

WEBANIMAL

Informe migración de datos

Introducción

Una migración de datos es el proceso mediante el cual realizamos una transferencia de datos de unos sistemas de almacenamiento de datos a otros, esto con el fin de reemplazar o actualizar información. A continuación, se presentan los objetivos, el alcance, justificación y la configuración de los gestores de bases de datos MySQL y PostgreSQL.

Objetivos

- Este documento tiene como objetivo brindar instrucciones para la configuración y conexión a los gestores de base de datos
- Facilitar la implementación de las migraciones de datos a los diferentes gestores de base de datos.

Justificación

Las migraciones de datos se llevan a cabo para reemplazar o actualizar servidores o equipos de almacenamiento o para llevar a cabo el mantenimiento de un servidor. Por eso el propósito de este documento es aplicar la migración y definir procesos que permitan apoyar para la mejora de este y la configuración de las bases de datos.

Conexión del SGBD

El SGBD (Sistema gestor de base de datos) es un software que se utiliza para acceder, extraer y administrar datos almacenados en una fuente o base de datos. Permiten definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de estos.

Conexión con MySQL desde Django.

MySQL es un sistema de administración de bases de datos relacional de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado. A continuación, se presenta el paso a paso para la conexión a la base de datos ya dicha.

1.

Abrir el editor de texto que se tenga disponible.

2.

Posteriormente activaremos el ambiente virtual, esto se puede lograr con el siguiente comando:

```
./activate
```

3. Abriremos la carpeta en la que se encuentra el ambiente virtual

4. Instalamos el contralador de la base de datos con el siguiente comando

```
pip install mysqlclient
```

Después de esta deberá arrojar el siguiente mensaje, para confirmar que fue instalado correctamente.

```
(env) A pip install mysqlclient
Collecting mysqlclient
  Downloading mysqlclient-2.0.3-cp39-cp39-win_amd64.whl (179 kB)
    | 179 kB 3.3 MB/s
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.0.3
WARNING: You are using pip version 21.1.3; however, version 21.2.4 is available.
You should consider upgrading via the 'c:\users\janus\documents\github\personal-laurahidalgo78\hola_mundo\webanisaal2\env\scripts\python.exe -m pip install --upgrade pip' command.
```

5. Ahora abriremos el Visual Studio Code y dentro de la carpeta del ambiente virtual creamos un archivo llamado db.py

6. En el archivo db.py pegaremos las siguientes lineas de codigo.

```
1  import os
2
3  BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
4
5  SQLITE = {
6      'default': {
7          'ENGINE': 'django.db.backends.sqlite3',
8          'NAME': os.path.join(BASE_DIR, 'db.sqlite3')
9      }
```

```

10 }
11
12 POSTGRESQL = {
13     'default': {
14         'ENGINE': 'django.db.backends.postgresql_psycopg2',
15         'NAME': 'db',
16         'USER': 'postgres',
17         'PASSWORD': 'password',
18         'HOST': 'localhost',
19         'PORT': '5432'
20     }
21 }
22
23 MYSQL = {
24     'default': {
25         'ENGINE': 'django.db.backends.mysql',
26         'NAME': 'db',
27         'USER': 'root',
28         'PASSWORD': 'password',
29         'HOST': 'localhost',
30         'PORT': '3306'
31     }
32 }

```

7. En el apartado de MYSQL configurar la credenciales de la base de datos como el user, password y name.

```

1  MYSQL = {
2      'default': {
3          'ENGINE': 'django.db.backends.mysql',
4          'NAME': 'db.prueba',
5          'USER': 'root',
6          'PASSWORD': '123',
7          'HOST': 'localhost',
8          'PORT': '3306'
9      }
10 }

```

8. Abrir el archivo [settings.py](#), aquí se va a configurar la base de datos a usar. Procederemos a importar todas las librerías necesarias para este propósito.

```
1 import os
2 import webanimal.db as db
3 from pathlib import Path
```

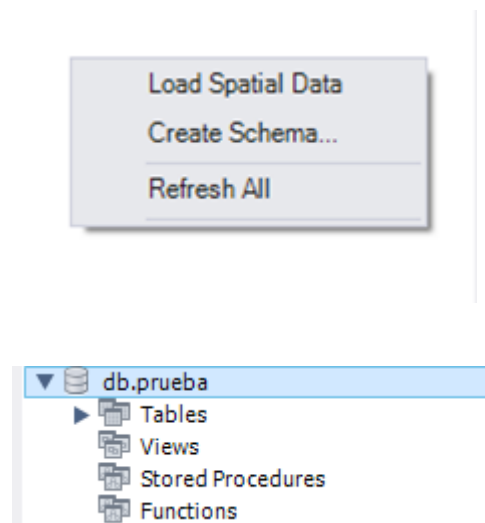
9. En el archivo settings.py bajamos hasta el apartado DATABASES y borramos el contenido, y lo cambiaremos por db.MYSQL.

```
DATABASES = db.MYSQL
```

10.

Crearemos una base de datos vacía con el mismo nombre que le pusimos a "name" en el archivo db.py

11. Abriremos el motor de base de datos MySQL Workbench.



- Migración de datos

12.

Posteriormente nos dirigimos a Windows terminal o el editor de texto de su preferencia y ejecutamos el siguiente comando.

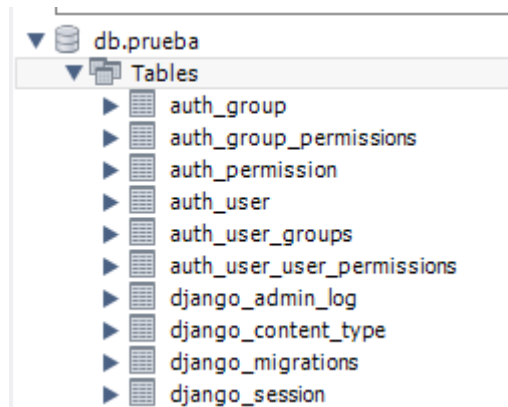
```
python manage.py migrate
```

13. Después de ejecutar el comando deberá mostrar el siguiente mensaje como confirmación

```
C:\Users\Janus\Documents\GitHub\personal1-laurahidalgo78\hola_mundo\webanimal2\webanimal2 (master -> origin)
(env) λ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

14.

En el motor de base de datos MYSQL podrás encontrar las tablas de la base de datos.



Conexión con PostgreSQL desde Django

Es un sistema para gestionar bases de datos de muy alto nivel, de código abierto de administración de bases de datos del tipo relacionales y completamente de software libre

1. Abrir el editor de texto que se tenga disponible.

2. Posteriormente activaremos el ambiente virtual, esto se puede logra con el siguiente comando:

```
./activate
```

3. Abriremos la carpeta en la que se encuentra el ambiente virtual
4. Instalamos el controlador de la base de datos con el siguiente comando

```
pip install psycopg2
```

Después de esta deberá arrojar el siguiente mensaje, para confirmar que fue instalado correctamente.

```
C:\Users\Janus\Documents\GitHub\personal1-laurahidalgo78\hola_mundo\webanimal\webanimal (master -> origin)
(env) λ pip install psycopg2
Collecting psycopg2
  Downloading psycopg2-2.9.1-cp39-cp39-win_amd64.whl (1.2 MB)
    | 1.2 MB 3.3 MB/s
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.1
WARNING: You are using pip version 21.1.3; however, version 21.2.4 is available.
You should consider upgrading via the 'c:\users\janus\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.
```

5.
Ahora abriremos el Visual Studio Code, dentro de la carpeta del ambiente virtual creamos un archivo llamado db.py.
6. En el archivo db.py pegaremos las siguientes líneas de código.

```
1 import os
2
3 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
4
5 SQLITE = {
6     'default': {
7         'ENGINE': 'django.db.backends.sqlite3',
8         'NAME': os.path.join(BASE_DIR, 'db.sqlite3')
9     }
```

```

10 }
11
12 POSTGRESQL = {
13     'default': {
14         'ENGINE': 'django.db.backends.postgresql_psycopg2',
15         'NAME': 'db',
16         'USER': 'postgres',
17         'PASSWORD': 'password',
18         'HOST': 'localhost',
19         'PORT': '5432'
20     }
21 }
22
23 MYSQL = {
24     'default': {
25         'ENGINE': 'django.db.backends.mysql',
26         'NAME': 'db',
27         'USER': 'root',
28         'PASSWORD': 'password',
29         'HOST': 'localhost',
30         'PORT': '3306'
31     }
32 }

```

7. En el apartado de POSTGRESQL configurar la credenciales de la base de datos como el user, password y name.

```

1  POSTGRESQL = {
2      'default': {
3          'ENGINE': 'django.db.backends.postgresql_psycopg2',
4          'NAME': 'BASE_DATOS_1',
5          'USER': 'USUARIO',
6          'PASSWORD': '53658',
7          'HOST': 'localhost',
8          'PORT': '5432'
9      }
10 }

```

8. Abrir el archivo settings.py, aquí se va a configurar la base de datos a usar. Procederemos a importar todas las librerías necesarias para este propósito.

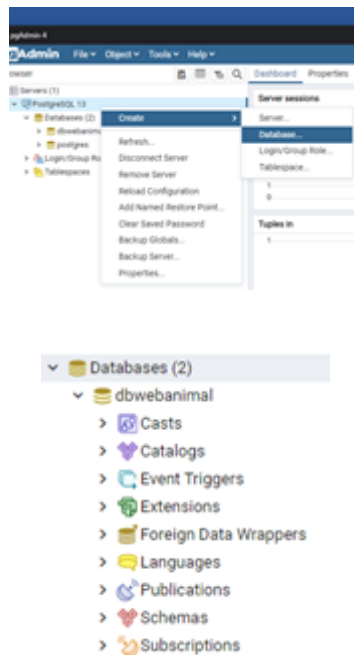

```
1 import os
2 import webanimal.db as db
3 from pathlib import Path
```

9. En el archivo settings.py bajamos hasta el apartado DATABASES y borramos el contenido, y lo cambiaremos por db.POSTGRESQL

```
DATABASES = db.POSTGRESQL
```

10. Abriremos el motor de base de datos PostgreSQL.

11. Crearemos una base de datos vacía con el mismo nombre que le pusimos a "name" en el archivo db.py.



- Migración de datos

12. Posterior nos dirigimos al editor de texto de su preferencia y ejecutamos el siguiente comando

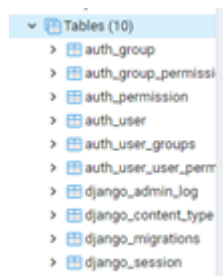
```
python manage.py migrate
```

Después de ejecutar el comando deberá mostrar el siguiente mensaje como confirmación

```
C:\Users\Janus\Documents\GitHub\personal1-laurahidalgo78\hola_mundo\webanimal2\webanimal2 (master -> origin)
(env) λ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK

C:\Users\Janus\Documents\GitHub\personal1-laurahidalgo78\hola_mundo\webanimal2\webanimal2 (master -> origin)
(env) λ |
```

13. En el motor de base de datos posgreSQL podrás encontrar las tablas de la base de datos



Conexión a SQLite desde django

Es una herramienta de software libre, que permite almacenar información en dispositivos empujados de una forma sencilla, eficaz, potente y rápida.

1. Abrir el editor de texto que se tenga disponible.
2. Posteriormente activaremos el ambiente virtual, esto se puede logra con el siguiente comando:

```
./activate
```

3. Ahora abriremos el Visual Studio Code, dentro de la carpeta del ambiente virtual creamos un archivo llamado db.py.

4. En el archivo db.py pegaremos las siguientes líneas de código.

```
1  import os
2
3  BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
4
5  SQLITE = {
6      'default': {
7          'ENGINE': 'django.db.backends.sqlite3',
8          'NAME': os.path.join(BASE_DIR, 'db.sqlite3')
9      }
10 }
11
12 POSTGRESQL = {
13     'default': {
14         'ENGINE': 'django.db.backends.postgresql_psycopg2',
15         'NAME': 'db',
16         'USER': 'postgres',
17         'PASSWORD': 'password',
18         'HOST': 'localhost',
19         'PORT': '5432'
20     }
21 }
22
23 MYSQL = {
24     'default': {
25         'ENGINE': 'django.db.backends.mysql',
26         'NAME': 'db',
27         'USER': 'root',
28         'PASSWORD': 'password',
29         'HOST': 'localhost',
30         'PORT': '3306'
31     }
32 }
```

5. En el apartado de SQLITE configurar el nombre de la base de datos.

```

1  SQLITE = {
2      'default': {
3          'ENGINE': 'django.db.backends.sqlite3',
4          'NAME': os.path.join(BASE_DIR, 'db.sqlite3')
5      }
6  }

```

6. Abrir el archivo settings.py, aquí se va a configurar la base de datos a usar. Procederemos a importar todas las librerías necesarias para este propósito.

```

1  import os
2  import webanimal.db as db
3  from pathlib import Path

```

7. En el archivo settings.py bajamos hasta el apartado DATABASES y borramos el contenido, y lo cambiaremos por db.SQLITE

```
DATABASES = db.SQLITE
```

8. Abriremos el motor de base de datos SQLite.

9. Crearemos una base de datos vacía con el mismo nombre que le pusimos a "name" en el archivo db.py.



- Migración de datos

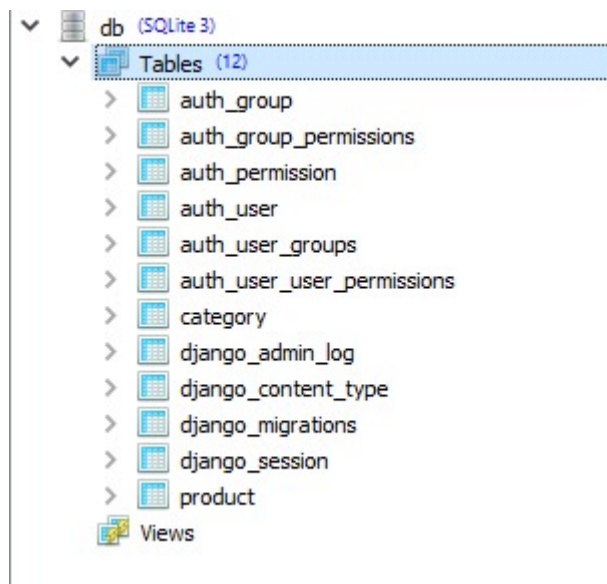
10. Posterior nos dirigimos al editor de texto de su preferencia y ejecutamos el siguiente comando:

```
python manage.py migrate
```

Después de ejecutar el comando deberá mostrar el siguiente mensaje como confirmación

```
(env) PS C:\Users\Janus\Documents\GitHub\personal1-laurahidalgo78\hola_mundo\webanimal\webanimal2> py manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, product, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying product.0001_initial... OK
  Applying sessions.0001_initial... OK
```

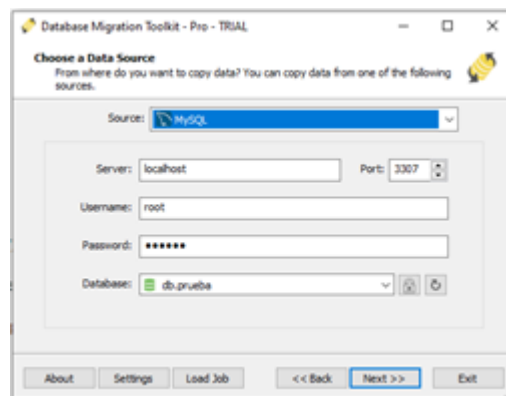
11. En el motor de base de datos SQLite podrás encontrar las tablas de la base de datos



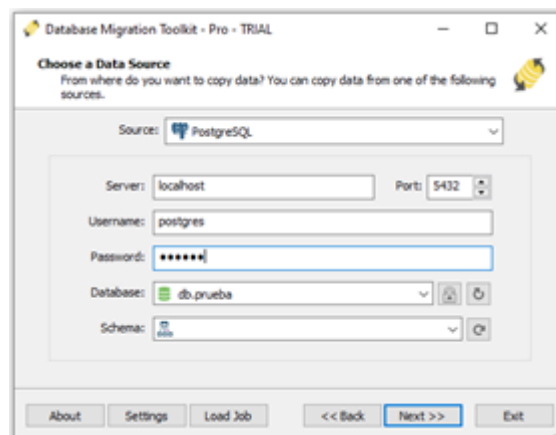
Migración de base de datos desde el Kit de herramientas de migración de bases de datos

1. Seleccionar la base de datos a la que se va a copiar los datos.

2. Ingresar el nombre del servidor
3. Ingresar el nombre del usuario
4. Ingresar su contraseña
5. Dar clic en next

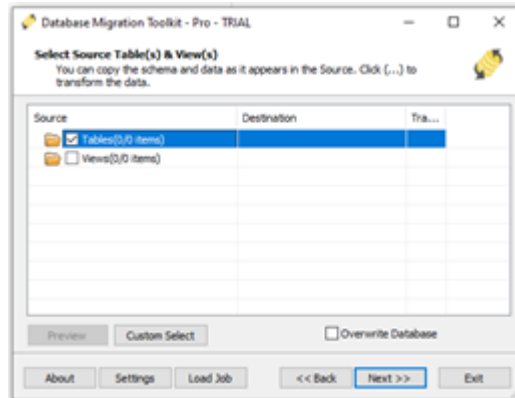


6. Seleccione el gestor al que va a migrar los datos
7. Ingresar el nombre de usuario
8. Ingresar su contraseña
9. Dar clic en next

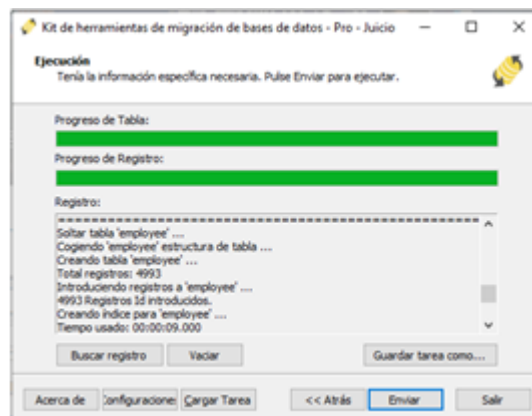


10. Seleccionamos el apartado de tablas

11. Dar clic en next



12. Puede configurar el método de transferencia de datos (Sobrescribir tabla / Vaciar datos / Agregar datos / Omitir tabla) o filtrar los datos antes de transferirlos. Elija "Asignación de campos" opción, puede redefinir los campos en la tabla de destino, por ejemplo: nombre del campo, tipo de datos, valor predeterminado, comentario y también y dar clic en ok.



13. Haga clic en "Enviar" para comenzar la migración, este kit de herramientas lo ayudará a migrar rápidamente los datos de MySQL a PostgreSQL sin intervención.

14. Ya se abran migrado los datos