



CONFIGURACIÓN DE LA BASE DE DATOS PERSISTENTE EN TUTRAMITE

La configuración de la base de datos se encuentra en el archivo de configuración de Django, llamado, por omisión, settings.py en el Tutramite se encuentra en la ruta /tramite/settings.py. Edita este archivo y busca las opciones de la variable DATABASES, el cual es un diccionario que contiene los ajustes necesarios, para configurar la base datos:

```
ENGINE = ''
NAME = ''
USER = ''
PASSWORD = ''
HOST = ''
DATABASE_PORT = ''
```

Aquí hay un resumen de cada propiedad.

- **ENGINE:** Le indica a Django qué base de datos utilizar. Si usas una base de datos con Django, ENGINE debe configurarse con una cadena de los mostrados en la Tabla 1.

Configuración	Base de datos	Adaptador requerido
django.db.backends.postgresql_psycopg2	PostgreSQL	Psycopg version 2.x
django.db.backends.mysql	MySQL	MySQLdb
django.db.backends.sqlite3	SQLite	No necesita adaptador
django.db.backends.oracle	Oracle	cx_Oracle

Tabla 1. Adaptadores requeridos en Django

Nota: Cualquiera que sea la base de datos que se use, se necesitará instalar el adaptador apropiado. Cada uno de estos está disponible libremente en la web.

- **NAME:** Le indica a Django el nombre de tu base de datos. Si estás usando SQLite, especifica la ruta completo del sistema de archivos



hacia el archivo de la base de datos (por ej. `'/home/django/datos.db'`).

- **USER:** Le indica a Django cual es el nombre de usuario a usar cuando se conecte con tu base de datos. Si estás usando SQLite, deja este en blanco.
- **PASSWORD:** Le indica a Django cual es la contraseña a utilizar cuando se conecte con tu base de datos. Si estás utilizando SQLite o tienes una contraseña vacía, deja este en blanco.
- **HOST:** Le indica a Django cual es el host a usar cuando se conecta a tu base de datos. Si tu base de datos está sobre la misma computadora que la instalación de Django (o sea localhost), deja este en blanco. Si estás usando SQLite, deja este en blanco.
- **PORT:** Le indica a Django qué puerto usar cuando se conecte a la base de datos. Si estás utilizando SQLite, deja este en blanco. En otro caso, si dejas este en blanco, el adaptador de base de datos subyacente usará el puerto por omisión acorde al servidor de base de datos.

La variable `DATABASES`, por omisión usa la configuración más simple posible, la cual está configurada para utilizar SQLite, por lo que no se tendrá que configurar nada, si vas a usar SQLite como base de datos que es la que por defecto se encuentra configurada en el proyecto TuTramite:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    }  
}
```

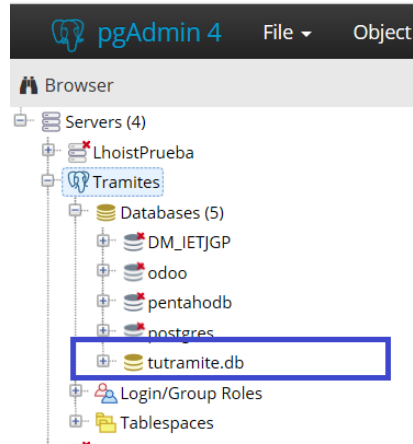
Sin embargo si se requiere usar otra base de datos como MySQL, Oracle, o PostgreSQL es necesario especificar algunos parámetros adicionales, que serán requeridos en el archivo de configuración.

CONFIGURACIÓN DE POSTGRESQL EN DJANGO

El siguiente ejemplo asume que quieres utilizar PostgreSQL debemos seguir los siguientes pasos:



- ❖ Suponemos que tenemos instalado nuestro gestor de base de datos y hemos creado una base de datos, como muestra la siguiente ilustración utilizando postgresQL



- ❖ Paso siguiente será instalar el adaptador indicado en la Tabla 1. Utilizando pip desde nuestro ambiente virtual podemos instalar el adaptador para PostgreSQL digitando el siguiente comando.

```
(entornoVirtual)$ pip install psycopg2
```

configuramos los parámetros de conexión de nuestra base de datos en el archivo de configuración settings.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': 'tutramite.db',  
        'USER': 'postgres',  
        'PASSWORD': 'password',  
        'HOST': '127.0.0.1',  
        'PORT': '5432',  
    }  
}
```

Como se podrá observar, lo único que se necesita cambiar es la 'ENGINE', si se quiere usar MySQL u otro gestor de base de datos e introducir los datos apropiados de acuerdo a la base de datos que estés usando.

Una vez que se haya ingresado estas configuraciones, se comprueban. Primero, desde el directorio del proyecto, ejecutando el comando:



```
(entornoVirtual)$ python manage.py shell
```

Notará que comienza un intérprete interactivo de Python. Una vez que se encuentre dentro de la shell, escribir estos comandos para probar la configuración de la base de datos:

```
>>> from django.db import connection
>>> cursor = connection.cursor()
>>>
```

Si no sucede nada, entonces la base de datos está configurada correctamente. De lo contrario revisar el mensaje de error para obtener un indicio sobre qué es lo que está mal.