

Universidad de Granada



Cloud Computing: Servicios y Aplicaciones

Despliegue de MVs y aplicaciones web

Marvin Matías Agüero Torales

maguero@correo.ugr.es

Curso 2016-2017

Sumario

Enunciado de la práctica.....	3
Configuración.....	3
MV1 (con servidor web).....	3
MV2 (con SGBD).....	7
Aplicación web.....	9
Objetivo.....	9
Funcionalidad.....	9
Arquitectura software.....	10
Base de datos.....	10
Tablas.....	10
Manuales.....	10
Despliegue de las MVs.....	10
MV1.....	10
MV2.....	10
Uso de la aplicación web.....	10
Opcionales.....	13
Proceso de instalación del S.O. (desde una .ISO) en la MV.....	13
Proceso de instalación, configuración y despliegue de Owncloud.....	14
Bibliografía consultada.....	14
Principal.....	14
Para MV1.....	14
Para MV2.....	14
Anexos.....	15

Enunciado de la práctica

Dentro de la plataforma de prácticas habilitada para la asignatura, **OpenNebula**, accesible a través de docker.ugr.es (vía ssh), cada alumno deberá:

1. Crear dos MVs, cada una con una distribución de Linux (p.ej. CentOS, Fedora o Ubuntu).
 - a. En la primera MV instalará y configurará un servidor web (p.ej. Apache, nginx, o cualquier otro que el alumno decida).
 - b. En la segunda instalará y configurará un SGBD (p. ej. MySQL, PostgreSQL, etc.).
2. Desarrollar una aplicación web sencilla alojada en la MV1, que use una base de datos manejada por el SGBD instalado en la MV2. La aplicación web debe incluir el uso de formularios y la consulta y modificación de datos almacenados en la BD.
3. Realizar el despliegue de ambas Mvs, para evaluar el funcionamiento de la aplicación.
4. Elaborar un breve documento detallando el trabajo realizado.

Opcionales

5. Crear MVs con S.O. distintos de los disponibles en el Marketplace (p. ej. Fedora o Ubuntu).
6. Crear una MV para desplegar (<https://www.owncloud.org>).

Configuración

Como tuve errores para hacer funcionar máquinas de VirtualBox sobre OpenNebula (ver apartado Opcionales, Proceso de instalación del S.O. (desde una .ISO) en la MV), ambas MVs se montaron con imágenes disponibles en la plataforma.

MV1 (con servidor web)

Para la MV1, se selecciona la imagen de CentOS-6.5

```
ssh mcc4423998@docker.ugr.es
```

Creamos nuestro template en nuestra red

```
onemplate create --name "P_ServicioWeb" --cpu 1 --vcpu 1 --memory 1024 --arch x86_64  
--disk 8 --nic 216 --vnc --ssh --net_context
```

Creamos nuestra MV a partir de nuestro template

```
onemplate instantiate 623  
onevm show 761
```

Ingresamos a la MV creada

```
ssh root@192.168.10.61
```

La actualizamos, con esto hace upgrade a CentOS 6.8

```
yum -y update
```

Solucionamos una pequeña excepción del fastestmirror [1]

```
vi /etc/yum/pluginconf.d/fastestmirror.conf
```

```
#Change enabled=1 to enabled=0 to disable the fastestmirror plugin
```

Volvemos a actualizar el repositorio y

```
reboot
```

Empezamos a trabajar para configurar la aplicación web [2],[3] y [4]; primero copiamos el directorio de la aplicación en nuestro entorno

```
ssh mcc4423998@docker.ugr.es
```

```
mkdir Perl-Dancer_Simple-CRUD
```

```
exit
```

```
sftp mcc4423998@docker.ugr.es
```

```
put -r Perl-Dancer_Simple-CRUD
```

```
exit
```

Ingresamos y lo copiamos en el servidor web

```
ssh mcc4423998@docker.ugr.es
```

```
ssh root@192.168.10.61
```

```
mkdir Perl-Dancer_Simple-CRUD
```

```
sftp root@192.168.10.61
```

```
put -r Perl-Dancer_Simple-CRUD
```

```
exit
```

Intentamos levantar la aplicación, pero debemos instalar lo necesario (ver el aviso de error)

```
./simplecrud-example.pl --port 80
```

Control + C, e instalamos lo necesario, perl, PerlDancer y CPAN para los módulos necesarios que no existen en repositorios con yum

```
yum install perl
```

```
yum install cpan
```

```
yum install "perl(DBD::mysql)"
```

```
curl -L http://cpanmin.us | perl - --sudo Dancer
```

```
perl -MCPAN -e shell
```

```
install Dancer::Plugin::SimpleCRUD
```

```
quit
```

Como tuvimos algunos errores de dependencia, las anotamos e instalamos una a una

```
sudo yum install wget
```

```
wget http://search.cpan.org/CPAN/authors/id/B/BI/BIGPRESH/Dancer-1.3202.tar.gz
```

```
tar -xzf Dancer-1.3202.tar.gz
```

```
cd Dancer-1.3202
```

```
perl Makefile.PL
```

```
make
```

```
make test
```

```
make install
```

Vemos errores de dependencia aún, las instalamos [5]

```
cpanm Dancer Starman Daemon::Control
```

```
perl -MCPAN -e
```

```
get Dancer Hash::Merge::Simple HTTP::Body
```

```
make Dancer Hash::Merge::Simple HTTP::Body
```

Luego seguimos en CPAN, y una por una hacemos lo necesario, pero una a la vez

```
test Dancer ...
```

```
install Dancer ...
```

```
exit
```

```
cpanm Dancer::Plugin::SimpleCRUD::ParamsObject Dancer::Plugin::Auth::Extensible  
HTTP::Server::Simple::PSGI
```

```
perl -MCPAN -e'install Text::CSV'
```

Ya tenemos todo lo necesario, verificamos si levanta la aplicación

```
./simplecrud-example.pl --port 80
```

Ahora instalamos el Apache Server para poder rutear sobre el mismo, en el puerto 80, nuestra aplicación

```
sudo yum install httpd
```

```
chkconfig httpd on
```

```
service httpd start
```

Nmap para verificar los puertos

```
yum install nmap
nmap localhost -p 80
nmap 192.168.10.61 -p 80
```

Nuestro puerto está escuchando pero faltan dar los permisos, antes instalamos lo necesario en Apache

```
yum install gcc make kernel-devel perl
yum groupinstall "Development tools"
yum groupinstall "Web Server"
yum install httpd-devel
service httpd reload
```

Ponemos en permissive SELinux, para poder publicar la web

```
vim /etc/sysconfig/selinux
#set permissive
reboot
```

Ahora abrimos el firewall, es importante agregar la línea en el lugar indicado del archivo

```
vi /etc/sysconfig/iptables
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
#add in correct line group INPUT
service iptables restart
```

Ya tenemos abierto el puerto, instalamos lo necesario para hacer el proxy reverso [6]

```
vi /etc/httpd/conf/httpd.conf
#press / and search mod_proxy
/mod_proxy
#already exists, search and add VirtualHost
/VirtualHost
<VirtualHost *:80>
    ServerName simplecrud.com
    ServerAlias www.simplecrud.com
    DocumentRoot /root/Perl-Dancer_Simple-CRUD
    <Proxy *>
        Order deny,allow
```

```
    Allow from allow

</Proxy>

ProxyPass      / http://docker.ugr.es:15061/
ProxyPassReverse / http://docker.ugr.es:15061/

</VirtualHost>
```

Salimos y reiniciamos el servicio httpd

```
service httpd restart
```

Cambiamos el nombre de nuestro servidor de localhost a docker.ugr.es [7]

```
vi /etc/sysconfig/network

#set "HOSTNAME=myserver.domain.com"

vi /etc/hosts

#127.0.0.1    localhost localhost.localdomain myserver.domain.com

hostname hostname.domain.com

hostname

/etc/init.d/network restart
```

Probamos la aplicación

```
./simplecrud-example.pl --port 15061
```

Va todo bien, ahora instalamos el cliente para probar la conexión a la BD (antes debemos tener instalado todo lo necesario en el otro servidor)

```
yum install mysql

mysql -u webu -h 192.168.10.60 -p
```

De esta manera tenemos configurado el servidor listo para el despliegue.

MV2 (con SGBD)

Para la MV2 se elige la imagen de Ubuntu-14.04, ingresamos y creamos un template

```
ssh mcc4423998@docker.ugr.es

onetemplate create --name "P_ServicioSGBD" --cpu 1 --vcpu 1 --memory 1024 --arch
x86_64 --disk 10 --nic 216 --vnc --ssh --net_context
```

Iniciamos una MV a partir del template creado

```
onetemplate instantiate 627
```

Ingresamos a la MV y la ponemos a punto

```
ssh root@192.168.10.60
```

```
apt-get update -y
```

```
reboot
```

La MV está corriendo, ingresamos e instalamos MySQL

```
onevm list
```

```
ssh root@192.168.10.60
```

```
apt-get update -y
```

```
apt-get upgrade -y
```

```
apt-get install mysql-server
```

```
apt-get purge mysql-server
```

Errores en el repositorio de Ubuntu no nos permiten instalar MySQL, forzamos la limpieza

```
apt-get autoremove
```

```
apt-get autoclean
```

```
reboot
```

Ahora actualizamos la versión del S.O. a Ubuntu 16.04.2 LTS

```
do-release-upgrade
```

Agregamos al repo MySQL, luego probamos instalar de vuelta [8]

```
wget http://dev.mysql.com/get/mysql-apt-config_0.6.0-1_all.deb
```

```
sudo dpkg -i mysql-apt-config_0.6.0-1_all.deb
```

```
sudo apt-get update
```

```
sudo apt-get install mysql-server
```

```
mysql_secure_installation
```

```
mysql --version
```

```
service mysql status
```

```
mysqladmin -p -u root version
```

Ahora debemos habilitar acceso remoto a nuestra BD [9]

```
vi /etc/mysql/my.cnf
```

```
#[mysqld]
```

```
#bind-address = 192.168.1.60
```

```
# #skip-networking
```

```
quit
```

```
/etc/init.d/mysql restart
```



```
mysql -u root -p
GRANT ALL ON base1.* TO 'webu'@'192.168.10.61' IDENTIFIED BY 'webu123';
quit

ifconfig

iptables -A INPUT -i eth0 -s 192.168.10.61 -p tcp --destination-port 3306 -j ACCEPT
iptables -A INPUT -i eth0 -s 192.168.10.60 -p tcp --destination-port 3306 -j ACCEPT
iptables-save
```

Finalmente instauramos el esquema de persistencia [10]

```
touch schema.sql

sftp mcc4423998@docker.ugr.es
put -r schema.sql
exit

ssh mcc4423998@docker.ugr.es
sftp root@192.168.10.60
put -r schema.sql
exit

ssh root@192.168.10.60
mysql -u root -p base1 < schema.sql
```

Ya tenemos la BD instalada y corriendo.

Aplicación web

Objetivo

El objetivo de la aplicación web es de un administrador de usuarios, dar de alta usuarios, editarlos o borrarlos, vincularlos a una empresa y poder consultarlos, escribiendo notas sobre los mismos. Es un CRUD muy simple accesible desde cualquier dispositivo con conexión a Internet.

Funcionalidad

Las funcionalidades de esta aplicación son las esperadas de un CRUD: Crear, Leer, Actualizar y Borrar: en este contexto de personas o usuarios. Tiene una pantalla de login simple (si se quiere probar, en el archivo config.yml de la aplicación se pueden ver las credenciales) que ingresa a un listado de usuarios dados de alta, a la derecha de la tabla se pueden editarlos o borrarlos, también esta pantalla permite filtrar los datos para todas las columnas, se puede ordenar clicando las cabeceras de columnas; en le menú se puede cerrar sesión o agregar un nuevo usuario, volver a home.

Arquitectura software

La aplicación es un CRUD simple realizado sobre Perl, el framework Dancer y el plugin SimpleCRUD. Utiliza MVC, algunos patrones de diseño y puede funcionar con una arquitectura cliente/servidor y microservicios (la MV1 despliega el servicio web y MV2 el de datos). Al usar Bootstrap 4 es *responsiva*, pudiéndose acceder desde cualquier plataforma que cuenta con un navegador e Internet.

Base de datos

El SGBD utilizado por la aplicación es MySQL, motor relacional usado en ambientes web ampliamente, en su versión 5.7. Se ha montado una base de datos con dos tablas relacionadas.

Tablas

Las tablas creadas son people y employer. La primera es una tabla con datos personales básicos: nombre, edad, sexo; correo, comentarios, y su rol de usuario, además de a que empresa pertenece. Justamente la segunda tabla describe esa relación.

Manuales

Despliegue de las MVs

MV1

Para el despliegue es necesario tener configurado el servidor, ejecutar lo siguiente y la aplicación ya estará arriba en el puerto 80

```
./simplecrud-example.pl --port 15061 --daemon
```

```
service httpd restart
```

Hacemos un snapshot de nuestra VM

```
onevm snapshot-create ID
```

MV2

Podemos verificar que este corriendo

```
service mysql status
```

O hacer restart

```
service mysql restart
```

Antes debimos crear la BD con sus tablas mediante un script

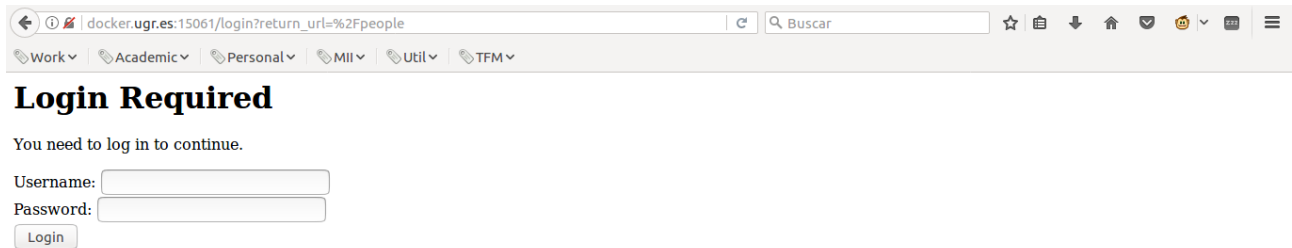
```
mysql -u root -p base1 < schema.sql
```

Hacemos un snapshot de la MV

```
onevm snapshot-create ID
```

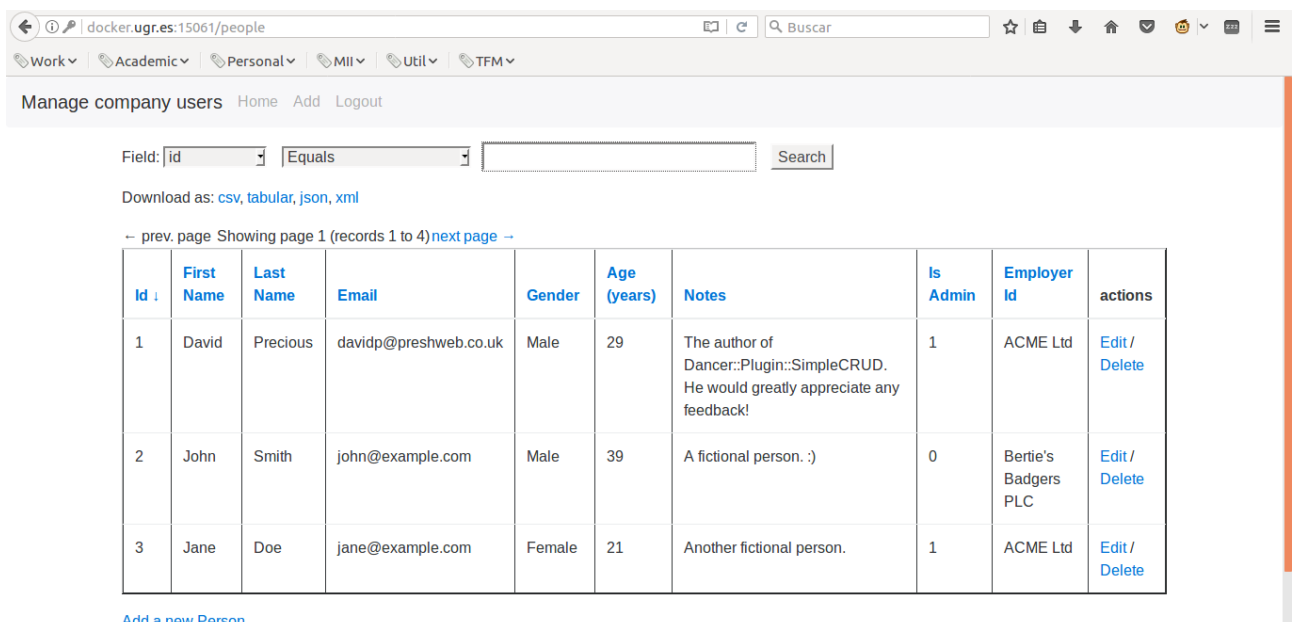
Uso de la aplicación web

Ir a la página de login en <http://docker.ugr.es:15061>, se pueden usar las credenciales editor@editor para ingresar.



The screenshot shows the login page of the application. The browser address bar shows the URL `docker.ugr.es:15061/login?return_url=%2Fpeople`. The page has a navigation bar with tabs: Work, Academic, Personal, MII, Util, and TFM. The main heading is "Login Required". Below it, a message says "You need to log in to continue." There are two input fields: "Username:" and "Password:". Below the password field is a "Login" button.

Al ingresar se observa el listado de usuarios con las distintas opciones habilitadas, como un dashboard simple de administración de usuarios.

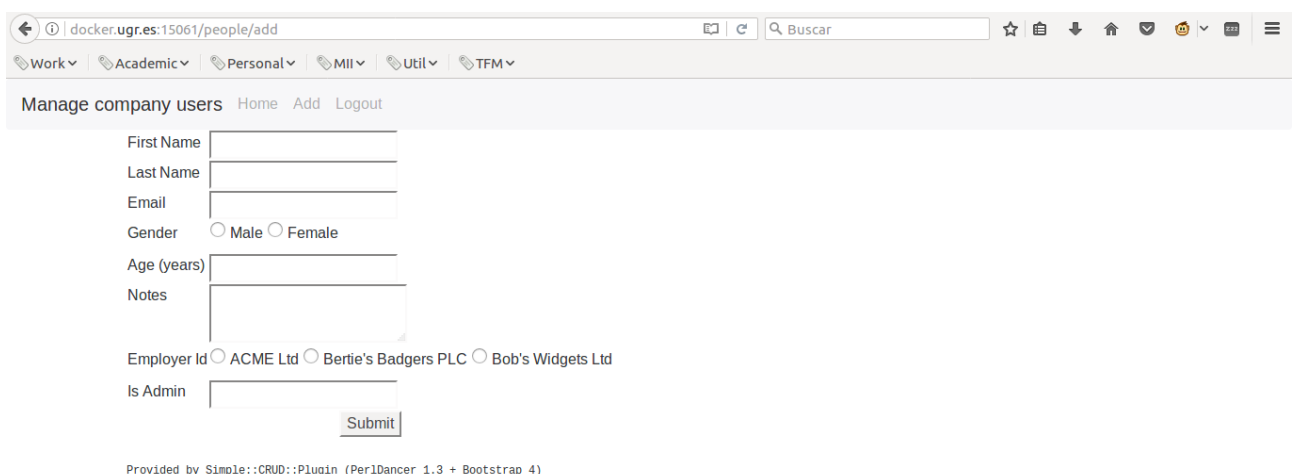


The screenshot shows the "Manage company users" dashboard. The browser address bar shows the URL `docker.ugr.es:15061/people`. The page has a navigation bar with tabs: Work, Academic, Personal, MII, Util, and TFM. The main heading is "Manage company users". Below it, there are links: Home, Add, and Logout. There is a search bar with a dropdown menu for "Field:" (set to "id") and a "Search" button. Below the search bar, there is a link "Download as: csv, tabular, json, xml". Below that, there is a link "prev. page" and a link "next page". The main content is a table with the following data:

Id	First Name	Last Name	Email	Gender	Age (years)	Notes	Is Admin	Employer Id	actions
1	David	Precious	davidp@preshweb.co.uk	Male	29	The author of Dancer::Plugin::SimpleCRUD. He would greatly appreciate any feedback!	1	ACME Ltd	Edit / Delete
2	John	Smith	john@example.com	Male	39	A fictional person. :)	0	Bertie's Badgers PLC	Edit / Delete
3	Jane	Doe	jane@example.com	Female	21	Another fictional person.	1	ACME Ltd	Edit / Delete

Below the table, there is a link "Add a new Person".

En el menú se puede agregar usuarios nuevos, volver a home o salir de la aplicación.



The screenshot shows the "Add new user" form. The browser address bar shows the URL `docker.ugr.es:15061/people/add`. The page has a navigation bar with tabs: Work, Academic, Personal, MII, Util, and TFM. The main heading is "Manage company users". Below it, there are links: Home, Add, and Logout. The form has the following fields: "First Name", "Last Name", "Email", "Gender" (radio buttons for Male and Female), "Age (years)", "Notes", "Employer Id" (radio buttons for ACME Ltd, Bertie's Badgers PLC, and Bob's Widgets Ltd), and "Is Admin". There is a "Submit" button at the bottom right of the form.

A la derecha de la tabla se puede editar o borrar usuarios existentes. Los títulos o cabeceras permiten ordenar los valores de cada columna.

docker.ugr.es:15061/people/edit/10

Work Academic Personal MII Util TFM

Manage company users

Home Add Logout

First Name: Michelle
Last Name: Doe
Email: michelle.doe@example.com
Gender: ☐ Male ☒ Female
Age (years): 29
Notes:
Employer Id: ☒ ACME Ltd ☐ Bertie's Badgers PLC ☐ Bob's Widgets Ltd
Is Admin: 0
Submit

Provided by Simple::CRUD::Plugin (PerlDancer 1.3 + Bootstrap 4)

docker.ugr.es:15061/people?o=email&d=asc&q=&searchfield=id&searchtype=e&p=3

Work Academic Personal MII Util TFM

Manage company users

Home Add Logout

Field: id Equals Search

Download as: csv, tabular, json, xml

prev. page Showing page 4 (records 10 to 13) next page

Id	First Name	Last Name	Email	Gender	Age (years)	Notes	Is Admin	Employer Id	actions
10	Michelle	Doe	michelle.doe@example.com	Female	29		0	ACME Ltd	Edit / Delete
8	Sophie	Doe	sophie.doe@example.com	Female	21		0	ACME Ltd	Edit / Delete
4	Test	User	test.user@example.com	Male	18	A test user.	0	Bob's Widgets Ltd	Edit / Delete

Add a new Person

Provided by Simple::CRUD::Plugin (PerlDancer 1.3 + Bootstrap 4)

El listado también puede filtrarse seleccionando la columna y el tipo de filtro.

docker.ugr.es:15061/people?searchfield=id&searchtype=e&q=10&o=email&d=asc&searchsubmit

Work Academic Personal MII Util TFM

Manage company users

Home Add Logout

Field: id Equals 10 Search

Showing results from searching for "id" equals "10" — Reset search

Download as: csv, tabular, json, xml

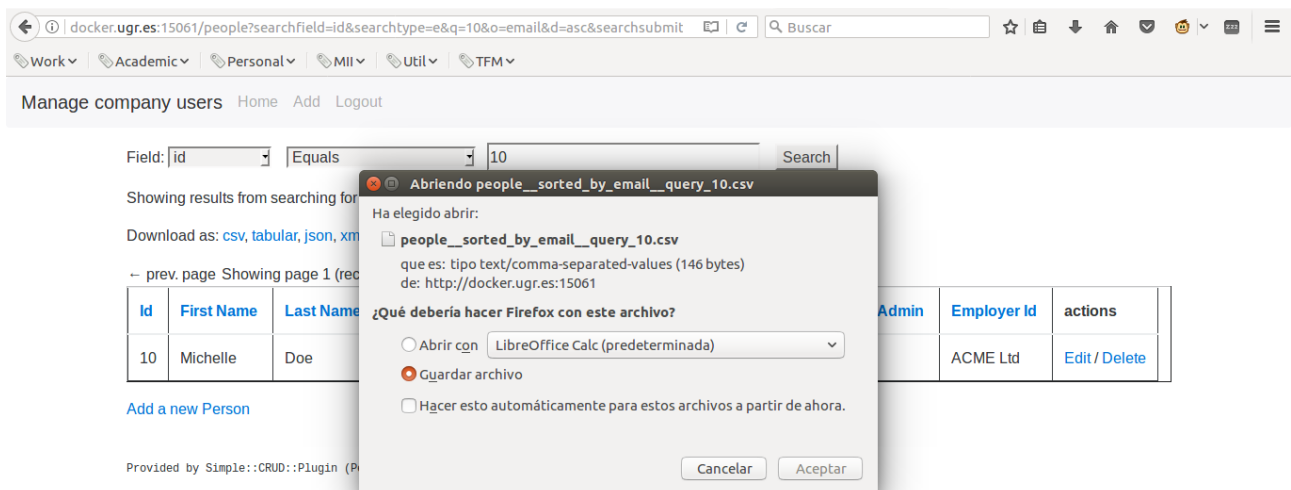
prev. page Showing page 4 (records 10 to 13) next page

Id	First Name	Last Name	Email	Gender	Age (years)	Notes	Is Admin	Employer Id	actions
10	Michelle	Doe	michelle.doe@example.com	Female	29		0	ACME Ltd	Edit / Delete

Add a new Person

Provided by Simple::CRUD::Plugin (PerlDancer 1.3 + Bootstrap 4)

Además es posible descargar el listado actual en CSV, txt o JSON.



Opcionales

Proceso de instalación del S.O. (desde una .ISO) en la MV

Como tenía montado un servidor Fedora 15 con MySQL 5.5 corriendo y escuchando en el puerto 3306, quise subirlo a OpenNebula, hice los pasos que venían en el guión, incluso me rebusque en la documentación oficial, pero siempre me arrojaba el error de Net::ReadTimeout en el último paso. De todas formas voy a describir los pasos que seguí, hasta el último, donde da un fallo.

Siguiendo los pasos de la instalación mixta con VirtualBox y la contextualización de OpenNebula para MV, primero no tuve que descargar la ISO puesto que ya disponía de una funcionando en VirtualBox con RAM de 1 GB, HD de 15 GB y 1 Core. Entonces pase a realizar la contextualización, descargué e instalé el paquete:

```
wget http://dev.opennebula.org/attachments/download/846/one-context\_4.10.0.rpm
```

```
rpm -i one-context_4.10.0.rpm
```

Se convierte la imagen de disco a OpenNebula

```
VBoxManage clonehd --format RAW "Fedora 15.vdi" FedoraMySQL.img
```

Pasamos a zip

```
zip FedoraMySQL.zip FedoraMySQL.img
```

Lo copiamos

```
scp FedoraMySQL.zip mcc4423998@docker.ugr.es:/home/mcc4423998/
```

Ingresamos y descomprimos

```
unzip FedoraMySQL.zip
```

Pasamos a QCOW2

```
qemu-img convert -f raw FedoraMySQL.img -O qcow2 FedoraMySQL.qcow2
```

Luego la montamos como imagen para poder acceder desde un template

```
oneimage create -d default --name fedoramysql --path  
/home/mcc4423998/FedoraMySQL.qcow2 --type OS --driver qcow2 --description  
"MiFedoraMySQL"
```

y/o

```
oneimage create -d 1 --name fedoramysql --path /home/mcc4423998/FedoraMySQL.qcow2  
--prefix sd --type OS --driver qcow2 --description "MiFedoraMySQL"
```

y/o

```
oneimage create -d default --name fedoramysql --path  
/home/mcc4423998/FedoraMySQL.img
```

Devolviendo siempre el error Net::ReadTimeout.

Proceso de instalación, configuración y despliegue de Owncloud

Se obvia esta petición extra, puesto que en clase el Prof. de prácticas mencionó que lo haríamos en los siguientes guiones de prácticas.

Bibliografía consultada

Principal

Guión de la Práctica 1 de la asignatura Cloud Computing: Servicios y Aplicaciones.

Para MV1

- [1] <http://www.webhostingtalk.com/showthread.php?t=1049988>
- [2] <http://advent.perldancer.org/2011/2>
- [3] http://www.quackit.com/bootstrap/bootstrap_4/tutorial/bootstrap_navbars.cfm
- [4] <https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server>
- [5] <https://perlmaven.com/getting-started-with-perl-dancer-on-digital-ocean#aptitude>
- [6] <http://advent.perldancer.org/2010/14>
- [7] <https://support.rackspace.com/how-to/centos-hostname-change/>

Para MV2

- [8] <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-14-04>
- [9] <https://www.linuxito.com/gnu-linux/nivel-medio/170-como-habilitar-el-acceso-remoto-a-un-servidor-de-bases-de-datos-mysql>

[10] <http://joshualande.com/create-tables-sql>

Anexos

- /appweb
 - /web
 - incluye las vistas y layouts, configuraciones de aplicación como conexión a BD, usuarios, etc., archivo perl que ejecuta la aplicación.
 - /sgbd
 - incluye la creación de la base de datos, tablas e inserciones.

Tambien disponible en <https://github.com/mmaguero>