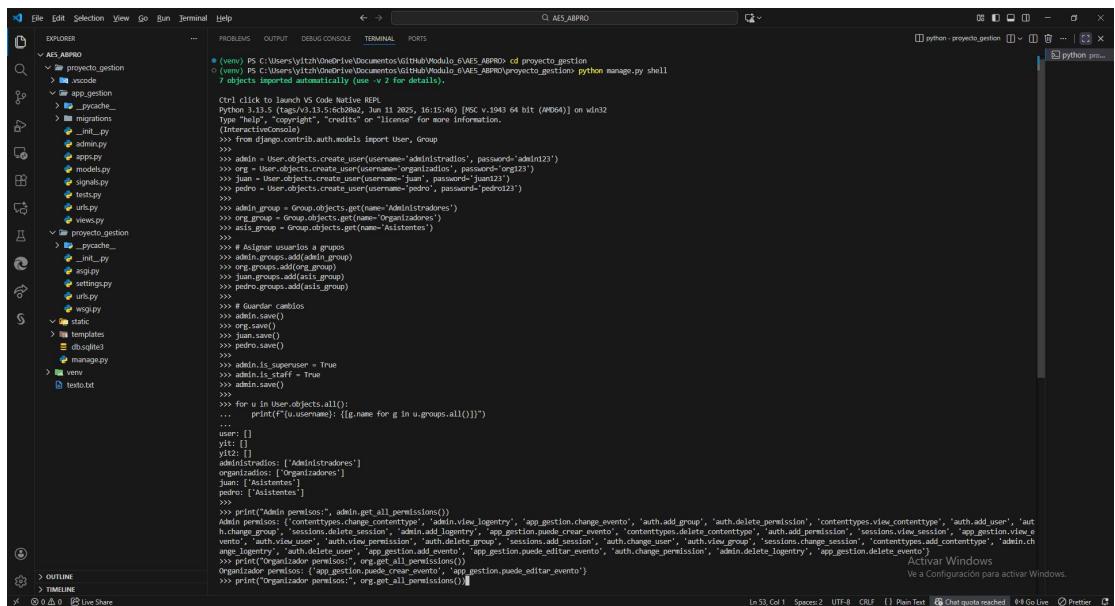


8. -



The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure for "AES_ABPRO". The "projecto_gestion" folder contains "app.gestion" and "app.gestion_". Inside "app.gestion", there are files: __init__.py, admin.py, apps.py, models.py, signals.py, tests.py, urls.py, views.py, and wsgi.py. The "app.gestion_" folder contains __init__.py, migrations, and settings.py.
- TERMINAL:** Displays Python code running in a Windows terminal window titled "AES_ABPRO". The code creates three users ("admin", "juan", "pedro") and two groups ("Administradores" and "Asistentes"). It then adds each user to their respective group and saves the changes.
- STATUS BAR:** Shows file statistics: Line 53, Col 1, Spaces: 2, UTF-8, CRLF, Plain Text, Chat quote required, Go Live, Preview, and Activar Windows.

```
# (venv) PS C:\Users\ytzh\OneDrive\Documentos\GitHub\Modulo_01AES_ABPRO> cd proyecto_gestion
# (venv) PS C:\Users\ytzh\OneDrive\Documentos\GitHub\Modulo_01AES_ABPRO\proyecto_gestion> python manage.py shell
7 objects imported automatically (use -v 2 for details).

Ctrl click to launch VS Code Native RDP
Python 3.13.5 (tags/v3.13.5:fc2b02, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.

>>> from django.contrib.auth.models import User, Group
>>>
>>> admin = User.objects.create_user(username='administrador', password='admin123')
>>> juan = User.objects.create_user(username='juan', password='juan123')
>>> pedro = User.objects.create_user(username='pedro', password='pedro123')
>>>
>>> admin_group = Group.objects.get(name='Administradores')
>>> asis_group = Group.objects.get(name='Asistentes')
>>>
>>> # Asignar usuarios a grupos
>>> admin.groups.add(admin_group)
>>> juan.groups.add(juan_group)
>>> juan.groups.add(asis_group)
>>> pedro.groups.add(asis_group)
>>>
>>> # Guardar cambios
>>> admin.save()
>>> juan.save()
>>> pedro.save()
>>>
>>> admin.is_superuser = True
>>> admin.is_staff = True
>>> admin.save()
>>>
>>> for u in User.objects.all():
...     print(f'{u.username}: {[g.name for g in u.groups.all()]}')
...
user: []
yit: []
admin: ['Administradores']
juan: ['Asistentes']
pedro: ['Asistentes']
>>> print("Admin permisos:", admin.get_all_permissions())
Admin permisos: ['ContentType.change_contenttype', 'admin.view_logentry', 'app_gestion.change_evento', 'auth.add_group', 'auth.delete_permission', 'contenttypes.view_contenttype', 'auth.add_user', 'auth.change_group', 'sessions.delete_session', 'admin.add_logentry', 'app_gestion.puede_crear_evento', 'contenttypes.delete_contenttype', 'auth.add_permission', 'sessions.view_session', 'app_gestion.view_evento', 'auth.view_permission', 'auth.delete_group', 'sessions.add_session', 'auth.change_user', 'auth.view_group', 'sessions.change_session', 'contenttypes.add_contenttype', 'admin.change_logentry', 'auth.add_logentry', 'app_gestion.puede_editar_evento', 'auth.change_permission', 'admin.delete_logentry', 'app_gestion.delete_evento']
>>> print("Organizador permisos:", org.get_all_permissions())
Organizador permisos: ['app_gestion.puede_crear_evento', 'app_gestion.puede_editar_evento']
>>> print("Asistente permisos:", org.get_all_permissions())
```