



Universidad de las Américas

Administración de Bases de Datos

Informe de Proyecto Integrador

Quito, Ecuador

04/07/2023

Nicolas Camacho, Felipe Novillo, Carlos Ayo

INTRODUCCIÓN

Se dice que las bases de datos envueltas en contenedores son bases de datos contenerizadas. Los contenedores son pequeños paquetes de software independientes que incluyen todos los componentes necesarios para ejecutar un programa, incluido el entorno de tiempo de ejecución, las bibliotecas y el código de la aplicación. Por lo tanto, pueden operar en cualquier infraestructura que admita Docker, la tecnología de contenerización más utilizada, y son fáciles de implementar y administrar.

El uso de bases de datos en contenedores tiene una serie de ventajas. En primer lugar, son transportables y fáciles de trasladar de un entorno a otro. Debido a que las bases de datos se pueden ampliar o reducir según sea necesario en implementaciones basadas en la nube, son perfectas para ello. En segundo lugar, dado que están separadas entre sí y de la infraestructura subyacente, las bases de datos en contenedores son más seguras que las bases de datos convencionales. Como resultado, son menos susceptibles a las agresiones. En tercer lugar, dado que las bases de datos en contenedores solo utilizan los recursos que necesitan, son más eficaces que las bases de datos convencionales. Esto puede resultar en reducciones significativas de costos.

OBJETIVOS

El trabajo por realizar tiene como finalidad implementar una solución de base de datos contenerizada que permita cubrir los siguientes puntos:

- Plantear el despliegue de una solución de base de datos contenerizada
- Efectuar el despliegue de una solución de base de datos contenerizada
- Validar el despliegue de una solución de base de datos contenerizada
- Analizar los riesgos de seguridad a nivel privilegios que puede poseer una base de datos contenerizada
- Plantear una alternativa de solución a los riesgos de seguridad a nivel privilegios que puede poseer una base de datos contenerizada, basándose en un enfoque RBA
- Implementar una solución de seguridad basada en RBA para la operación segura de una base de datos contenerizada

DESARROLLO

```
root@fgnadockerssrv:~  
File Edit View Search Terminal Help  
4 packets transmitted, 4 received, 0% packet loss, time 3010ms  
rtt min/avg/max/mdev = 16.926/17.126/17.284/0.184 ms  
[student@fgnadockerssrv ~]$ su -  
Password:  
Last login: Tue May 2 12:04:17 -05 2023 on pts/0  
[root@fgnadockerssrv ~]# docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
<none> <none> 32cccfa340e6 8 weeks ago 379MB  
postgres latest ceccf204404e 2 months ago 379MB  
[root@fgnadockerssrv ~]# docker pull mysql:lates  
Error response from daemon: manifest for mysql:lates not found: manifest unknown: manifest unknown  
[root@fgnadockerssrv ~]# docker pull mysql:latest  
latest: Pulling from library/mysql  
46ef68baacb7: Pull complete  
94c1114b2e9c: Pull complete  
ff05e3f38802: Pull complete  
41cc3fcd9912: Pull complete  
07bbc8bdf52a: Pull complete  
6d88f83726a9: Pull complete  
cf5c7d5d33f7: Pull complete  
9db3175a2a66: Pull complete  
feaede27fa9: Pull complete  
cf91e7784414: Pull complete  
b1770db1c329: Pull complete  
Digest: sha256:15f069202c46cf861ce429423ae3f8dfa6423306fbf399eaf36094ce30dd75c  
Status: Downloaded newer image for mysql:latest  
docker.io/library/mysql:latest
```

El comando **docker pull mysql:latest** se utiliza en Docker para descargar la imagen más reciente (latest) del contenedor de MySQL desde el registro de Docker Hub.

```
[root@fgnadockerssrv ~]# mkdir -p /mysql/configuracion  
[root@fgnadockerssrv ~]# mkdir -p /mysql/database
```

Creación de volúmenes, El comando `mkdir -p /mysql/` se utiliza para crear un directorio dentro de la ruta `"/mysql"`, Esto es útil para organizar los archivos de configuración de MySQL u otros archivos relacionados en una ubicación específica dentro del sistema de archivos.

```
[root@fgnadockerssrv ~]# docker run -p 33060:3306 --name cont_proyect -v /mysql/database:/var/lib/mysql -v /mysql/configuracion:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=Proyecto123 -d mysql
```

Arranque del contenedor y asociación de volúmenes con los directorios de arc de conf y directorios de db

```
[root@fgnadockerssrv ~]# docker exec -it cont_proyect bash
bash-4.4# cd /etc
bash-4.4# ls
GREP_COLORS          group                localtime            pm                   selinux
X11                  group-              login.defs           popt.d              services
aliases              gshadow             mtab                 printcap             shadow
alternatives         gshadow-            mtab                 profile              shadow-
bash_completion.d    gss                 my.cnf               profile.d            shells
bashrc               host.conf            my.cnf.d             protocols            skel
bindresvport.blacklist hostname             mysql                rc.d                 ssl
chkconfig.d          hosts               netconfig            rc0.d                subgid
crypto-policies       init.d              networks             rc1.d                subuid
csh.cshrc             inputrc             nsswitch.conf        rc2.d                swid
csh.login             issue              nsswitch.conf.bak    rc3.d                sysconfig
default              issue.net           openldap              rc4.d                system-release
dnf                   krb5.conf           opt                  rc5.d                system-release-cpe
environment           krb5.conf.d         oracle-release        rc6.d                terminfo
ethertypes            ld.so.cache         os-release            redhat-release       xattr.conf
exports              ld.so.conf          passwd                resolv.conf          xdg
filesystems          ld.so.conf.d        passwd-               rpc                  xinetd.d
gcrypt                libaudit.conf       pkcs11                rpm                  yum.repos.d
gnupg                 libssh              pki                   sasl2
bash-4.4# cp my.cnf /etc/mysql/conf.d/my.cnf
bash-4.4# exit
exit
```

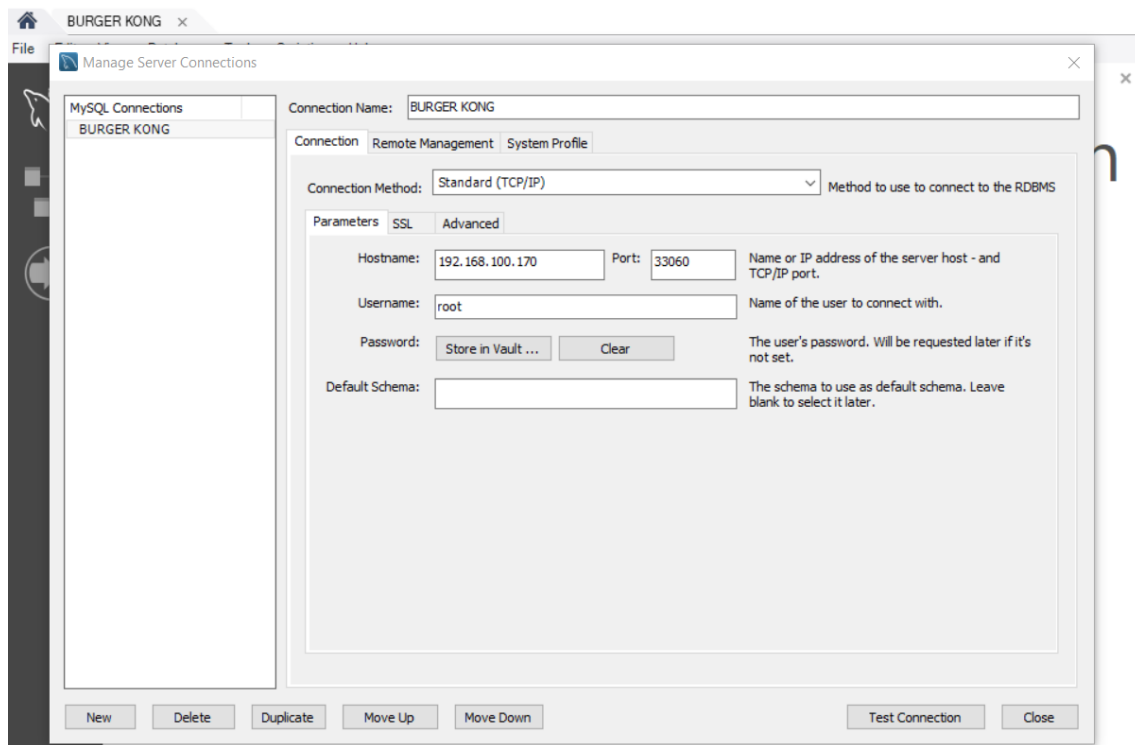
Copia del archivo de configuración para persistir la configuración de la base de datos.

```
root@fgnadockerssrv:/mysql/configuracion
File Edit View Search Terminal Help
[root@fgnadockerssrv configuracion]# cd /mysql/configuracion
[root@fgnadockerssrv configuracion]# nano my.cnf
[root@fgnadockerssrv configuracion]# cat my.cnf | grep port
# Remove leading # and set to the amount of RAM for the most important data
# Remove leading # to turn on a very important data integrity option: logging
# Remove leading # to set options mainly useful for reporting servers.
port=50
[root@fgnadockerssrv configuracion]#
```

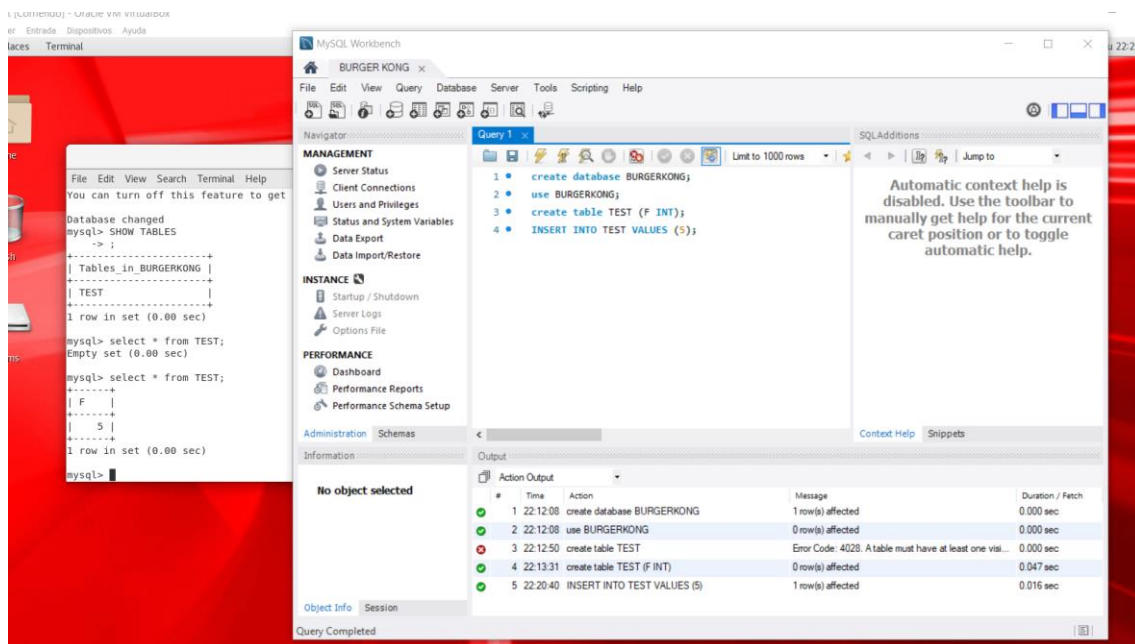
Cambio del puerto por default al puerto 55.

```
[root@fgnadockerssrv configuracion]# docker run -p 33060:50 --name cont_proyect -v /mysql/configuracion:/etc/mysql/conf.d -v /mysql/database:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=Proyecto123 -d mysql
ebf6cdeb75fcab1cc079dd1d790cf4cc550c6e081f5f635d36b940af9807a8e5
[root@fgnadockerssrv configuracion]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS      PORTS                               NAME
ebf6cdeb75fc   mysql    "docker-entrypoint.s..." 4 seconds ago Up 4 seconds 3306/tcp, 33060/tcp, 0.0.0.0:33060->50/tcp cont_proyect
[root@fgnadockerssrv configuracion]#
```

Subida de contenedor cambiando el puerto por default al puerto 33060



Se realiza la conexión desde cliente Workbench hacia la base de datos.



Para probar realizamos desde la interfaz de usuario de mysql workench una consulta y una inserción de datos a una tabla.

```
student@fgnadockerssrv:~/faker_BURGERKONG
File Edit View Search Terminal Help

Dload Upload Total Spent Left Speed
100 2518k 100 2518k 0 0 2298k 0 0:00:01 0:00:01 --:--:-- 2301k
(env) [student@fgnadockerssrv faker_BURGERKONG]$ python -m venv env
(env) [student@fgnadockerssrv faker_BURGERKONG]$ source env/bin/activate
(env) [student@fgnadockerssrv faker_BURGERKONG]$ pip install faker
Collecting faker
  Downloading https://files.pythonhosted.org/packages/36/e9/eb926239c310abc1502c
  affe651ac83e72a11c4cc9b9f8d9294be942758b/Faker-14.2.1-py3-none-any.whl (1.6MB)
    100% |████████████████████████████████████████| 1.6MB 807kB/s
Collecting typing-extensions>=3.7.4.3; python_version < "3.8" (from faker)
  Downloading https://files.pythonhosted.org/packages/45/6b/44f7f8f1e110027cf889
  56b59f2fad776cca7e1704396d043f89effd3a0e/typing_extensions-4.1.1-py3-none-any.whl
    100% |████████████████████████████████████████| 256kB 2.9MB/s
Collecting python-dateutil>=2.4 (from faker)
  Downloading https://files.pythonhosted.org/packages/36/7a/87837f39d0296e723bb9
  b62bbb257d0355c7f6128853c78955f57342a56d/python_dateutil-2.8.2-py2.py3-none-any.whl (247kB)
    100% |████████████████████████████████████████| 256kB 2.9MB/s
Collecting six>=1.5 (from python-dateutil>=2.4->faker)
  Downloading https://files.pythonhosted.org/packages/d9/5a/e7c31adbe875f2abbb91
  bd84cf2dc52d792b5a01506781dbcf25c91daf11/six-1.16.0-py2.py3-none-any.whl
Installing collected packages: typing-extensions, six, python-dateutil, faker
Successfully installed faker-14.2.1 python-dateutil-2.8.2 six-1.16.0 typing-extensions-4.1.1
```

Instalación de faker a través de Python

The screenshot displays a virtual machine environment with two main windows. The left window is a terminal titled 'student@fgnadockerssrv:~/faker_BURGERKONG' showing the installation of the Faker library using Python's venv and pip. The right window is a MySQL Workbench interface titled 'BURGERKONG UNIVERSITY' showing a query result for 'clientes'.

Terminal Output:

```
student@fgnadockerssrv:~/faker_BURGERKONG
File Edit View Search Terminal Help

self._mysql.query()
_mysql.connector.MySQLInterfaceError: Data too long for column 'telefono' at row
1

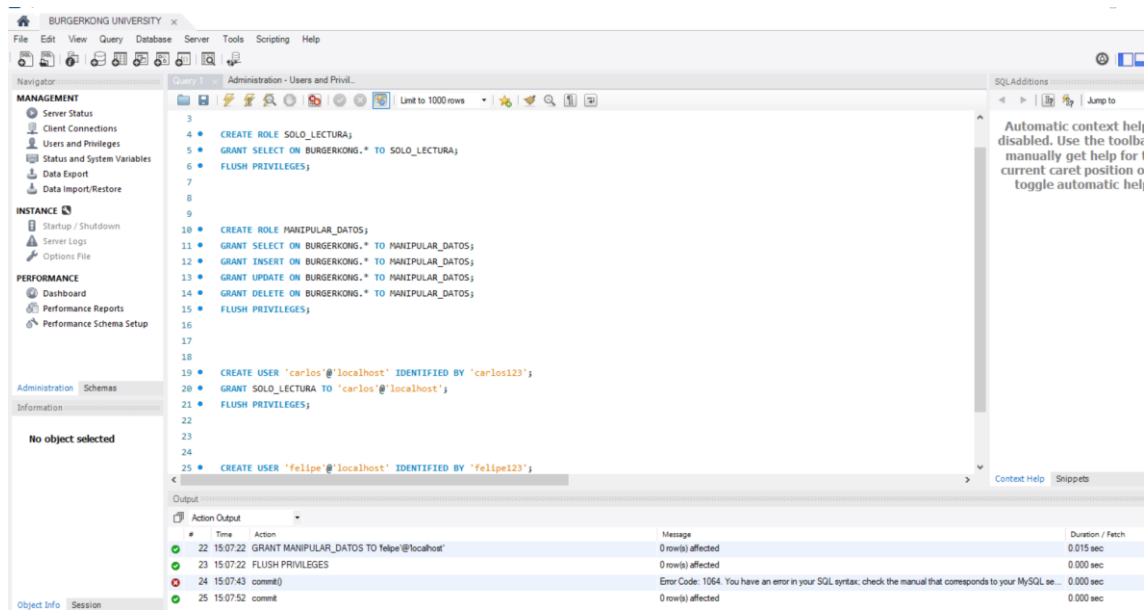
The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "faker_BURGERKONG script.py", line 107, in <module>
    cursor.executemany(clientes_query, clientes_data)
  File "/home/student/.local/lib/python3.8/site-packages/mysql/connector/cursor_
  .py", line 455, in executemany
    return self.execute(statement)
  File "/home/student/.local/lib/python3.8/site-packages/mysql/connector/cursor_
  .py", line 336, in execute
    result = self._cnx.cmd_query()
  File "/home/student/.local/lib/python3.8/site-packages/mysql/connector/connect
  ion.py", line 619, in cmd_query
    raise get_mysql_exception(
mysql.connector.errors.DataError: 1406 (22001): Data too long for column 'telefono' at row 1
[student@fgnadockerssrv faker_BURGERKONG]$ venv python3.8 faker_BURGERKONG script.py
[student@fgnadockerssrv faker_BURGERKONG]$ python3.8 faker_BURGERKONG script.py
```

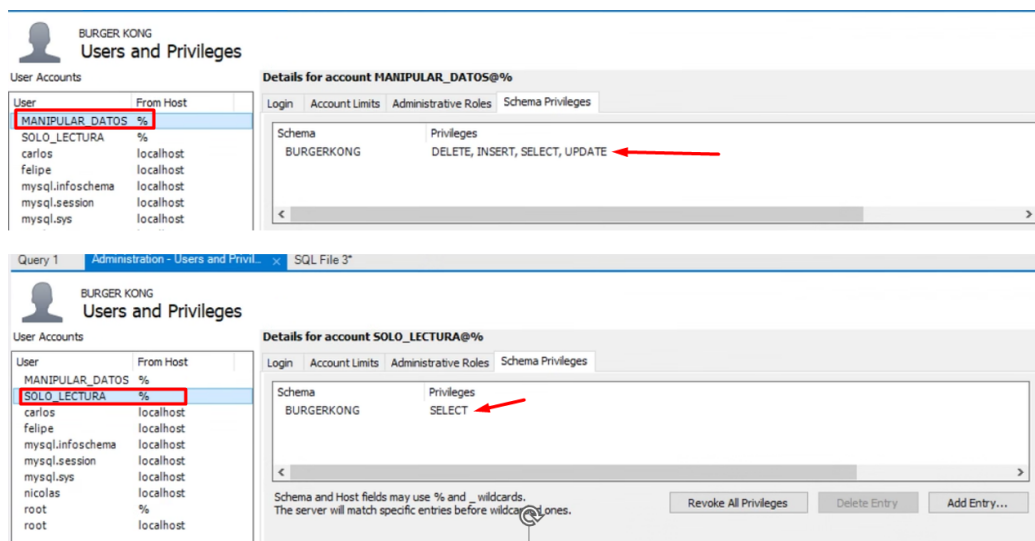
MySQL Workbench Output:

id_clientes	nombre	direccion	telefono
45	Christopher Gutierrez	43877 Hall Parkway West Ch...	761-4...
46	Jessica Williams	43341 Roy Spurs Apt. 180 Jah...	001-4...
47	Kenneth Graham	1817 Nancy Keys Apt. 730 Tra...	637-4...
48	Lisa Garcia	488 Gary Squares New Laura...	859-4...
49	Jennifer Williams	PSC 4584, Box 9011 APO AA...	328-4...
50	Courtney Valencia	13847 Castro Loaf South Just...	982-4...

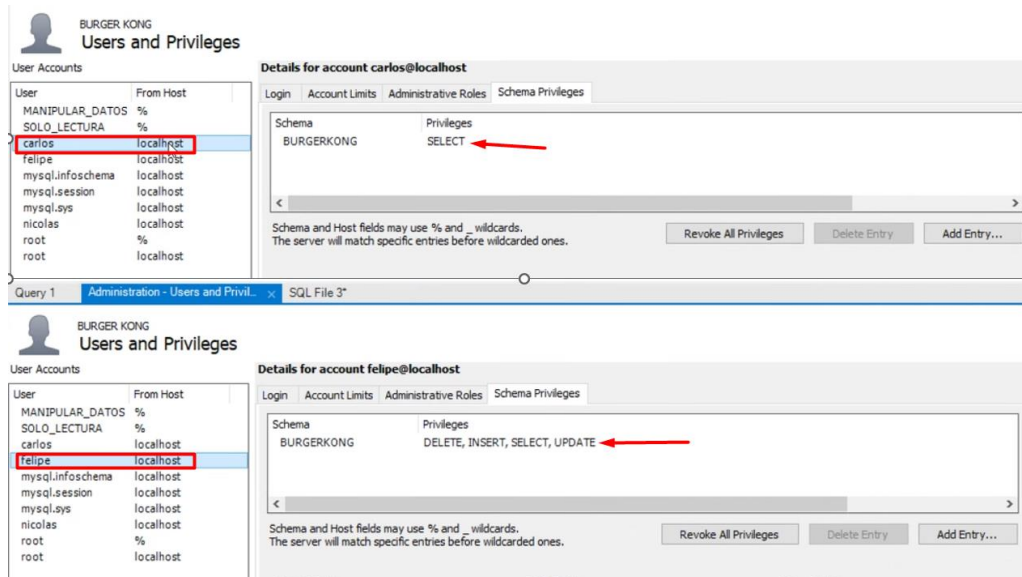
Ejecución de script de Python para la creación de 50 datos aleatorios en cada tabla de la base de datos



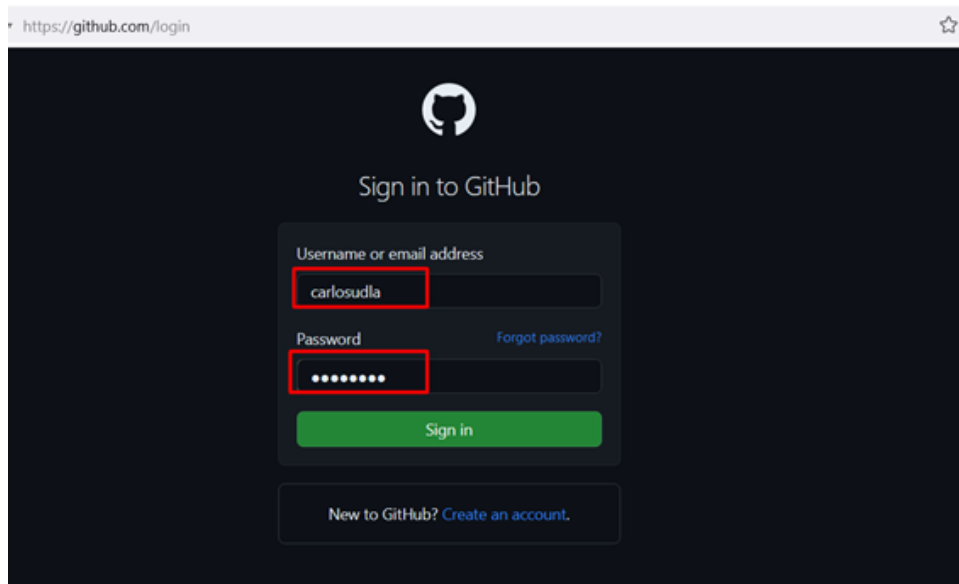
Se procede con la creación de roles



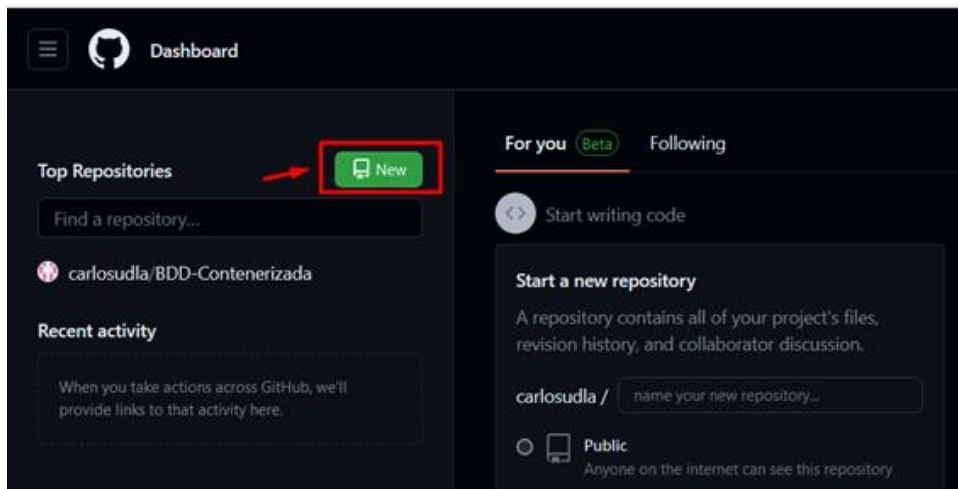
Se comprueba mediante la interfaz de workbench que los roles se han creado con los privilegios especificados.



La creación de los 2 usuarios con sus privilegios designados de acuerdo al rol se ha creado de manera adecuada.



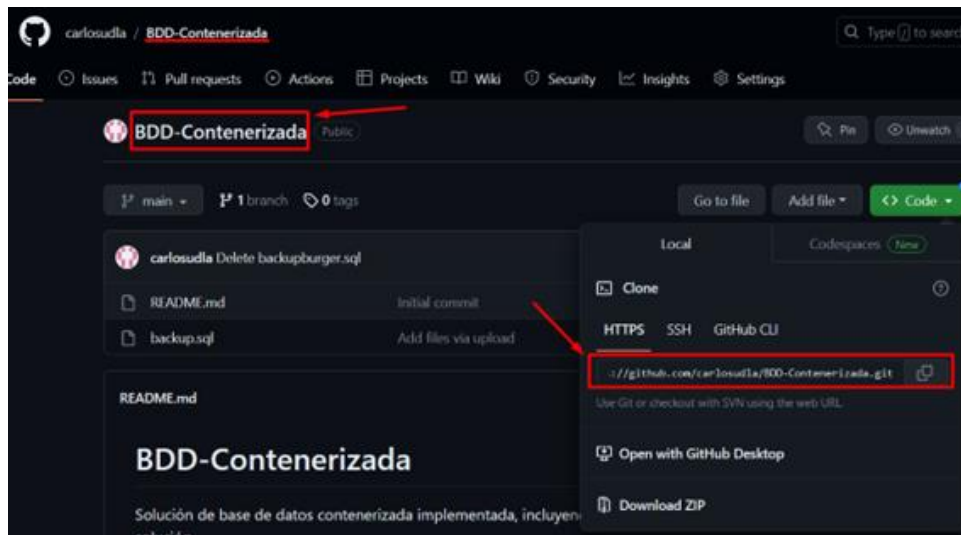
Para una vez registrado el usuario con cuenta de correo electrónico se procede a iniciar sesión en el portal de GitHub.



Al inicio de la página se selecciona la opción que permite añadir un nuevo repositorio en el cual se guardarán los archivos del proyecto.

The image shows the 'Create a new repository' form on GitHub. The form has a title 'Create a new repository' and a subtitle 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' Below this, there is a note 'Required fields are marked with an asterisk (*)'. The 'Owner' field is set to 'carlosudla'. The 'Repository name' field is highlighted with a red box and contains the text 'BDD-Contenerizada.'. A red arrow points to this field from the right. Below the name field, there is a green checkmark and the text 'BDD-Contenerizada. is available.'. The 'Description' field is optional and empty. The 'Public' radio button is selected, and the 'Private' radio button is unselected. The 'Initialize this repository with:' section has the 'Add a README file' checkbox checked. The 'Add .gitignore' section has the '.gitignore template: None' dropdown selected. The 'Choose a license' section has the 'License: None' dropdown selected. At the bottom, there is a green 'Create repository' button highlighted with a red box, and a red arrow points to it from the left. A note at the bottom says 'You are creating a public repository in your personal account.'

En este apartado se le asigna un nombre al repositorio y se asegura que el mismo esté configurado como “público” para que todo usuario tenga acceso. Por último, se da clic en “crear repositorio” para final la creación.



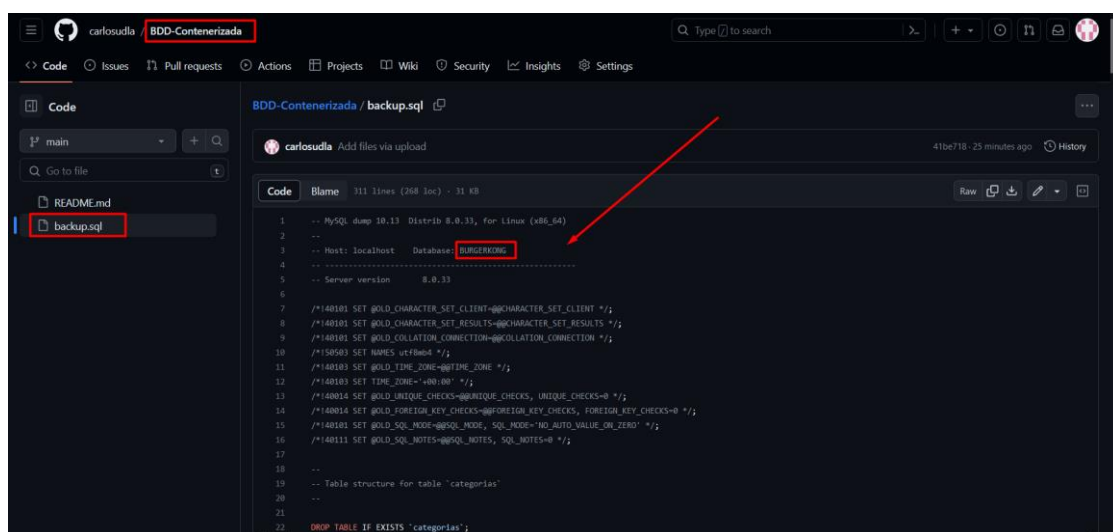
Una vez creado el repositorio se debe anotar el nombre del mismo y en el apartado “Code” extraer el enlace que nos permitirá crear la conexión desde el servidor hasta el repositorio.

```

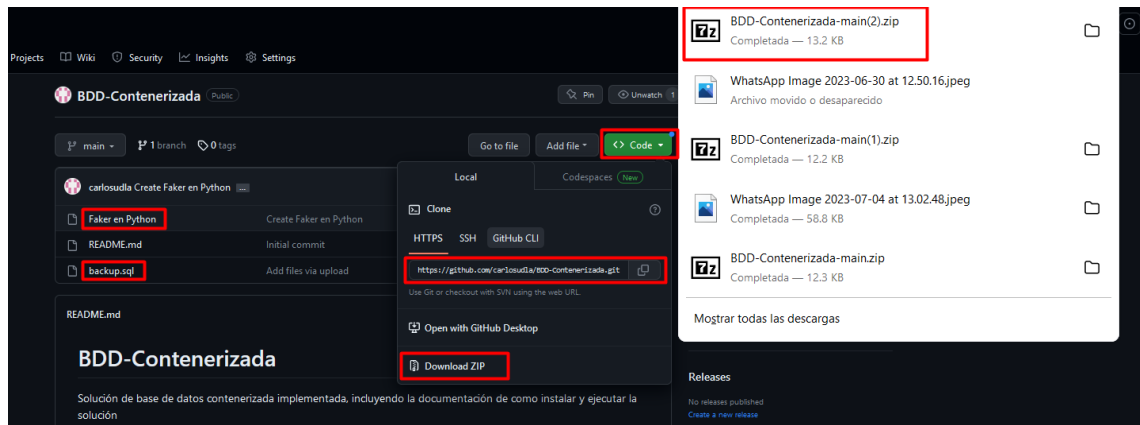
~
[root@fgnadockersrv BDD-Contenerizada]# docker exec abd32ca4b045 /usr/bin/mysqldump -u root --password=Proyecto123 BURGERKONG > backup.sql
mysqldump: [Warning] Skipping '!includedir /etc/mysql/conf.d/' directive as maximum include recursion level was reached in file /etc/mysql/conf.d/my.cnf at line 37.
mysqldump: [Warning] Using a password on the command line interface can be insecure.
[root@fgnadockersrv BDD-Contenerizada]# git init
Reinitialized existing Git repository in /home/student/BDD-Contenerizada/.git/
[root@fgnadockersrv BDD-Contenerizada]# git add backup.sql
[root@fgnadockersrv BDD-Contenerizada]#

```

Para realizar la conexión del servidor de docker hasta el repositorio de github es necesario conocer el usuario y la contraseña, una vez realizado esto verificamos que GIT se encuentra instalado en sistema. Mediante el comando DOCKER EXEC realizamos un backup de la base de datos que vamos a transferir al repositorio y luego subimos dicha copia de la base al repositorio.



Verificamos haciendo REFRESH al repositorio en GitHub y comprobamos que la base de datos ha sido guardada con éxito.



Para compartir el repositorio ya que es de dominio público contamos con dos opciones, ya sea realizando una búsqueda detallada dentro de GitHub o mediante el enlace que podremos compartir tal y como lo muestra la imagen arriba. El apartado "CODE" también nos permitirá descargar en documento ZIP todos los archivos que se encuentren en el repositorio.

Este equipo > Descargas > BDD-Contenerizada-main(2) > BDD-Contenerizada-main

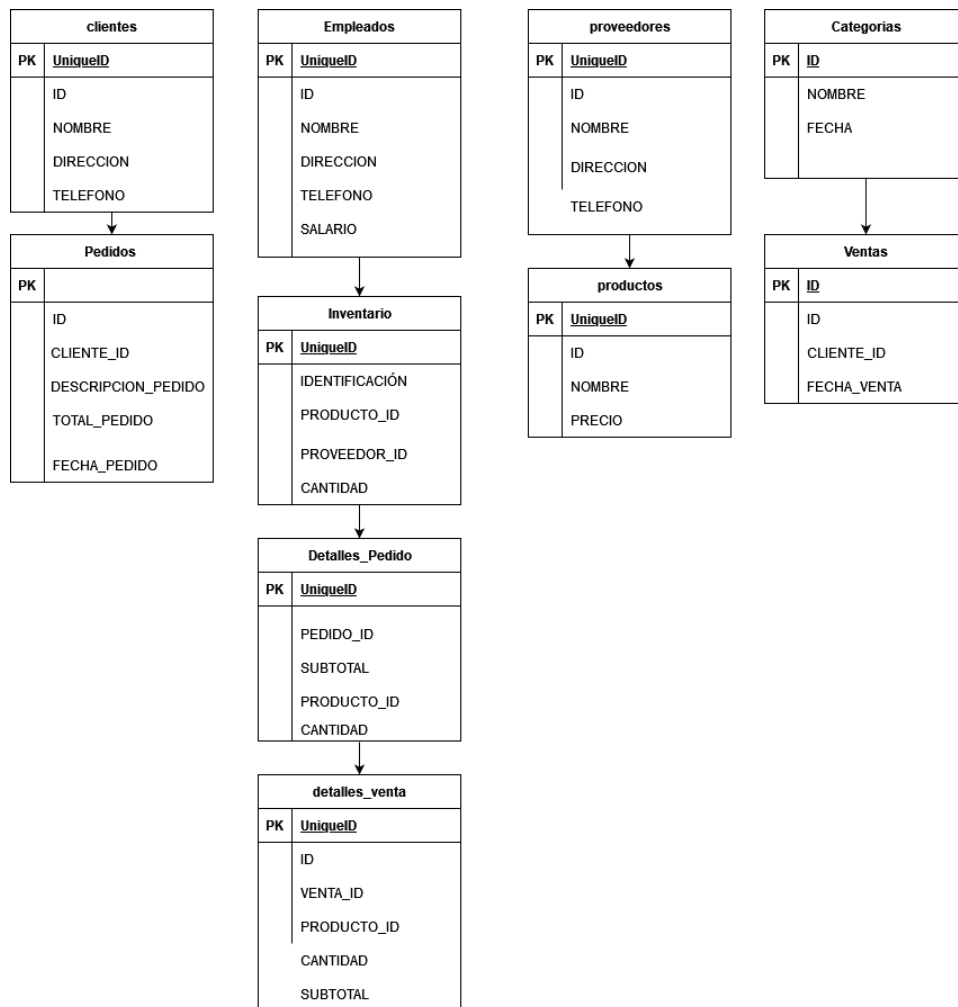
Nombre	Fecha de modificación	Tipo	Tamaño
backup	04/07/2023 15:37	Microsoft SQL Ser...	31 KB
Faker en Python	04/07/2023 15:37	Archivo	5 KB
README	04/07/2023 15:37	Archivo de origen ...	1 KB

S DE DATOS

archivos del repositorio

Una vez descargado y después de descomprimir el archivo se verifica en la carpeta donde se descargó que todos los mismos archivos del repositorio se han descargado con éxito.

Diseño de la base de datos:



Diccionario de Datos:

Tabla	Campo	Tamaño	Tipo de dato	Descripción
Cliente	id	N/A	INT	Identificador único
Cliente	nombre	50	VARCHAR(50)	Nombre del cliente
Cliente	direccion	100	VARCHAR(100)	Dirección del cliente
Cliente	telefono	15	VARCHAR(15)	Número de teléfono del cliente
Empleado	id	N/A	INT	Identificador único
Empleado	nombre	50	VARCHAR(50)	Nombre del empleado
Empleado	direccion	100	VARCHAR(100)	Dirección del empleado

Empleado	telefono	15	VARCHAR(15)	Número de teléfono del empleado
Empleado	salario	N/A	DECIMAL(10, 2)	Salario del empleado
Proveedor	id	N/A	INT	Identificador único
Proveedor	nombre	50	VARCHAR(50)	Nombre del proveedor
Proveedor	direccion	100	VARCHAR(100)	Dirección del proveedor
Proveedor	telefono	15	VARCHAR(15)	Número de teléfono del proveedor
Categoria	id	N/A	INT	Identificador único
Categoria	nombre	50	VARCHAR(50)	Nombre de la categoría de productos
Producto	id	N/A	INT	Identificador único
Producto	nombre	50	VARCHAR(50)	Nombre del producto
Producto	precio	N/A	DECIMAL(10, 2)	Precio del producto
Producto	categoria_id	N/A	INT	ID de la categoría
Producto	proveedor_id	N/A	INT	ID del proveedor
Inventario	id	N/A	INT	Identificador único
Inventario	producto_id	N/A	INT	ID del producto en el inventario
Inventario	proveedor_id	N/A	INT	ID del proveedor en el inventario
Inventario	cantidad	N/A	INT	Cantidad en inventario
Inventario	fecha_ingreso	N/A	DATE	Fecha de ingreso al inventario
Pedido	id	N/A	INT	Identificador único
Pedido	cliente_id	N/A	INT	ID del cliente
Pedido	fecha_pedido	N/A	DATE	Fecha del pedido
Pedido	total	N/A	DECIMAL(10, 2)	Total del pedido


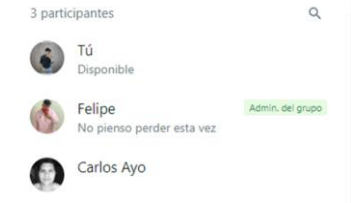
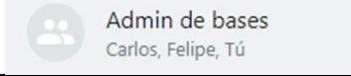
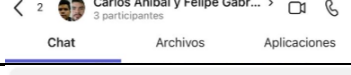
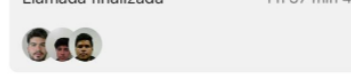

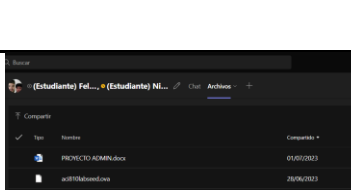
detalles_pedido	id	N/A	INT	Identificador único
detalles_pedido	pedido_id	N/A	INT	ID del pedido
detalles_pedido	producto_id	N/A	INT	ID del producto
detalles_pedido	cantidad	N/A	INT	Cantidad del producto
detalles_pedido	subtotal	N/A	DECIMAL(10, 2)	Subtotal del detalle
Ventas	id	N/A	INT	Identificador único
Ventas	cliente_id	N/A	INT	ID del cliente
Ventas	fecha_venta	N/A	DATE	Fecha de la venta
Ventas	total	N/A	DECIMAL(10, 2)	Total de la venta
detalles_venta	id	N/A	INT	Identificador único
detalles_venta	venta_id	N/A	INT	ID de la venta
detalles_venta	producto_id	N/A	INT	ID del producto
detalles_venta	cantidad	N/A	INT	Cantidad del producto
detalles_venta	subtotal	N/A	DECIMAL(10, 2)	Subtotal del detalle

Solución:

la solución para las ventas en una organización de comida rápida involucra el registro de las transacciones de venta, el seguimiento de los detalles de cada venta, el cálculo del total de la venta, la generación de informes y la integración con sistemas de pago. Esto proporciona una visión completa y actualizada de las ventas, permitiendo una gestión efectiva y una toma de decisiones informada.

Informe de cumplimiento de los siguientes criterios de desempeño:

Criterio	Porcentaje de Cumplimiento	Evidencias
Definición de los Objetivos de Trabajo	Se cumplió un 90%	Nicolas Alejandro, +2 Entrante de (Estudiante) Nicolas Alejandro Camacho Zambrano 1h 19m
Definición de Cronograma	Se cumplió un 85%	Nicolas Alejandro, +2 Entrante de (Estudiante) Nicolas Alejandro Camacho Zambrano 1h 19m

Definición de Roles	Se cumplió un 90%		
Asignación de Roles	Se cumplió un 100%		
Asignación de Responsabilidades	Se cumplió un 100%		
Cronograma de reuniones de trabajo	Se cumplió un 95%		
Ideas aportadas para la implementación de la solución por cada participante	Se cumplió un 95%		
Aporte individual para la consecución de los Objetivos de Trabajo	Se cumplió un 85%		
Aporte individual para la resolución de posibles conflictos	Se cumplió un 90%		

Evidencia escrita:

Se creo un grupo de WhatsApp para definir los objetivos del trabajo, los roles que cada integrante iba a tener y como íbamos a poder aportar en el proyecto, nos sirvió para asignar las responsabilidades de cada uno, así como para también aportar con ideas y generar acordar reuniones en Teams. Durante el transcurso de las semanas nos reuníamos para realiza avances tanto como en el documento escrito como en el ambiente de prueba que se había ya instalado, cabe recalcar que, aunque el proyecto final se ejecutó en un solo ambiente, cada uno de los participantes realizó pruebas dentro de su propio entorno con el fin de no desconfigurar la máquina virtual que se iba a presentar en la defensa.