

# Informe de Laboratorio 01

Nota

Estudiante	Escuela	Asignatura
Quispe Arratea Alexandra Raquel Cari Lipe Paul Andree	Escuela Profesional de Ingeniería de Sistemas	Pweb - II Semestre: I

Laboratorio	Tema	Duración
01		24 horas aprox.

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 30 Abril 2024	Al 4 Mayo 2024

## 1. Tarea

- Crear un contenedor en Docker basado en Ubuntu 20.04:
  - Especificaciones del Lab 01
    - Instale el servidor web Apache HTTP server 2.x
    - Instale cualquiera de estos lenguajes de programación: PHP, Perl, Python.
    - Configure el servidor web para que interprete uno de los lenguajes de programación.
    - Instale cualquiera de los servidores de base de datos: MySQL, MariaDB, PostgreSQL.
    - Instale el servidor Open SSH Server. Envíe archivos al servidor: imágenes, CSS, JS, etc.
    - Cree un usuario **pw2** con contraseña: **12345678**.
    - Otorgue permisos al usuario para acceder a la aplicación web. (Read/Write)
    - Finalmente, implemente el trabajo final del curso de **pw1** en ese contenedor.
    - Elabore un informe paso a paso donde explique funcionalmente el proyecto, demostrando que se trata de un contenedor Docker.
    - Adjunte la URL de un video donde muestre que se trata de un contenedor Docker.

## 2. Equipos, materiales y temas utilizados

- Sistemas Operativos Ubuntu y Windows
- NVIM v0.6.1, VSCode, Eclipse
- OpenJDK 64-Bits 11, 17, 18
- Git 2.34.1 y versiones posteriores
- Cuentas en GitHub/GitLab con el correo institucional.

### 3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- [https://github.com/aquispearr/pweb2\\_lab01](https://github.com/aquispearr/pweb2_lab01)
- URL del repositorio GitLab para clonar o recuperar.
- [https://gitlab.com/aquispearr/pweb2\\_lab01](https://gitlab.com/aquispearr/pweb2_lab01)
- URL del video explicativo.
- <https://docs.google.com/document/d/11xwpGD-ZwGulgnNTZ7G955Ih7jFJAvSSwohtfZHDsbQ/edit?hl=es>

### 4. Desarrollo

#### 4.1. Docker pull

Paso 1: Descargar la imagen de Ubuntu

En este paso, descargamos la imagen de Ubuntu desde el repositorio de Docker Hub utilizando el comando `docker pull`.

```
C:\Users\USER>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
49b384cc7b4a: Pull complete
Digest: sha256:3f85b7caad41a95462cf5b787d8a04604c8262cdcdf9a472b8c52ef83375fe15
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu

C:\Users\USER>
```

Figura 1: Proceso de descarga de la imagen de Ubuntu

En la captura de pantalla proporcionada, se puede observar el proceso de descarga de la imagen de Ubuntu. A continuación, se detallan los pasos realizados:

Es importante destacar que la imagen de Ubuntu descargada servirá como base para la creación de nuestro contenedor Docker, sobre el cual se llevará a cabo la instalación y configuración de los servicios requeridos para el laboratorio.

#### 4.2. Docker run

En este paso, creamos un contenedor Docker a partir de la imagen de Ubuntu descargada anteriormente. Además, configuramos los puertos necesarios para la conexión a la base de datos y otros servicios requeridos según las especificaciones del laboratorio.

```
C:\Users\USER>docker run --name pw2_lab1 -p 8084:80 -p 8085:3306 -p 8086:22 -it ubuntu:20.04 /bin/bash
root@1ba4c350e166:/#
```

Figura 2: Creación del contenedor Docker

En la captura de pantalla proporcionada, se muestra el comando `docker run` utilizado para crear el contenedor Docker. A continuación, se detallan los parámetros utilizados y su función:

- **Nombre del contenedor (--name pw2\_lab1):**

```
C:\Users\USER>docker run --name pw2_lab1 -p 8084:80 -p 8085:3306 -p 8086:22 -it
ubuntu:20.04 /bin/bash
```

Especificamos el nombre del contenedor como "pw2\_lab1" utilizando la opción `--name`. Este nombre será utilizado para hacer referencia al contenedor en operaciones posteriores.

- **Mapeo de puertos (-p 8084:80 -p 8085:3306 -p 8086:22):**

Utilizamos la opción `-p` para mapear los puertos del contenedor a los puertos del host. En este caso, hemos mapeado los siguientes puertos:

- Puerto 80 del contenedor (utilizado por el servidor web) al puerto 8084 del host.
- Puerto 3306 del contenedor (utilizado por la base de datos) al puerto 8085 del host.
- Puerto 22 del contenedor (utilizado por SSH) al puerto 8086 del host.

- **Interactividad y acceso a la terminal (-it):**

Utilizamos las opciones `-it` para mantener el contenedor en modo interactivo y acceder a su terminal.

- **Imagen base (ubuntu:20.04):**

Especificamos la imagen base a partir de la cual se creará el contenedor, que en este caso es Ubuntu 20.04.

- **Shell (/bin/bash):**

Indicamos el shell que queremos utilizar una vez que el contenedor esté en funcionamiento, en este caso, Bash.

### 4.3. Actualización del Sistema Operativo

En este paso, realizamos una actualización de los repositorios del sistema operativo Ubuntu utilizando el comando `apt-get update`.

```
root@1ba4c350e166:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [29.8 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [1201 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3595 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [4069 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3686 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [32.5 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1497 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [28.6 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [55.2 kB]
Get:18 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [3536 kB]
Fetched 31.2 MB in 7s (4274 kB/s)
Reading package lists... Done
root@1ba4c350e166:/#
```

Figura 3: Ejecución del comando `apt-get update`

En la captura de pantalla proporcionada, se muestra la ejecución del comando `apt-get update`. A continuación, se detalla la salida relevante del comando:

- **Descarga de actualizaciones de repositorios:**

El comando `apt-get update` realiza la descarga de información sobre los paquetes disponibles en los repositorios configurados en el sistema operativo.

```
root@1ba4c350e166: /# apt-get update
```

- **Progreso de actualización y lectura de listas de paquetes:**

Se observa el progreso de la descarga de la información de los repositorios y una vez completada la descarga, se muestra el mensaje `Reading package lists... Done`, indicando que el proceso ha finalizado satisfactoriamente.

Este paso es importante para asegurarnos de que nuestro sistema operativo tenga acceso a la información más reciente sobre los paquetes disponibles en los repositorios, lo cual es fundamental para la instalación de software adicional requerido para nuestro entorno de trabajo.

#### 4.4. Instalación del Editor de Texto Vim

En este paso, instalamos el editor de texto Vim dentro del contenedor Docker. Vim es un editor de texto altamente configurable, que facilita la edición de archivos de configuración y código fuente.

```
root@1ba4c350e166:/# apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf file libasound2 libasound2-data libcanberra0 libexpat1 libgpm2 libltdl7
  libmagic-mgc libmagic1 libmpdec2 libogg0 libpython3.8 libpython3.8-minimal libpython3.8-stdlib libreadline8
  libsqlite3-0 libssl1.1 libtdb1 libvorbis0a libvorbisfile3 mime-support readline-common sound-theme-freedesktop
  vim-common vim-runtime xxd xz-utils
Suggested packages:
  libasound2-plugins alsa-utils libcanberra-gtk0 libcanberra-pulse gpm readline-doc ctags vim-doc vim-scripts
The following NEW packages will be installed:
  alsa-topology-conf alsa-ucm-conf file libasound2 libasound2-data libcanberra0 libexpat1 libgpm2 libltdl7
  libmagic-mgc libmagic1 libmpdec2 libogg0 libpython3.8 libpython3.8-minimal libpython3.8-stdlib libreadline8
  libsqlite3-0 libssl1.1 libtdb1 libvorbis0a libvorbisfile3 mime-support readline-common sound-theme-freedesktop vim
  vim-common vim-runtime xxd xz-utils
0 upgraded, 30 newly installed, 0 to remove and 0 not upgraded.
Need to get 15.0 MB of archives.
After this operation, 70.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libmagic-mgc amd64 1:5.38-4 [218 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 libmagic1 amd64 1:5.38-4 [75.9 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal/main amd64 file amd64 1:5.38-4 [23.3 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libexpat1 amd64 2.2.9-1ubuntu0.6 [74.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/main amd64 libmpdec2 amd64 2.4.2-3 [81.1 kB]
```

Figura 4: Ejecución del comando `apt-get install vim`

En la captura de pantalla proporcionada, se muestra el comando `apt-get install vim` utilizado para instalar Vim dentro del contenedor Docker. A continuación, se detallan los pasos realizados y los detalles observados en la captura de pantalla:

- **Ejecución del comando `apt-get install vim`:**

Este comando se utiliza para iniciar el proceso de instalación del editor de texto Vim en el sistema operativo Ubuntu dentro del contenedor Docker.

- **Lectura de los paquetes a instalar:**

La terminal muestra una lista de los paquetes adicionales que serán instalados junto con Vim, así como la cantidad de espacio en disco que se utilizará.

- **Confirmación de la instalación:**

Se solicita confirmación para proceder con la instalación de los paquetes adicionales y la ocupación de espacio en disco.

- **Progreso de la instalación:**

La terminal muestra el progreso de la descarga e instalación de los paquetes necesarios para Vim.

Una vez completada la instalación, Vim estará disponible en el sistema y listo para su uso. Este editor de texto será útil para la edición de archivos de configuración, código fuente y cualquier otro tipo de archivo de texto necesario durante el desarrollo y configuración del entorno de trabajo en el contenedor Docker.

## 4.5. Instalación del Servidor MariaDB

En este paso, instalamos el servidor MariaDB dentro del contenedor Docker. MariaDB es un sistema de gestión de bases de datos relacional que proporciona un reemplazo compatible con MySQL.

```
root@1ba4c350e166: /# apt-get install mariadb-server
```

```
root@1ba4c350e166:/# apt-get install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-3 gawk iproute2 libaio1 libatm1 libbsd0 libcap2 libcap2-bin libcgi-fast-perl libcgi-pm-perl
  libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libelf1 libencode-locale-perl libfcgi-perl libgdbm-compat4
  libgdbm6 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libmnl0 libmpfr6 libmysqlclient21 libpam-cap libperl5.30 libpopt0
  libreadline5 libsigsegv2 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl libwrap0 libxtables12 lsof
  mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server-10.3 mariadb-server-core-10.3
  mysql-common netbase perl perl-modules-5.30 psmisc rsync socat
Suggested packages:
  gawk-doc iproute2-doc libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl gdbm-l10n
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx mariadb-test netcat-openbsd tinycat perl-doc
  libterm-readline-gnu-perl | libterm-readline-perl-perl make libb-debug-perl liblocale-codes-perl openssh-client
  openssh-server
The following NEW packages will be installed:
  galera-3 gawk iproute2 libaio1 libatm1 libbsd0 libcap2 libcap2-bin libcgi-fast-perl libcgi-pm-perl
  libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libelf1 libencode-locale-perl libfcgi-perl libgdbm-compat4
  libgdbm6 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libmnl0 libmpfr6 libmysqlclient21 libpam-cap libperl5.30 libpopt0
  libreadline5 libsigsegv2 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl libwrap0 libxtables12 lsof
  mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server mariadb-server-10.3
  mariadb-server-core-10.3 mysql-common netbase perl perl-modules-5.30 psmisc rsync socat
0 upgraded, 53 newly installed, 0 to remove and 0 not upgraded.
```

Figura 5: Ejecución del comando `apt-get install mariadb-server`

En la captura de pantalla proporcionada, se muestra el comando `apt-get install mariadb-server` utilizado para instalar el servidor MariaDB dentro del contenedor Docker.

## 5. Instalación del Servidor Apache

En este paso, instalamos el servidor Apache dentro del contenedor Docker. Apache es un servidor web ampliamente utilizado que permite alojar sitios web y servir contenido web estático y dinámico.

En la captura de pantalla proporcionada, se muestra el comando `apt-get install apache2` utilizado para instalar el servidor Apache dentro del contenedor Docker. A continuación, se detallan los pasos realizados y los detalles observados en la captura de pantalla:

- **Ejecución del comando `apt-get install apache2`:**

Este comando inicia el proceso de instalación del servidor Apache en el sistema operativo Ubuntu dentro del contenedor Docker.



```
root@693709a2395e:/# /etc/init.d/mysql start
* Starting MariaDB database server mysqld
root@693709a2395e:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.3.39-MariaDB-0ubuntu0.20.04.2 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> -- phpMyAdmin SQL Dump
0:14', 1, 1);
-----
```

```
MariaDB [bancafinal]> /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
Query OK, 0 rows affected (0.000 sec)

MariaDB [bancafinal]> /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
Query OK, 0 rows affected (0.000 sec)

MariaDB [bancafinal]>
MariaDB [bancafinal]> -- usuarios
MariaDB [bancafinal]> CREATE USER 'query'@'%' IDENTIFIED BY 'YR4AFJUC3nyRmasY';
Query OK, 0 rows affected (0.000 sec)

MariaDB [bancafinal]> GRANT SELECT, INSERT, DELETE ON `bancafinal`.* TO 'query'@'%;
Query OK, 0 rows affected (0.000 sec)

MariaDB [bancafinal]>
MariaDB [bancafinal]> INSERT INTO `operarios` (`id`, `nombre`, `usuario`, `clave`) VALUES (10, 'A', 'ale', '313');
Query OK, 1 row affected (0.002 sec)

MariaDB [bancafinal]> exit
Bye
```

```
root@1ba4c350e166:/# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils ca-certificates krb5-locales libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libasn1-8-heimdal libbrotli1 libcurl4 libgssapi-krb5-2 libgssapi3-heimdal libhcrypto4-heimdal
  libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal libicu66 libjansson4 libk5crypto3 libkeyutils1
  libkrb5-26-heimdal libkrb5-3 libkrb5support0 libldap-2.4-2 libldap-common liblua5.2-0 libnghttp2-14 libpsl5
  libroken18-heimdal librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libssh-4 libwind0-heimdal libxml2
  openssl publicsuffix ssl-cert tzdata
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser ufw krb5-doc krb5-user
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp
  libsasl2-modules-sql openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils ca-certificates krb5-locales libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libasn1-8-heimdal libbrotli1 libcurl4 libgssapi-krb5-2 libgssapi3-heimdal
  libhcrypto4-heimdal libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal libicu66 libjansson4 libk5crypto3
  libkeyutils1 libkrb5-26-heimdal libkrb5-3 libkrb5support0 libldap-2.4-2 libldap-common liblua5.2-0 libnghttp2-14
  libpsl5 libroken18-heimdal librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libssh-4 libwind0-heimdal
  libxml2 openssl publicsuffix ssl-cert tzdata
0 upgraded, 43 newly installed, 0 to remove and 0 not upgraded.
Need to get 14.8 MB of archives.
After this operation, 58.6 MB of additional disk space will be used.
```

Figura 6: Instalación del servidor Apache

- **Progreso de la instalación:**

La terminal muestra el progreso de la descarga e instalación de los paquetes necesarios para Apache.

- **Confirmación de la instalación:**

Se muestra un mensaje solicitando confirmación para proceder con la instalación de los paquetes adicionales y la ocupación de espacio en disco.

- **Finalización de la instalación:**

Una vez completada la instalación, Apache estará disponible en el sistema y listo para servir contenido web.

```
root@693709a2395e:/# /etc/init.d/apache2 start
* Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
*
```

Figura 7: Reinicio del servidor Apache

Este paso es fundamental para configurar un entorno de desarrollo web dentro del contenedor Docker, donde podremos alojar y probar nuestros sitios y aplicaciones web de manera local.

## 6. Comprobación de la Instalación del Servidor Apache

En esta sección, verificaremos la instalación del servidor Apache dentro del contenedor Docker. La siguiente imagen muestra la página de inicio predeterminada de Apache al acceder a localhost en el puerto 8084:

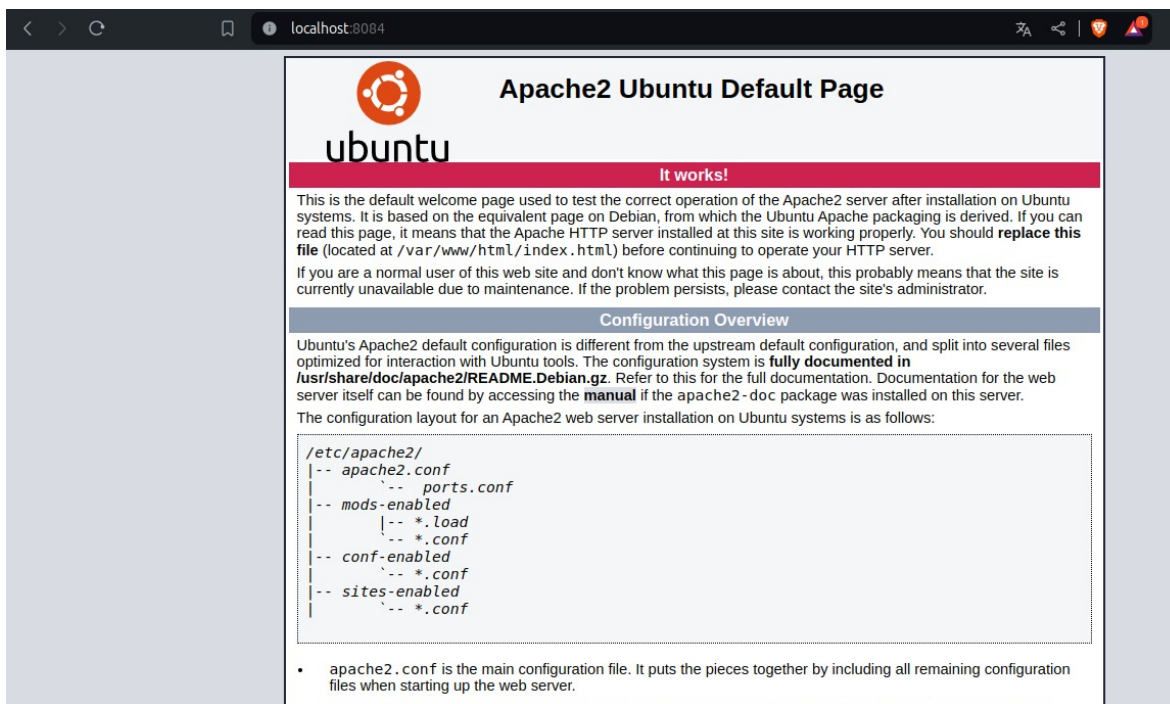


Figura 8: Página de inicio predeterminada de Apache



## 7. Modificación de carpetas/directorios

```
root@693709a2395e:/# cd ../../var/www/html/
root@693709a2395e:/var/www/html# git clone https://github.com/ynoacamino/bancaFinal.git
Cloning into 'bancaFinal'...
remote: Enumerating objects: 485, done.
remote: Counting objects: 100% (485/485), done.
remote: Compressing objects: 100% (310/310), done.
remote: Total 485 (delta 336), reused 316 (delta 167), pack-reused 0
Receiving objects: 100% (485/485), 3.19 MiB | 4.29 MiB/s, done.
Resolving deltas: 100% (336/336), done.
root@693709a2395e:/var/www/html# cd bancaFinal
root@693709a2395e:/var/www/html/bancaFinal# git switch pw2
Branch 'pw2' set up to track remote branch 'pw2' from 'origin'.
Switched to a new branch 'pw2'
```

Figura 9: Primera modificación de carpetas/directorios

```
root@693709a2395e:/var/www/html/bancaFinal# cp -r ./cgi-bin ../../usr/lib/
root@693709a2395e:/var/www/html/bancaFinal# cd ../../usr/lib/cgi-bin
root@693709a2395e:/usr/lib/cgi-bin# ls
eliminar_clientes.pl  eliminar_tarjetas.pl  lista_tarjetas.pl  movimiento_cliente.pl  registro_cuentas.pl  sesion.pl
eliminar_cuentas.pl  estado_tarjeta.pl    login.pl           registro_clientes.pl   registro_tarjetas.pl
root@693709a2395e:/usr/lib/cgi-bin# chmod 755 *.pl
root@693709a2395e:/usr/lib/cgi-bin# ls
eliminar_clientes.pl  eliminar_tarjetas.pl  lista_tarjetas.pl  movimiento_cliente.pl  registro_cuentas.pl  sesion.pl
eliminar_cuentas.pl  estado_tarjeta.pl    login.pl           registro_clientes.pl   registro_tarjetas.pl
root@693709a2395e:/usr/lib/cgi-bin# /etc/init.d/apache2 restart
* Restarting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[ OK ]
root@693709a2395e:/usr/lib/cgi-bin#
```

Figura 10: Segunda modificación de carpetas/directorios

## 8. Funcionamiento final de la Pagina Web

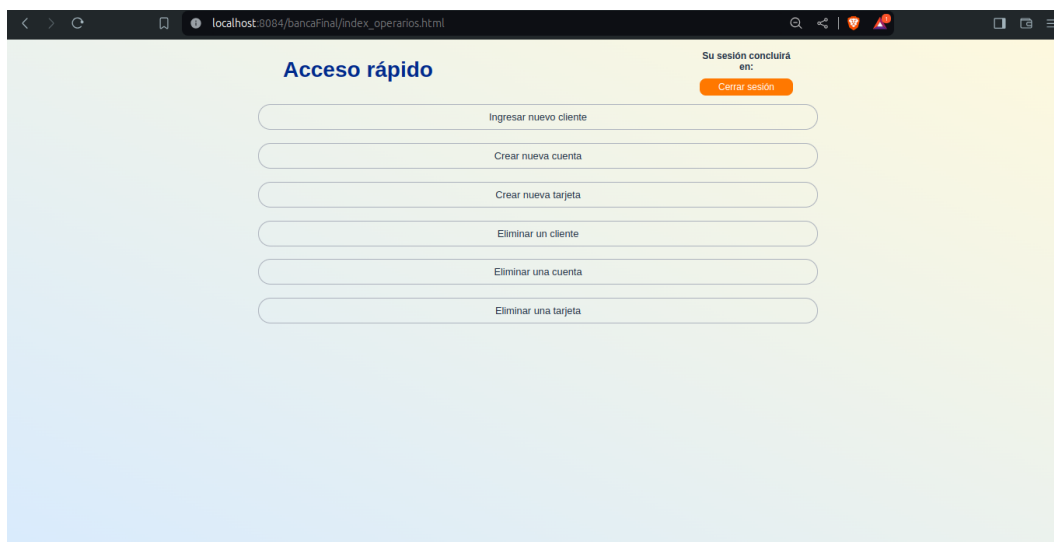


Figura 11: Resultado final 1

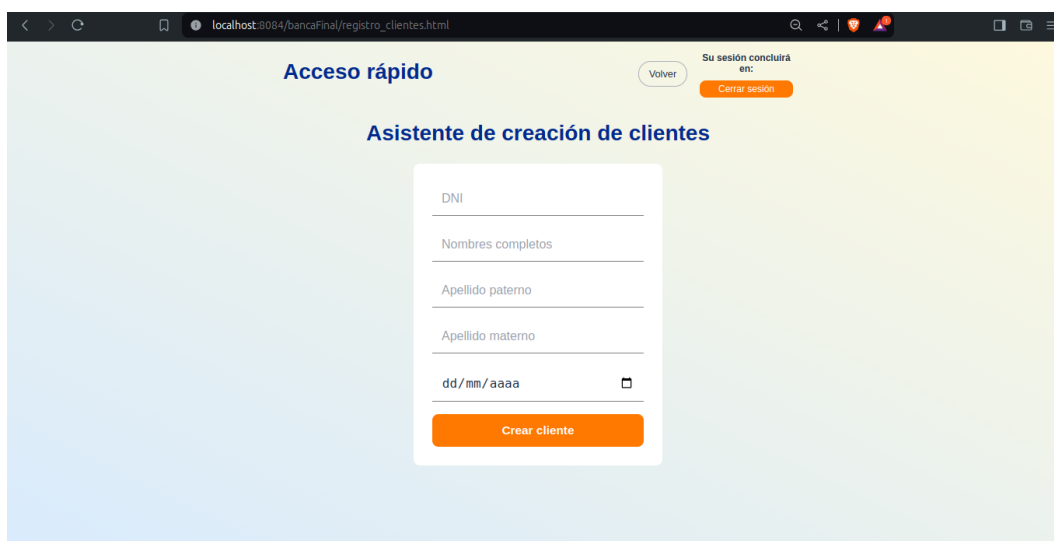


Figura 12: Resultado final 2

## 9. Rúbricas

### 9.1. Entregable Informe

Tabla 1: Tipo de Informe

<b>Informe</b>	
<b>Latex</b>	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

### 9.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

<b>Puntos</b>	<b>Nivel</b>			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
<b>2.0</b>	0.5	1.0	1.5	2.0
<b>4.0</b>	1.0	2.0	3.0	4.0