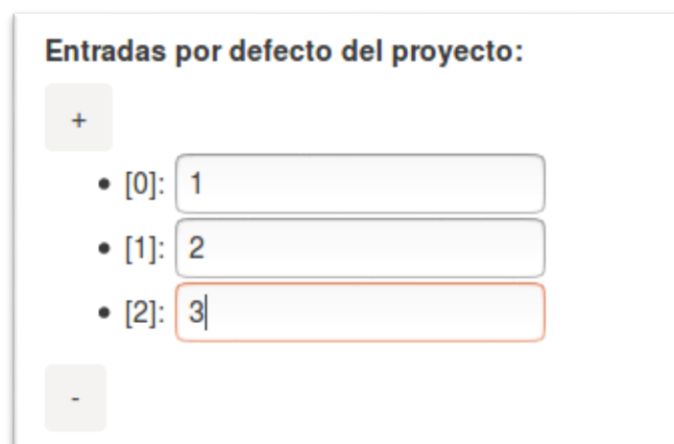


Ilustración 8-12 Nuevo grupo: Entradas por defecto

Por ultimo solo se deberá revisar los datos y pulsar el botón crear proyecto. La aplicación realizará la acción desplegando una ventana emergente informando del estado de la acción.

8.5. Crear una ejecución:

Para crear una ejecución deberemos navegar primero por los grupos, buscando un proyecto que podamos ejecutar. Tras eso deberemos pulsar el icono del indicador y se desplegará una ventana de ejecución. Esta ventana estará dividida en dos partes: Una parte la entrada y otra la salida (bloqueada).

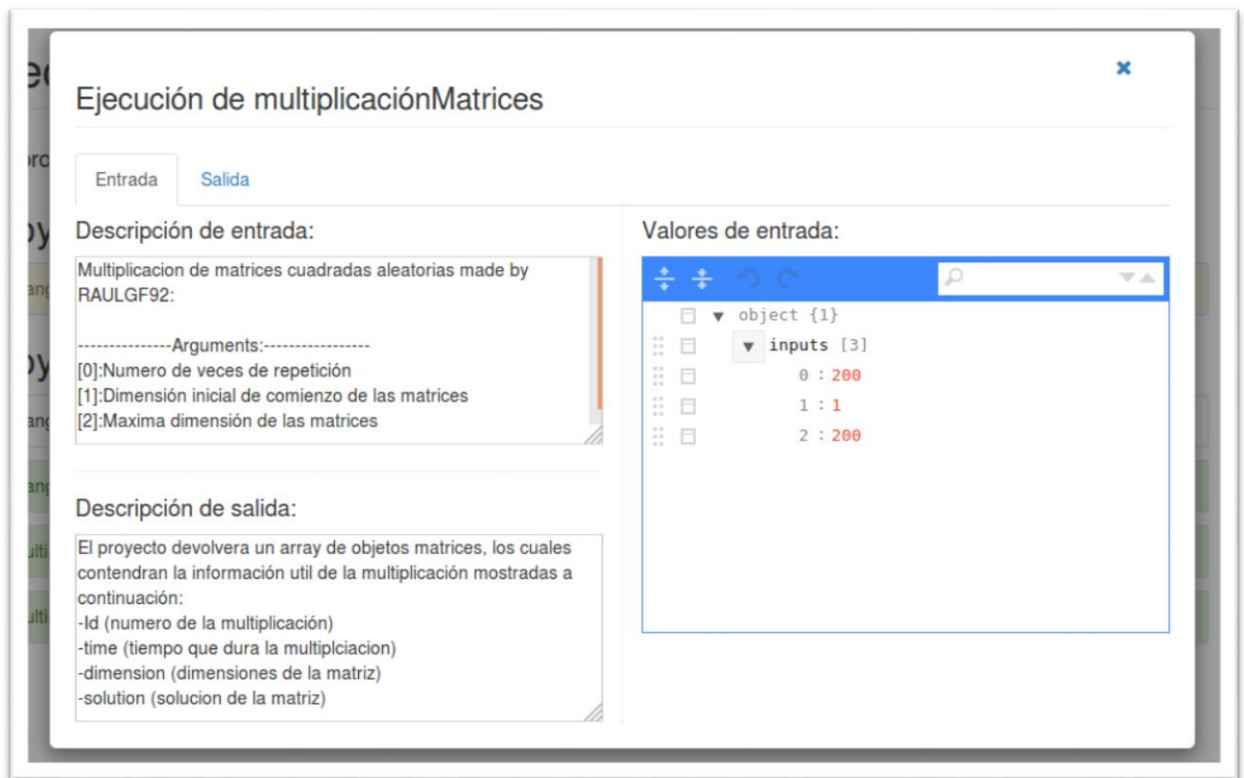



Ilustración 8-13 Nueva ejecución

En la sección “entrada” se presentará la descripción de la entrada de datos, es decir, los datos que se deberá poner en la sección de la derecha. Y debajo la descripción de la salida, es decir lo que se espera que salga de la ejecución. La sección de la derecha servirá para introducir datos de manera sencilla en un lenguaje máquina que entiende el usuario y la máquina denominado JSON. Si le das al cuadro derecho se desplegará un menú. En él se deberá pulsar la opción “+Append” e ir siguiendo con cada elemento hasta tener todos los que se desean. Tras eso se le dará al botón ejecutar y la ejecución se iniciará.

8.6. Gestionar un proyecto propio:


Cuando queramos modificar alguna opción de un proyecto deberemos acceder al espacio del menú “Proyectos”. En él, se encuentran todos los grupos creados por el usuario en cuestión. Solo deberemos pulsar en el grupo que quiera saber la información de él. La aplicación se movilizará y desplegará una ventana modal con toda la información del proyecto.

Mis proyectos:




numerosPrimos

Cuenta numeros primos de A a B



multiplicaciónMatrices




Primer proyecto totalmente funcional en Uniapi, tanto en codigo como en estructura dentro del sistema. Este proyecto multiplica matrices totalmente aleatorias de dimensiones dadas por el usuario. Provee la información básica para poder generar estudios.



Prueba1(NOEJECUTAR)

Triangulo de gauss, no ejecutar. Es el primer proyecto que entrara en la Base de datos.

Ilustración 8-14 Proyectos



multiplicaciónMatrices:

Primer proyecto totalmente funcional en Uniapi, tanto en código como en estructura dentro del sistema. Este proyecto multiplica matrices totalmente aleatorias de dimensiones dadas por el usuario. Provee la información básica para poder generar estudios.

Url del repositorio GIT:

`https://raulgf92@bitbucket.org/raulgf92/multiplicacionmatrices.git`

Usuario del repositorio:

`raulgf92@gmail.com`

Contraseña del repositorio:

.....

Nombre del programa principal proyecto:

`main.py`

Nombre del archivo de respuesta del proyecto:

`response.json`

Entradas por defecto del proyecto:

Ilustración 8-15 Gestión proyecto

Toda la información podrá ser modificada. Al final de la modificación solo tendrá que darle al botón del lápiz de arriba a la derecha para realizar una modificación del proyecto. Por

el contrario, si quiere cancelar la modificación solo deberá cerrar la aplicación. La aplicación mandará un aviso emergente sobre el estado de la acción.

Si desea eliminar el proyecto deberá pulsar el botón de arriba a la derecha con la forma de un cubo de la basura. La aplicación preguntará si desea eliminar el proyecto. Tras la aceptación la aplicación mandará un aviso emergente con el estado de la acción.

8.7. Gestionar un grupo propio:

Cuando se quiera gestionar o modificar un grupo, se deberá ir al espacio destinado a albergar el conjunto de los grupos creados por el usuario denominado “grupos”. En él, se encuentra todos los grupos creados por el usuario. Para poder gestionar cualquier grupo solo se tendrá que pulsar en el grupo. Y la aplicación se movilizará para generar una ventana emergente con la información del grupo.

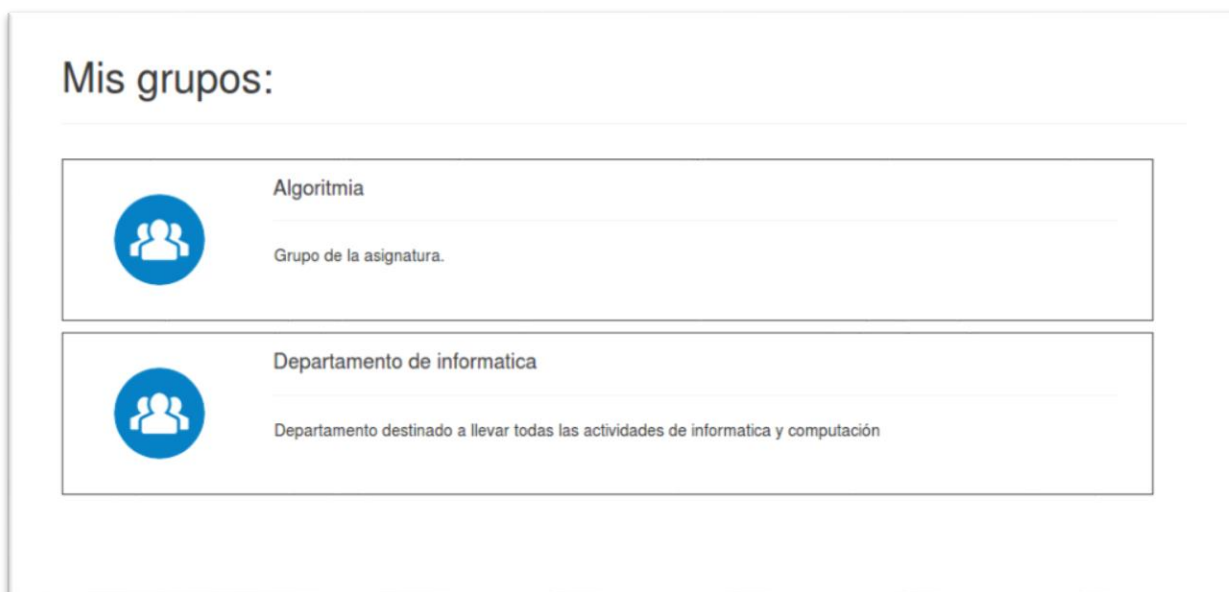
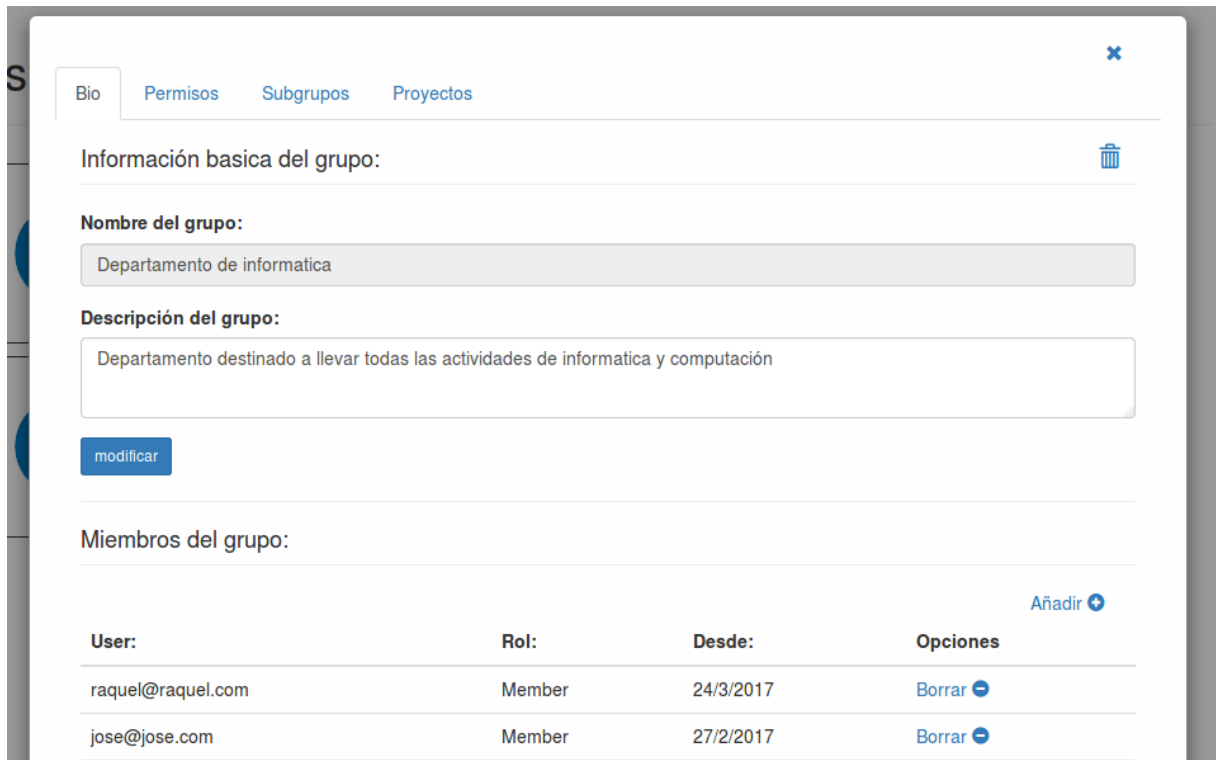


Ilustración 8-16 Mis Grupos



Bio Permisos Subgrupos Proyectos

Información básica del grupo:

Nombre del grupo:
Departamento de informatica

Descripción del grupo:
Departamento destinado a llevar todas las actividades de informatica y computación

modificar

Miembros del grupo:

Añadir +

User:	Rol:	Desde:	Opciones
raquel@raquel.com	Member	24/3/2017	Borrar -
jose@jose.com	Member	27/2/2017	Borrar -

Ilustración 8-17 Gestión grupo

La ventana emergente esta ordenada en pestañas donde se gestionará cada diversos conjuntos de elementos. Cada uno será una lista mostrada como en la figura 33 que contendrá distinta información y dos acciones la de añadir o eliminar. Si se le da al botón de añadir se desplegará una opción donde podremos indicar lo que queremos añadir.

8.8. Información básica:

Para la información básica solo podremos modificar la descripción del grupo. Donde escribiremos lo que queramos y pulsaremos a modificar. Tras pulsar el botón la aplicación informara al usuario como ha salido la modificación. En la parte superior tenemos el botón de forma de cubo de basura, este botón realizara la eliminación del grupo. Una eliminación del grupo. Conlleva la eliminación de todos los subgrupos hijos de él y sus integraciones de proyectos (pero no los proyectos).

8.9. Miembros:

Los miembros es una lista con el conjunto de todos los miembros de un grupo, en el podremos añadir miembros pulsando al botón de arriba “Añadir”. La aplicación desplegará un formulario donde podremos introducir el email de usuario del sistema que quieres invitar. Por el contrario, si se quiere eliminar un usuario, solo se tendrá que pulsar en el botón “Borrar” asociado a su fila. La eliminación consiste en la expulsión del usuario del grupo, pero no la eliminación de los elementos creados o integrados por él.

8.9.1. Subgrupos:

Subgrupos es una lista con todos los subgrupos de nivel 1 (hijos directos) que dependen del grupo. En la lista podremos eliminar cualquier grupo albergado en el grupo, sea público o privado. También podremos crear un grupo, lo que hará la aplicación será llevarnos al punto 7.1.3 para realizar una creación sobre este grupo. La eliminación de un subgrupo, es la eliminación de todos los subhijos del subgrupo y sus integraciones de proyectos.

8.9.2. Permisos:

Los permisos son celdas iguales que cuando se crea los permisos, en ellos se pueden modificar los permisos y mandar a la aplicación modificarlos. Una modificación de los permisos genera una modificación en recursión, es decir, una modificación de todos los hijos y sus hijos...

8.9.3. Gestionar una ejecución propia:

Para gestionar una ejecución propia, versión contenido, su estado... necesitamos entrar en el espacio de Ejecuciones, esta se puede acceder a través del menú lateral de la izquierda de la aplicación. La aplicación se moviliza generando un espacio con listas informativas de las ejecuciones. Estas se organizan en no finalizadas, la lista de arriba con elementos de color amarillo. Y la lista de finalizadas con elementos de color verde para éxito y rojo para fallo.

Ejecución de proyectos:

Mis proyectos:

Ejecutar

Proyectos en ejecución:

triangulo_de_gauss

Proyectos finalizados:

triangulo_de_gauss

triangulo_de_gauss

multiplicaciónMatrices

multiplicaciónMatrices

Si pulsamos en cualquier elemento se nos enviara a una ventana emergente con la gestión de la ejecución:

- En ejecución: Se nos enviará a la ventana de salidas, la respuesta estará vacía, la cual es una ventana igual que la descrita en la creación de nueva ejecución. Esta ventana traduce el lenguaje maquina a una interfaz que ayuda a la comprensión del usuario. También nos indicara la cuenta de tiempo que lleva la ejecución.

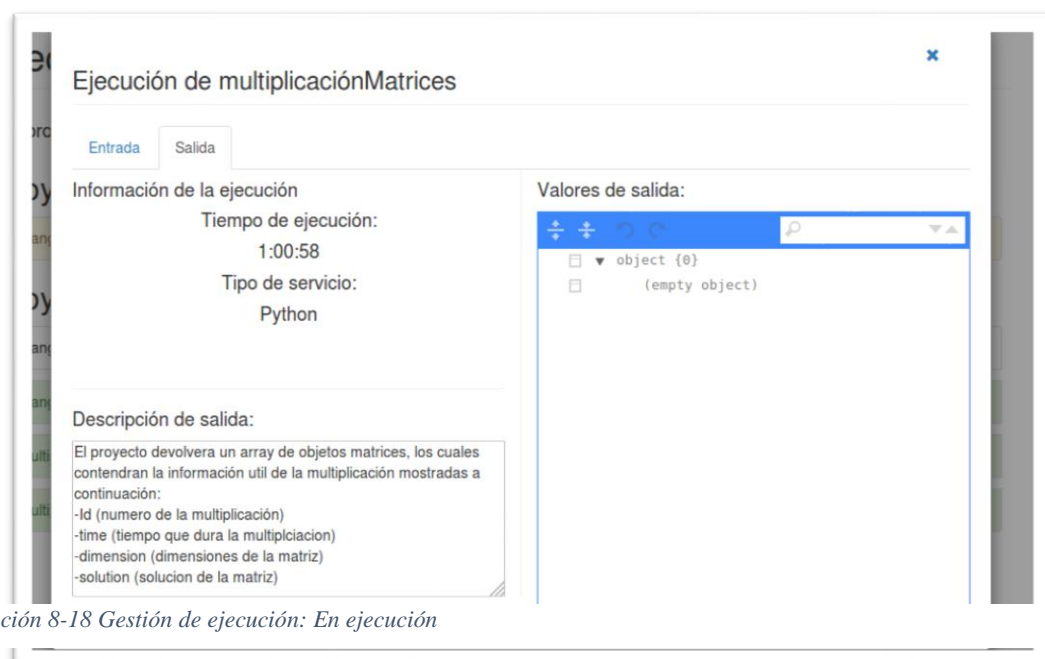


Ilustración 8-18 Gestión de ejecución: En ejecución

- Finalizado: Finalizado nos enviará a la ventana de salidas, la respuesta estará llena de los datos de respuesta de la ejecución del proyecto, salvo si ha finalizado con fallo que estará vacío. La ventana también nos dará el tiempo de la ejecución.

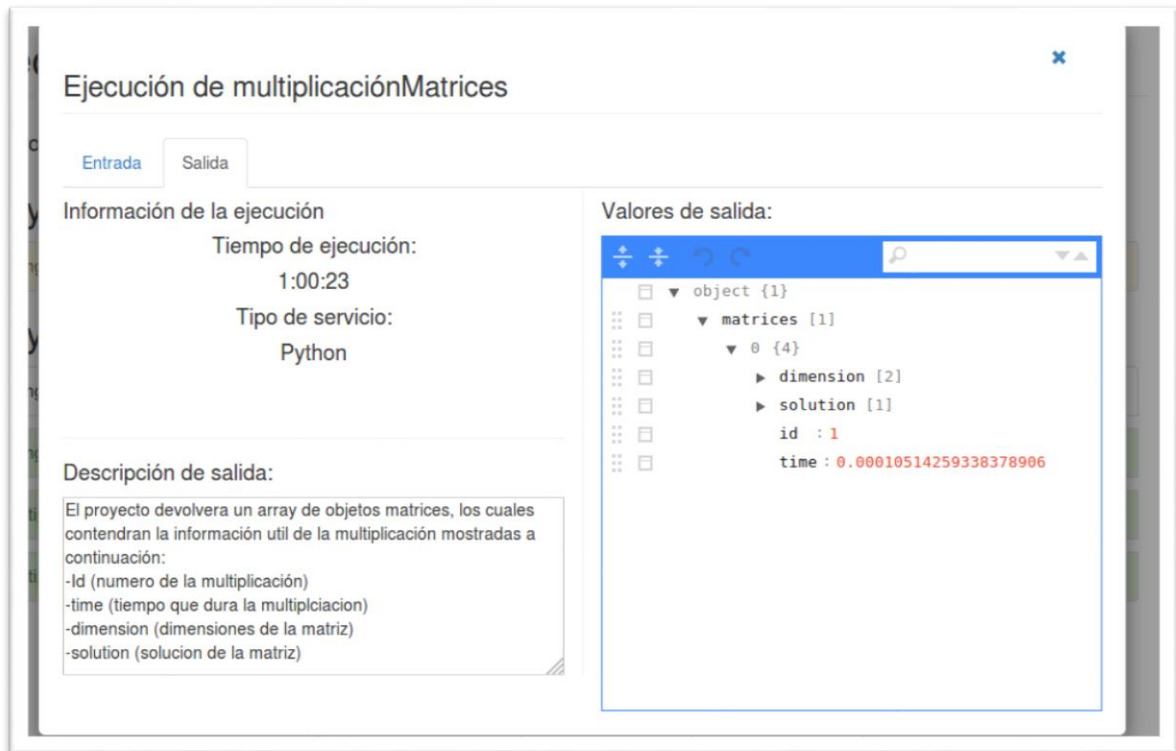


Ilustración 8-19 Gestión de ejecución:

Finalizado