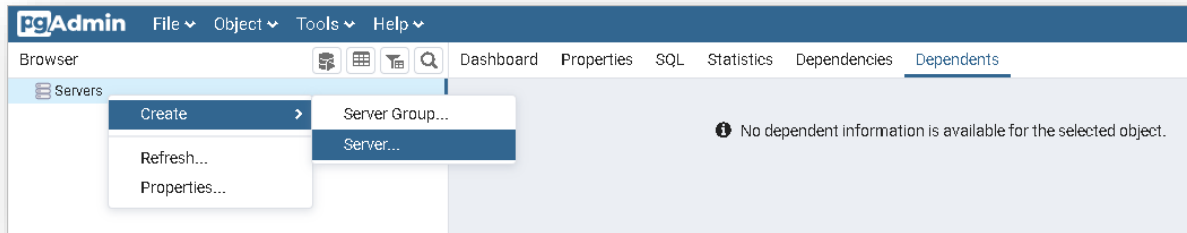


DJANGO CON POSTGRESQL

Por: Jose Fernando Galindo Suarez



CREANDO EL SERVIDOR POSTGRESQL

The screenshot shows the 'Create - Server' dialog box in pgAdmin 4. The dialog has a title bar 'Create - Server' and a close button. It features four tabs: 'General' (selected), 'Connection', 'SSL', 'SSH Tunnel', and 'Advanced'. The 'General' tab contains the following fields and controls:

- Name:** A text input field containing 'SenaServer'.
- Server group:** A dropdown menu showing 'Servers'.
- Background:** A checkbox with an 'X' icon, indicating it is unchecked.
- Foreground:** A checkbox with an 'X' icon, indicating it is unchecked.
- Connect now?:** A checkbox with a checkmark icon, indicating it is checked.
- Comments:** A large text area for entering comments.

At the bottom of the dialog, there is a red error message bar that reads: 'Either Host name, Address or Service must be specified.' Below this bar are four buttons: an information icon, a question mark icon, a 'Cancel' button, a 'Reset' button, and a 'Save' button.

Create - Server

General

Connection

SSL

SSH Tunnel

Advanced

Host name/address

localhost

Port

5432

Maintenance database

postgres

Username

postgres

Password

...

Save password?

☐

Role

Service

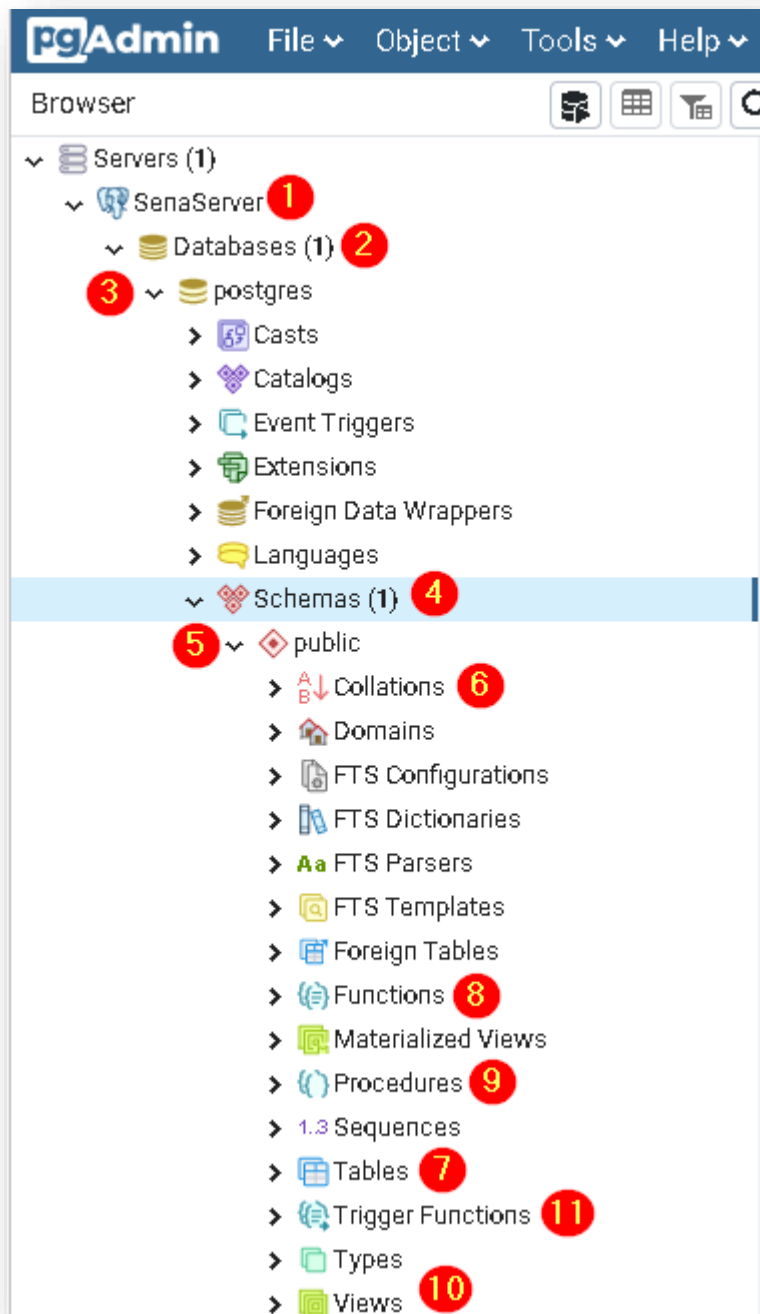
i

?

✕ Cancel

↺ Reset

4 Save



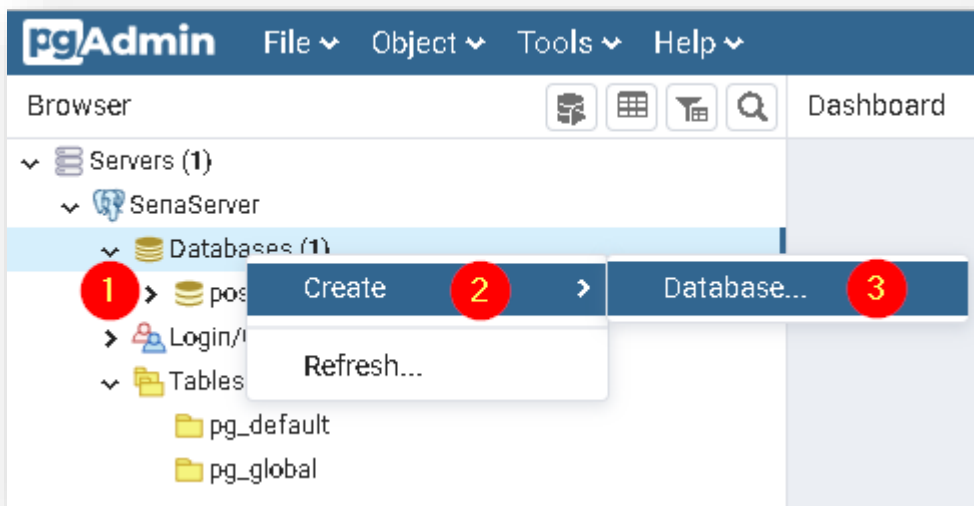
CONFIGURANDO A settings.py PARA POSTGRESQL

```
POSTGRES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'academico',
        'USER': 'postgres',
        'PASSWORD': '',
        'HOST': '127.0.0.1',
        'DATABASE_PORT': '5432',
    }
}
DATABASES = POSTGRES
```

INSTALANDO PYTHON PARA POSTGRESQL

```
pip install psycopg2
```

CREANDO LA BASE DE DATOS “académico”



Create - Database

1

General

3

DefinitionSecurityParametersAdvancedSQL

Database

academico2

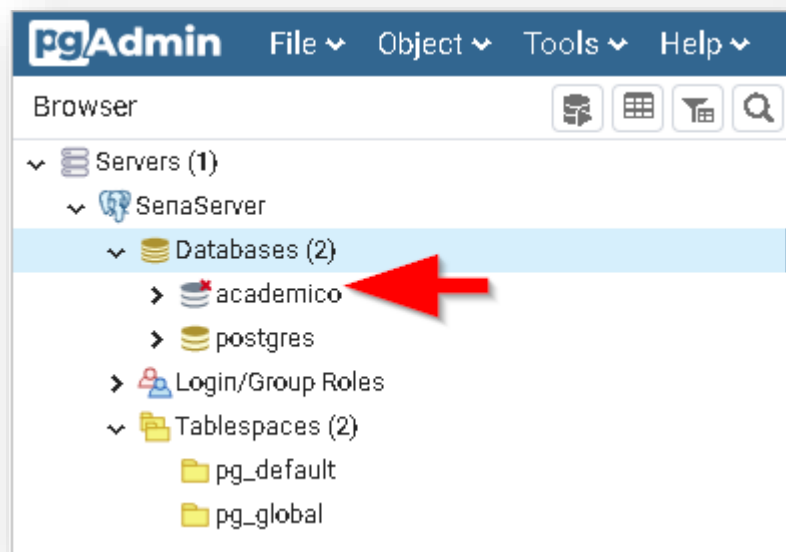
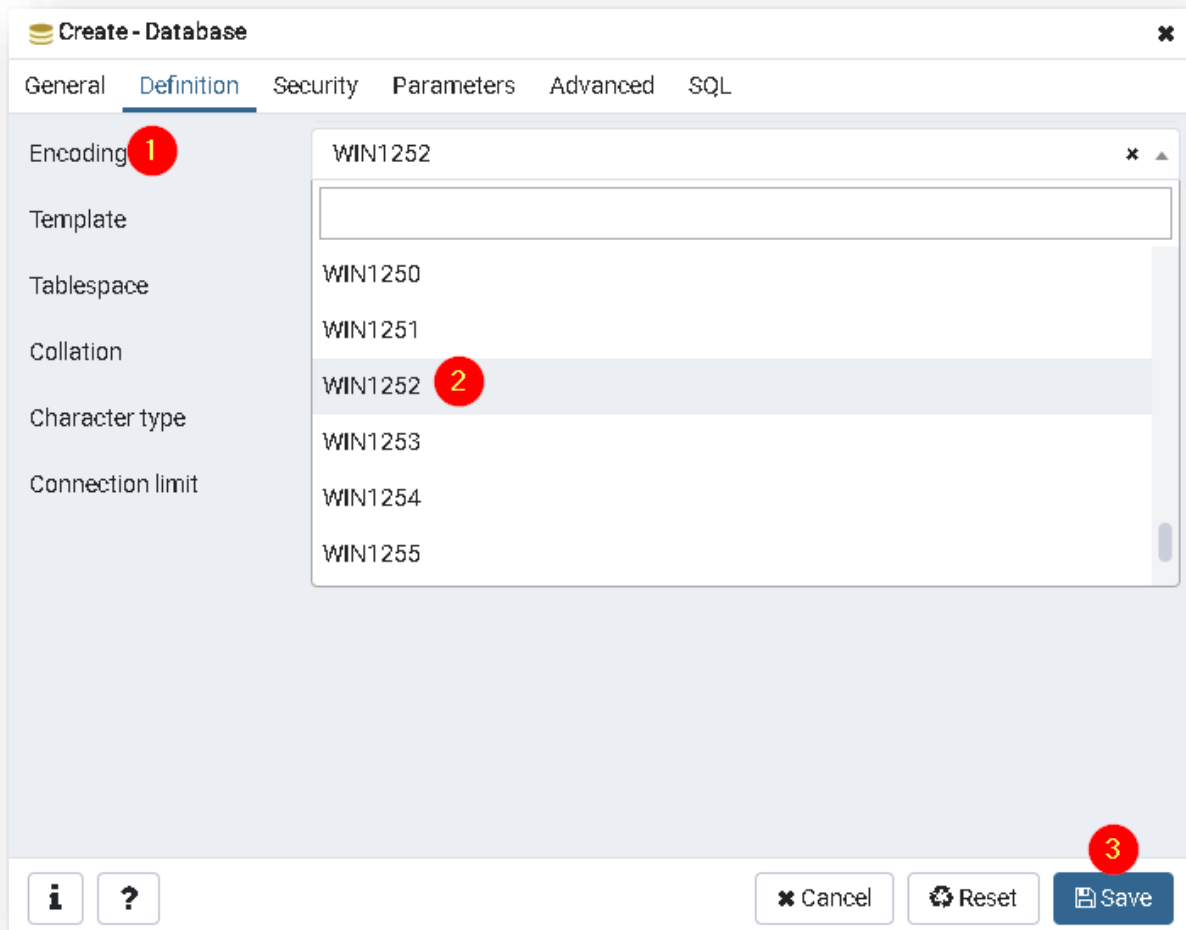
Owner

postgres

Comment

i?

CancelResetSave



CREANDO EL MODELO DE DATOS

```
from django.db import models

# Create your models here.
VACTIVO=(
    (1,"ACTIVO"),
    (2,"INACTIVO")
)

class COORDINACION(models.Model):
    nom=models.TextField(verbose_name="Nombre de la coordinación")
    def __str__(self):
        return self.nom

class AREA(models.Model):
    nom=models.CharField(max_length=4,verbose_name="Nombre del area")
    descripcion=models.TextField()
    idcoordina=models.ForeignKey(COORDINACION,null=True,blank=True,on_delete
=models.CASCADE)
    def __str__(self):
        return self.nom

class TITULACION(models.Model):
    cod=models.IntegerField(verbose_name="codigo de la titulacion")
    nom=models.TextField()
    idarea=models.ForeignKey(AREA,null=True,blank=True,on_delete=models.CASC
ADE)

class FICHA(models.Model):
    nom=models.TextField()
    estado=models.IntegerField(default=0)
    idtitulacion=models.ForeignKey(TITULACION,null=True,blank=True,on_delete
=models.CASCADE)

class NCL(models.Model):
    cod=models.TextField()
    nom=models.TextField()
    des=models.TextField()
    idtitulacion=models.ForeignKey(TITULACION,null=True,blank=True,on_delete
=models.CASCADE)

class RAP(models.Model):
    cod=models.TextField()
    nom=models.TextField()
    idncl=models.ForeignKey(NCL,null=True,blank=True,on_delete=models.CASCAD
E)

class ACTIVIDAD(models.Model):
    nom=models.TextField()
```

```
estado=models.IntegerField(default=1,choices=VACTIVO)
horas=models.IntegerField(default=0)
def __str__(self):
    return self.nom
class RAP_ACTIVIDAD(models.Model):
    idrap=models.ForeignKey(RAP,null=True,blank=True,on_delete=models.CASCADE)
    idactividad=models.ForeignKey(ACTIVIDAD,null=True,blank=True,on_delete=models.CASCADE)
class FICHAS_ACTIVIDAD(models.Model):
    idfichas=models.ForeignKey(FICHA,null=True,blank=True,on_delete=models.CASCADE)
    idactividad=models.ForeignKey(ACTIVIDAD,null=True,blank=True,on_delete=models.CASCADE)
```

CREAR LA MIGRACIÓN

```
(proyecto) C:\Users\JOSE F GALINDO\proyecto\sena>python manage.py makemigrations
```

```
(proyecto) C:\Users\JOSE F GALINDO\proyecto\sena>python manage.py migrate
```