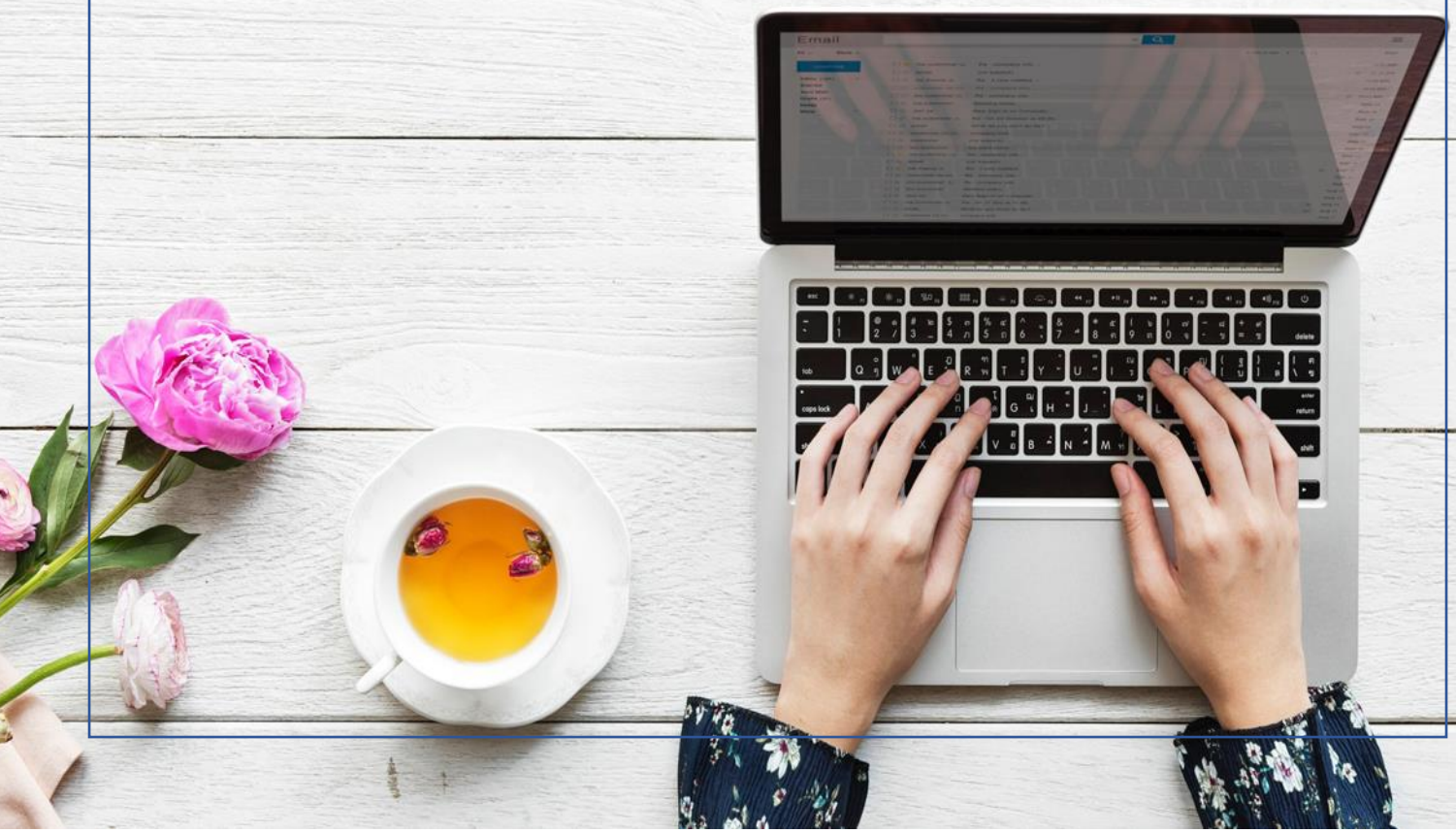


# APLICACIÓN WEB DE ADMINISTRACION DE BASE DE DATOS

Pablo Cruz Garrido



# CONTENIDO

Descripción del proyecto .....	3
2. HERRAMIENTAS EMPLEADAS.....	4
• SPRING BOOT .....	4
• SERVIDOR WEB .....	4
• BASE DE DATOS.....	4
• LENGUAJES DE PROGRAMACION.....	5
• LENGUAJES DE DISEÑO.....	6
3. PATRON DE ARQUITECTURA.....	7
MODELO-VISTA-CONTROLADOR.....	7
3. BASE DE DATOS.....	8
BASE DE DATOS.....	8
4º FUNCIONAMIENTO DE LA APLICACIÓN WEB.....	9
• INICIO.....	9
✓ BACKUPS.....	10
✓ USUARIOS.....	11
✓ LOGS.....	13
✓ COPIA SEGURIDAD DIARIA .....	13
CONFIGURAR PARAMETROS DE LA APLICACION .....	15
✓ SUBTITULO .....	15
✓ PARAMETROS DE LA BASE DE DATOS .....	15
✓ OTRAS RUTAS ABSOLUTAS.....	15
✓ INSTALACION DEL CLIENTE DE DRIVE .....	16
USAR OTROS CREDENCIALES DE DRIVE .....	18



## 1. DESCRIPCIÓN DEL PROYECTO

Este proyecto ha consistido en crear una aplicación web con secciones para backups de la base, usuarios y logs.

En la sección de “backups” podemos crear una copia en el instante en local y se subirá automáticamente a drive, controlar quien ha hecho cada copia y cuando, crear una nueva backup en local y en la nube, borrar la copia local o en el drive, restaurar la copia en la base de datos y subir la copia al drive si la habíamos borrado antes de allí o descargarla en local si la borramos en local.

En la sección de usuarios podemos hacer las operaciones de CRUD (create, read, update, delete) de los usuarios.

En la sección logs podremos ver los registros de todas las operaciones, fechas, ips desde donde se inició sesión, sobre que archivo...

Una vez el usuario cierre sesion sera redirigido a el login con un mensaje de éxito.

## 2. HERRAMIENTAS EMPLEADAS

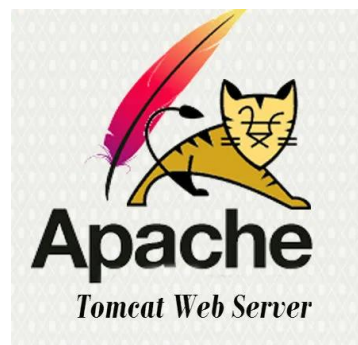
### 2.1 SPRING BOOT

Primero hay que hablar de “Spring Framework” el cual es un framework código abierto para hacer aplicaciones de Groovy, Kotlin y Java muy potente. Sin embargo, la configuración inicial y el desplegar las aplicaciones era muy difícil y aquí es donde Spring boot surge para facilitar la vida a los desarrolladores. Esto lo consigue en gran parte a una serie de dependencias llamadas “Starters” que añadiremos dependiendo de nuestro proyecto.



### 2.2 SERVIDOR WEB

En este caso hemos usado Apache Tomcat que es un contenedor Java Servlet...¿Y que es un servlet? Esto permite a los servidores web manejar contenido web dinámico basado en java mediante los parametros de petición que envía el navegador. Este está integrado en Spring boot por defecto



### 2.3 BASE DE DATOS

En este caso hemos usado MariaDB. Este es derivado de MySQL, tiene licencia GPL.

Surgió cuando MySQL fue comprado por Oracle, así surgió este fork para la comunidad asegurarse una versión de este con GPL. Recordemos que GPL garantiza a sus usuarios usar, estudiar, compartir y modificar el software.



## 2.4 LENGUAJES DE PROGRAMACION

- Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.



- Javascript

Lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., probablemente JavaScript está involucrado.





## 2.5 LENGUAJES DE DISEÑO

- HTML

HTML es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas. Este lenguaje nos ofrece una gran adaptabilidad, una estructuración lógica y es fácil de interpretar tanto por humanos como por máquinas.



- CSS

Las siglas **CSS** (*Cascading Style Sheets*) significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar **estilos** (colores, formas, márgenes, etc...) a uno o varios documentos (*generalmente documentos HTML, páginas webs*) de forma masiva.



- Bootstrap

Framework CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

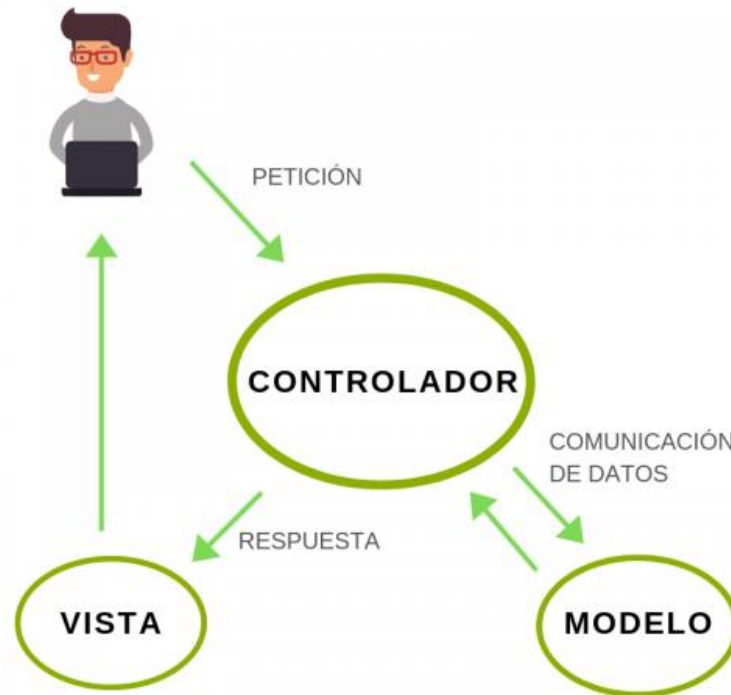
El framework combina [CSS](#) y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces. Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.



### 3. PATRON DE ARQUITECTURA

## MODELO-VISTA-CONTROLADOR

Al usar Spring Boot, la arquitectura usada es el patrón “modelo-vista-controlador” esto nos facilita mucho la vida ya que separamos nuestra aplicación en componentes y por ejemplo si modificamos nuestra base de datos no es necesario ir cambiándolo archivo por archivo, con cambiarlo en la entidad será más que suficiente en muchos casos. Muy bien..pero ¿qué son esos componentes?



## MODELO

Se suele encargar de los datos, es quien está en contacto con la base de datos y las CRUD van en esta capa.

## VISTA

Se encarga de la parte que el usuario ve, esta recibe los datos. Se podría decir que es el frontend de la aplicación.

## CONTROLADOR

Este recibe las peticiones del usuario y ejecuta las acciones correspondientes en el modelo y le devuelve los datos a la vista.

### 3. BASE DE DATOS

#### BASE DE DATOS

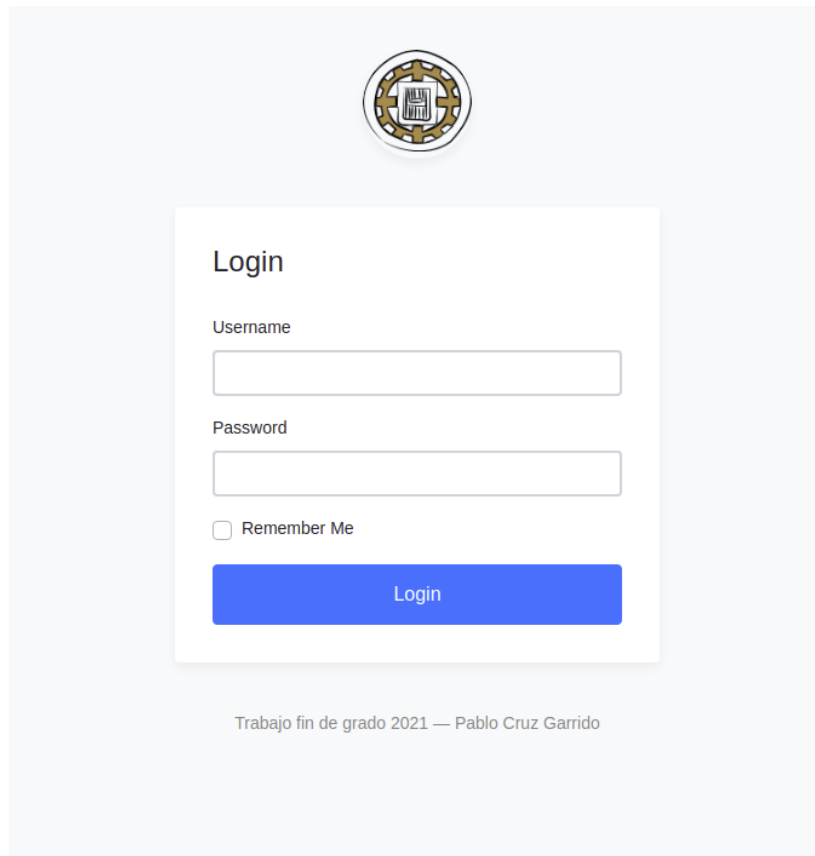
Texto



## 4º FUNCIONAMIENTO DE LA APLICACIÓN WEB

- INICIO

Cada vez que arranques el proyecto y vayas a meterte en cualquier página automáticamente te redirigirá a el login hasta que inicies sesión.

A screenshot of a web application's login page. At the top center is a circular logo with a gear-like border and a central emblem. Below the logo is a white rectangular login form. The form has the title "Login" at the top. It contains two input fields: "Username" and "Password". Below the password field is a checkbox labeled "Remember Me". At the bottom of the form is a blue button with the text "Login". Below the form, centered, is the text "Trabajo fin de grado 2021 — Pablo Cruz Garrido".

Trabajo fin de grado 2021 — Pablo Cruz Garrido

Esto ocurre gracias al uso de “Spring security” el cual gestiona todo lo relacionado a la seguridad de la aplicación como los roles de los usuarios, que roles se necesitan para acceder a un recurso...

En cuanto a los roles de esta aplicación he creado dos: user y admin el cual veremos sus diferencias explicando la sección backup.

## ✓ BACKUPS

Esta sección es la que nos redirige la aplicación al iniciar sesión y es donde veremos la diferencia entre los roles.

Los controller que utilizamos para esta sección son:

Veamos como lucen los dos roles:

### ➤ ROL USER

The screenshot shows the user interface for a user role. At the top, there is a dark header with a logo on the left and 'Backups' and 'Desconectarse' links on the right. Below the header, a greeting 'Hola Jose!' is displayed. The main content area features a table with columns: 'Archivo', 'Usuario', 'Fecha', 'Acción local', and 'Acción drive'. The table lists nine backup entries, each with a filename, the user 'admin', a timestamp, and download icons for both local and drive actions. A search bar is located at the top right of the table. Below the table, there is a pagination bar showing 'Mostrando 1 a 9 de 9 Entradas' and navigation links 'Anterior', '1', and 'Siguiente'. At the bottom, there is a blue button labeled 'Backup' with a plus icon.

Archivo	Usuario	Fecha	Acción local	Acción drive
copia_seguridad_2021-06-07_04-48-54.sql	admin	2021-06-07 16:48:54.0	Download	Download
copia_seguridad_2021-06-08_09-53-09.sql	admin	2021-06-08 09:53:09.0	Download	Download
copia_seguridad_2021-06-08_09-53-48.sql	admin	2021-06-08 09:53:48.0	Download	Download
copia_seguridad_2021-06-09_04-48-58.sql	admin	2021-06-09 16:48:58.0	Download	Download
copia_seguridad_2021-06-09_06-06-38.sql	admin	2021-06-09 18:06:38.0	Download	Download
copia_seguridad_2021-06-09_09-35-37.sql	admin	2021-06-09 09:35:37.0	Download	Download
copia_seguridad_2021-06-09_09-37-40.sql	admin	2021-06-09 09:37:40.0	Download	Download
copia_seguridad_2021-06-09_10-34-27.sql	admin	2021-06-09 10:34:27.0	Download	Download
copia_seguridad_2021-06-12_10-05-20.sql	admin	2021-06-12 22:05:20.0	Download	Download

El usuario como vemos, solo tiene acceso a la sección de backups donde puede crear un backup nuevo, descargarse uno de google drive o de los locales. No puede hacer nada más.

### ➤ ROL ADMIN

The screenshot shows the user interface for an admin role. The header is similar to the user role, but includes additional links: 'Backups', 'Usuarios', 'Logs', and 'Desconectarse'. The greeting is 'Hola admin!'. The table structure is the same, but the 'Acción local' and 'Acción drive' columns contain more icons, including a red square icon with a white 'X' and a blue square icon with a white 'X', indicating additional actions available to the admin. The pagination and 'Backup' button are also present.

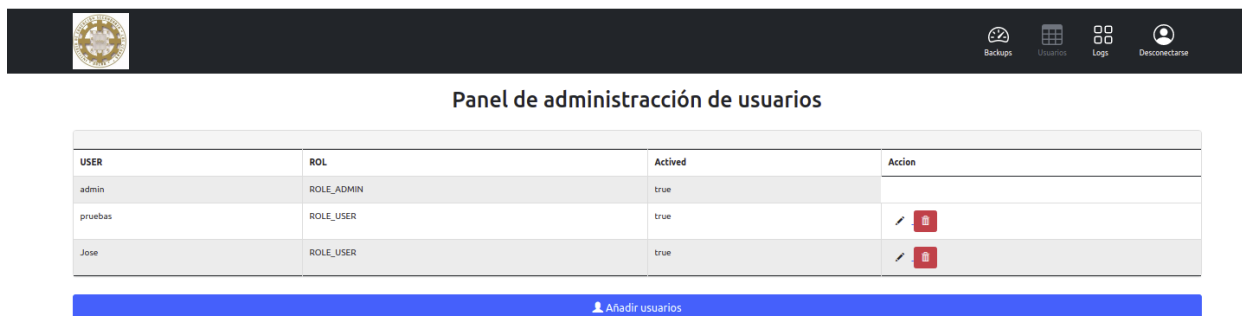
Archivo	Usuario	Fecha	Acción local	Acción drive
copia_seguridad_2021-06-07_04-48-54.sql	admin	2021-06-07 16:48:54.0	Download, Delete, Upload	Download, Delete, Upload
copia_seguridad_2021-06-08_09-53-09.sql	admin	2021-06-08 09:53:09.0	Download, Delete, Upload	Download, Delete, Upload
copia_seguridad_2021-06-08_09-53-48.sql	admin	2021-06-08 09:53:48.0	Download, Delete, Upload	Download, Delete, Upload
copia_seguridad_2021-06-09_04-48-58.sql	admin	2021-06-09 16:48:58.0	Download, Delete, Upload	Download, Delete, Upload
copia_seguridad_2021-06-09_06-06-38.sql	admin	2021-06-09 18:06:38.0	Download, Delete, Upload	Download, Delete, Upload
copia_seguridad_2021-06-09_09-35-37.sql	admin	2021-06-09 09:35:37.0	Download, Delete, Upload	Download, Delete, Upload
copia_seguridad_2021-06-09_09-37-40.sql	admin	2021-06-09 09:37:40.0	Download, Delete, Upload	Download, Delete, Upload
copia_seguridad_2021-06-09_10-34-27.sql	admin	2021-06-09 10:34:27.0	Download, Delete, Upload	Download, Delete, Upload
copia_seguridad_2021-06-12_10-05-20.sql	admin	2021-06-12 22:05:20.0	Download, Delete, Upload	Download, Delete, Upload

Ahora vemos que tenemos muchas más acciones sobre los archivos y 3 apartados en el menú:




Backups, usuarios y logs.

## ✓ USUARIOS

Los controller que utilizamos para esta sección son:



The screenshot shows a web application interface for user management. At the top, there is a dark navigation bar with a logo on the left and icons for 'Backups', 'Usuarios', 'Logs', and 'Desconectarse' on the right. Below the navigation bar, the title 'Panel de administración de usuarios' is centered. The main content area features a table with the following data:

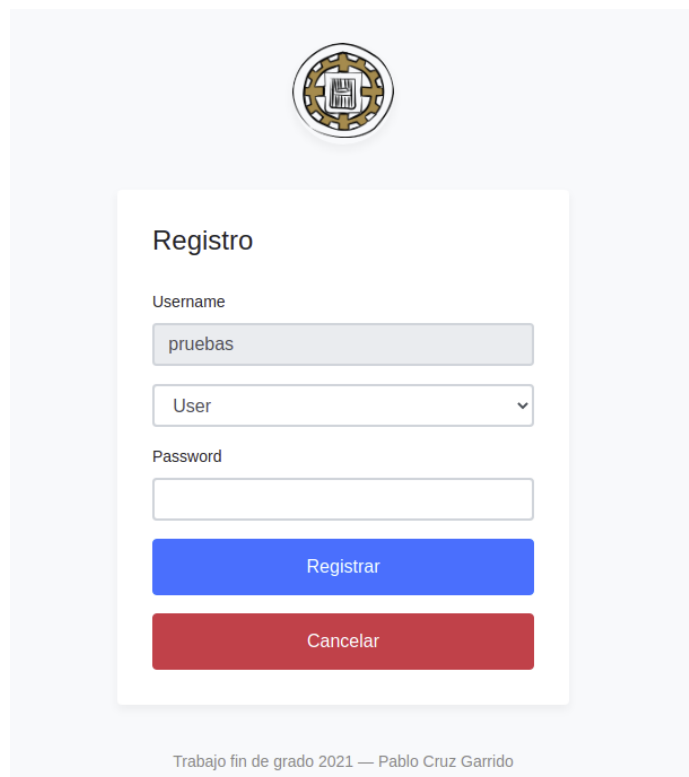
USER	ROL	Actived	Action
admin	ROLE_ADMIN	true	
pruebas	ROLE_USER	true	 
Jose	ROLE_USER	true	 

Below the table, there is a blue button labeled 'Añadir usuarios'.

Aquí se muestra los usuarios donde podemos hacer la operación de CRUD menos para el actual.

Podemos editar y borrar usuarios.

Así es el formulario de editar usuarios:




The screenshot shows a 'Registro' (Registration) form. At the top, there is a circular logo. The form contains the following fields and buttons:

- Username:** A text input field with the value 'pruebas'.
- User:** A dropdown menu with the value 'User'.
- Password:** A text input field.
- Registrar:** A blue button.
- Cancelar:** A red button.

At the bottom of the page, there is a footer that reads: 'Trabajo fin de grado 2021 — Pablo Cruz Garrido'.

Así es el formulario de registro:



## Registro

Username

Password

Seleccionar rol de usuario ▼

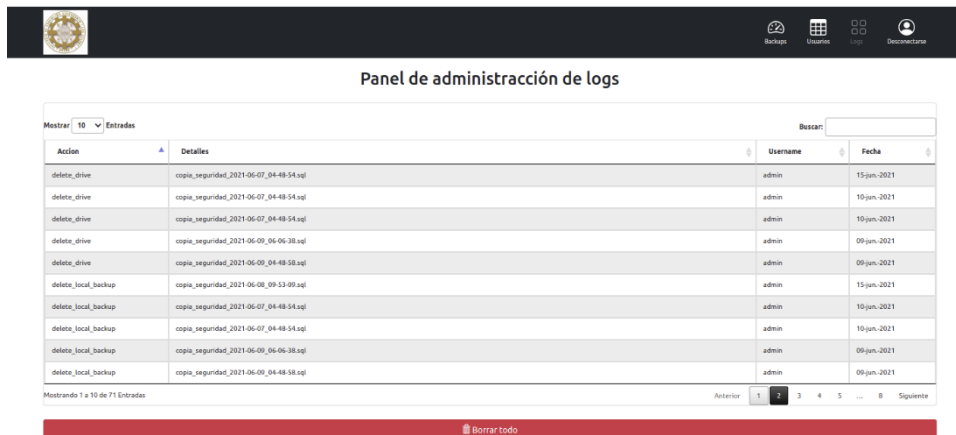
Registrar

Cancelar

Trabajo fin de grado 2021 — Pablo Cruz Garrido

## ✓ LOGS

Los controller que utilizamos para esta sección son:



Acción	Detalles	Username	Fecha
delete_drive	copia_seguridad_2021-06-07_04-48-54.sql	admin	15-jun-2021
delete_drive	copia_seguridad_2021-06-07_04-48-54.sql	admin	10-jun-2021
delete_drive	copia_seguridad_2021-06-07_04-48-54.sql	admin	10-jun-2021
delete_drive	copia_seguridad_2021-06-09_06-06-38.sql	admin	09-jun-2021
delete_drive	copia_seguridad_2021-06-09_04-48-58.sql	admin	09-jun-2021
delete_local_backup	copia_seguridad_2021-06-08_09-53-09.sql	admin	15-jun-2021
delete_local_backup	copia_seguridad_2021-06-07_04-48-54.sql	admin	10-jun-2021
delete_local_backup	copia_seguridad_2021-06-07_04-48-54.sql	admin	10-jun-2021
delete_local_backup	copia_seguridad_2021-06-09_06-06-38.sql	admin	09-jun-2021
delete_local_backup	copia_seguridad_2021-06-09_04-48-58.sql	admin	09-jun-2021

Mostrando 1 a 10 de 71 Entradas

Anterior 1 2 3 4 5 ... 8 Siguiente

Borrar todo

Donde vemos la acción, detalles (que puede ser el archivo sobre el que se realiza o desde que ip se inicia sesión) el usuario que la ha hecho y la fecha.

Abajo hay un botón donde podemos borrar todo.

Muy bien, pero... ¿Cómo se crean estos logs?

Por ejemplo, en el caso del log para controlar cuando borramos un archivo en local lo que hacemos es irnos al controller donde se halla el método que borra el archivo en local, hacer un autowired del repositorio de los logs, invocar al repositorio y decirle que guarde el log de esa acción.

## ✓ COPIA SEGURIDAD DIARIA

Hay una copia de seguridad que se ejecuta todas las noches a las 12 automáticamente por Spring Boot, esta puede ser modificado mediante el archivo `"/proyecto-gestion-base/src/main/java/proyecto/com/configuration/Mantenimiento.java"`

El método encargado de esa gestión es el siguiente:

```
//Hace la copia todas las noches a las 12
@Scheduled(cron = "0 0 0 * * *")
public void copiaProgramada() throws IOException {
    //Nombre archivo
    String fichero = new SimpleDateFormat("'copia_seguridad_'yyyy-MM-dd_hh-mm-ss'.sql'").format(new Date());
    /*Windows
    Runtime.getRuntime().exec("cmd /c start "+ViewConstant.PROYECTO+"src/main/resources/scripts/script_backup.bat "+fichero);*/
    //Linux BACKUP_SCL Local
    ProcessBuilder plocal = new ProcessBuilder("src/main/resources/scripts/script_backup.sh", fichero);
    plocal.start();
    //La nube
    ProcessBuilder pdrive = new ProcessBuilder("src/main/resources/scripts/script_drive_upload.sh", fichero);
    pdrive.start();
    backupRepository.save(new proyecto.com.entity.Backup(fichero, "SYSTEM", new Date(), true, true));
    String action = "new_backup";
    logService.debug(action, fichero);
}
```

Como vemos en la anotación Scheduled vemos una expresión cron, así que cambiando esa expresión por la deseada y lanzando otra vez la aplicación conseguiremos que las copias ahora se hagan cuando deseamos.

Otra cosa sobre esta clase es que es donde tenemos que meter las tareas programadas ya que sus anotaciones indica que la clase se encarga de eso, no podemos coger y poner un método programado en cualquier clase y esperar que funcione.

```
@Configuration
@EnableScheduling
public class Mantenimiento {
```

Con esto habríamos terminado la parte de la explicación del proyecto, en las siguientes paginas vamos a explicar cómo hacemos para que se puede usar esta aplicación web en otras cuentas de google drive, sobre otras bases con otros parámetros...



# CONFIGURAR PARAMETROS DE LA APLICACION

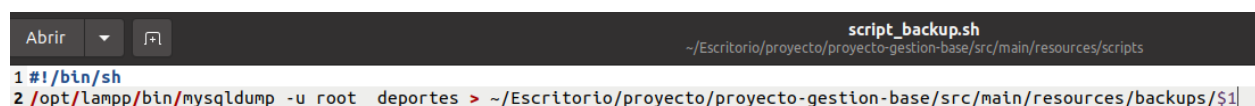
## ✓ SUBTITULO

Supongamos que queremos hacer funcionar esta aplicación con otra cuenta de drive, otra base de datos y/u otro ordenador, para ello tendríamos que seguir los siguientes pasos

## ✓ PARAMETROS DE LA BASE DE DATOS

Si queremos cambiar sobre qué base de datos se hacen las backups tenemos que modificar dos scripts de shell ubicados en: /proyecto-gestion-base/src/main/resources/scripts/script\_backup.sh y los archivos a cambiar son:

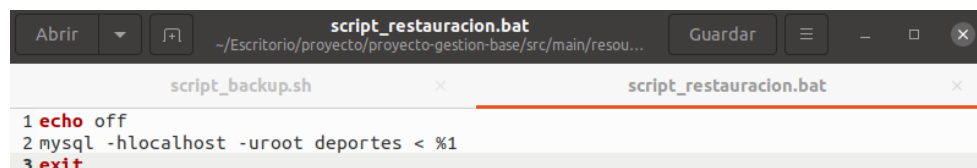
- script\_backup.sh :



```
1 #!/bin/sh
2 /opt/lampp/bin/mysqldump -u root deportes > ~/Escritorio/proyecto/proyecto-gestion-base/src/main/resources/backups/$1
```

Habría que cambiar el user, la base, poner contraseña si fuese necesario y cambiar la ruta absoluta donde se guarda la copia por donde guardemos nuestro proyecto ahora.

- script\_restauracion.sh



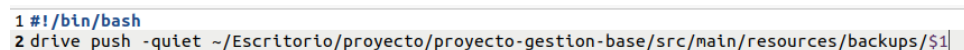
```
1 echo off
2 mysql -hlocalhost -uroot deportes < %1
3 exit
```

Volvería a ser tan fácil como poner las credenciales correctas y cambiar “deportes” por la base correspondiente.

## ✓ OTRAS RUTAS ABSOLUTAS

Ahora vamos a cambiar otros scripts con ruta absoluta:

- Script\_drive\_upload.sh



```
1 #!/bin/bash
2 drive push -quiet ~/Escritorio/proyecto/proyecto-gestion-base/src/main/resources/backups/$1
```

Este script se halla en el mismo sitio que los anteriores.

Ahora vamos a ver los archivos donde necesitamos modificar sus rutas que se hallan en “/proyecto-gestion-base/src/main/resources/backups”

- script\_drive\_download.sh

```
1 #!/bin/bash
2 cd ~/Escritorio/proyecto/proyecto-gestion-base/src/main/resources/backups
3 drive pull -quiet -files $1
```

- Script\_drive\_delete

```
1 #!/bin/bash
2 cd ~/Escritorio/proyecto/proyecto-gestion-base/src/main/resources/backups
3 drive delete -quiet $1
```

Con esto habríamos terminado lo relacionado a cambiar scripts.

## ✓ INSTALACION DEL CLIENTE DE DRIVE

Esto lo hacemos cuando no hemos instalado el cliente de drive previamente

### INSTALAMOS GO

Es necesario instalar el lenguaje go antes de instalar el drive, para ello en la consola ejecutamos:

- wget <https://storage.googleapis.com/golang/go1.6.2.linux-amd64.tar.gz>
- tar -C /usr/local -axvf go1.6.2.linux-amd64.tar.gz
- /usr/local/go/bin/go version

Con este último comando si se ha instalado correctamente nos mostrara la versión instalada.

Ahora debemos irnos al archivo “/etc/profile” y configurar las siguientes variables del entorno.

```
export GOPATH=$HOME/gopath
```

```
export PATH=$PATH:/usr/local/go/bin:$GOPATH:$GOPATH/bin
```

Verificamos el funcionamiento:

- . /etc/profile
- go version

Si imprime la versión de go, ya hemos instalado go correctamente.

```
pablo@pcg:~$ go version
go version go1.16.4 linux/amd64
```

### INSTALAR EL CLIENTE DRIVE

Ejecutamos el siguiente comando:

- go get -u github.com/odeke-em/drive/cmd/drive

## CONFIGURAR EL CLIENTE DRIVE

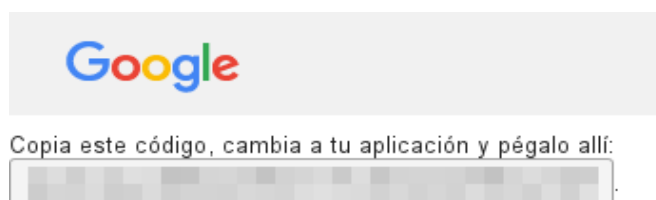
Con esto habríamos instalado el cliente de drive, ahora tenemos que introducirle nuestras credenciales y inicializarnos en nuestro directorio de trabajo.

- Ponemos el siguiente comando “drive init DIR”
- El dir lo sustituimos por la ruta absoluta hasta “/proyecto-gestion-base/src/main/resources/backups”

Tras hacer eso nos mostrara una url en la consola donde debemos acceder, iniciar sesion con nuestra cuenta google y darle a "Permitir".



Tras eso se genera un código que es el que debemos copiar



Si hacemos un `ls -la` sobre el directorio y vemos una carpeta oculta llamada `“.gd”` la operación ha sido un éxito.

```
mblo@pcg:~/Escritorio/proyecto/proyecto-gestion-base/src/main/resources/backups$ ls -a
.                  copia_seguridad_2021-05-08_10-09-19.sql  copia_seguridad_2021-06-09_06-06-38.sql  .gd
..                 copia_seguridad_2021-05-17-09-32-05.sql  copia_seguridad_2021-06-09-09-35-37.sql  script_drive_delete.sh
copia_seguridad_2021-04-30_10-43-56.sql  copia_seguridad_2021-06-07_04-48-54.sql   copia_seguridad_2021-06-09-09-37-40.sql  script_drive_download.sh
copia_seguridad_2021-05-01_01-13-21.sql  copia_seguridad_2021-06-08-09-53-48.sql   copia_seguridad_2021-06-09-10-34-27.sql
copia_seguridad_2021-05-04_07-45-20.sql  copia_seguridad_2021-06-09_04-48-58.sql   copia_seguridad_2021-06-12_10-05-20.sql
```

## ¿Por qué inicializamos Google drive en ese directorio?

Esto lo hacemos porque es el directorio desde donde vamos a descargar, subir y borrar archivos.

Si quisiéramos algo más de este comando o ayuda recurrimos a su ayuda con “drive -h”

Con esto ya tendríamos todo listo para empezar a usar nuestra aplicación con otra base y/o cuenta distinta.

## USAR OTROS CREDENCIALES DE DRIVE

En caso de tener un directorio asociado a una cuenta de drive sería tan sencillo como borrar la carpeta .gd y seguir los pasos indicados en la sección “configurar el cliente drive”