Django 01

Programación Web 2

Do I need to install pip?

“pip is the package installer for Python. You can use pip to install packages from the Python Package Index and other indexes.”

“pip is already installed if you are using Python 2 >=2.7.9 or Python 3 >=3.4 downloaded from python.org or if you are working in a Virtual Environment created by virtualenv or pyvenv. Just make sure to upgrade pip.”

https://packaging.python.org/tutorials/installing-packages/

Virtualenv

https://virtualenv.pypa.io/en/latest/

“The basic problem being addressed is one of dependencies and versions, and indirectly permissions. Imagine you have an application that needs version 1 of LibFoo, but another application requires version 2. How can you use both these libraries? If you install everything into your host python (e.g. python3.8) it’s easy to end up in a situation where two packages have conflicting requirements.”

```
pip3 install virtualenv
```

https://virtualenv.pypa.io/en/latest/installation.html

https://stackoverflow.com/questions/41573587/what-is-the-difference-between-venv-pyvenv-pyenv-virtualenv-virtualenvwrappe

Usando virtualenv

https://virtualenv.pypa.io/en/latest/user_guide.html

- virtualenv -p python3 **destino**
  - Crea un entorno virtual para trabajar con python3
  - Debe ejecutarse dentro de un directorio de trabajo
- source **destino**/bin/activate (para desactivarlo: deactivate)
  - Activa el entorno virtual de desarrollo
  - Note que cambia el prompt del shell
  - Tendrá que repetir este paso cada vez que inicie una nueva terminal
  - Todo lo que se instale será local y no afectará a la instalación de python en el computador
- pip freeze
  - Muestra las bibliotecas instaladas en el entorno activo actual
- pip install Django
  - Instala Django
  - Note el resultado de freeze ahora

Why Django?

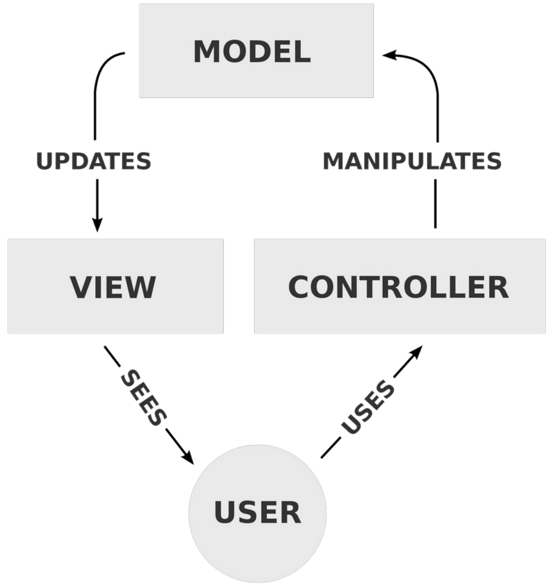
“With Django, you can take Web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.”

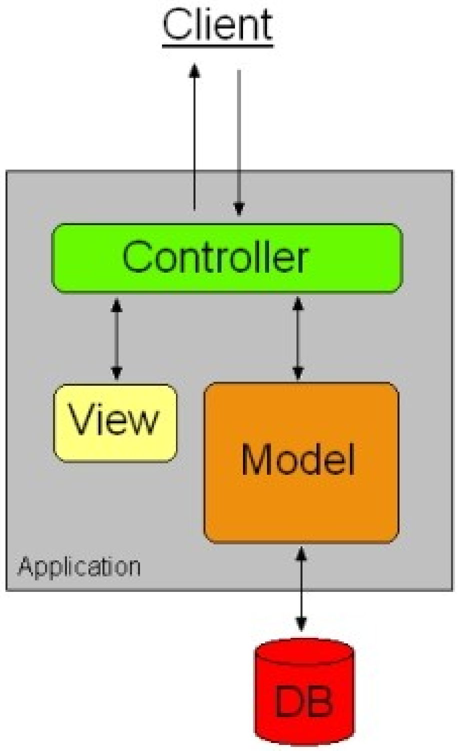
https://www.djangoproject.com/start/overview/#:~:text=With%20Django%2C%20you%20can%20take,It's%20free%20and%20open%20source.

“Because Django was developed in a fast-paced newsroom environment, it was designed to make common Webdevelopment tasks fast and easy.”

https://docs.djangoproject.com/es/3.2/intro/overview/

Model View Controller





Creando un primer proyecto en Django

- Es importante crear un ambiente virtual dentro de un directorio de trabajo
  - mkdir src; cd src
  - source ../bin/activate
  - django-admin **startproject** *listaContactos* **.**
    - Crea un nuevo proyecto llamado “listaContactos” con la configuración básica
- vim *listaContactos/settings.py*
  - Es importante actualizar la información de la configuración
  - LANGUAGE_CODE = 'es'
  - TIME_ZONE = 'America/Lima'
- python **manage.py** runserver

https://docs.djangoproject.com/es/3.2/intro/tutorial01/#creating-a-project

Migraciones

Migrations are Django’s way of propagating changes you make to your models (adding a field, deleting a model, etc.) into your database schema. They’re designed to be mostly automatic, but you’ll need to know when to make migrations, when to run them, and the common problems you might run into.

- python **manage.py** migrate
  - Actualiza los cambios hechos en el proyecto
  - Sincroniza el trabajo con la base de datos elegida
  - No hay que diseñar tablas o relaciones
  - Cambiar de SGBD es muy simple

https://docs.djangoproject.com/es/3.2/topics/migrations/

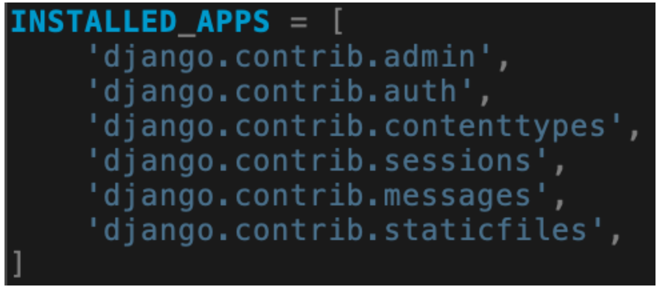
https://docs.djangoproject.com/es/3.2/ref/applications/

Projects and applications

The term **project** describes a Django web application. The project Python package is defined primarily by a settings module, but it usually contains other things.

A **project's root directory** (the one that contains manage.py) is usually the container for all of a project's applications which aren't installed separately.

The term **application (app)** describes a Python package that provides some set of features. Applications may be reused in various projects.



```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Creación de Apps en Django

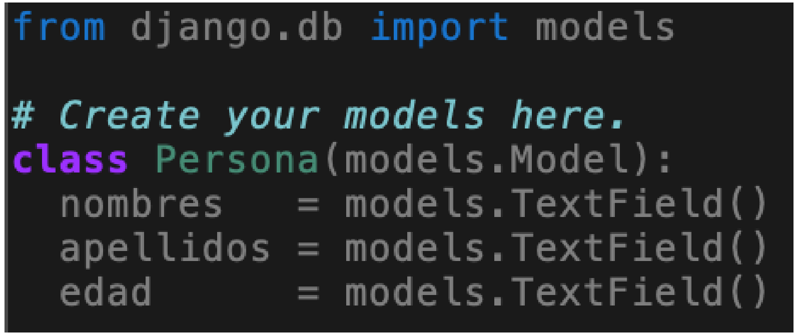
Creates a Django app directory structure for the given app name in the current directory or the given destination.

```
django-admin startapp personas
personas/
├── __init__.py
├── admin.py
├── apps.py
├── migrations
│   └── __init__.py
├── models.py
├── tests.py
└── views.py
```

https://docs.djangoproject.com/es/3.2/ref/django-admin/#startapp

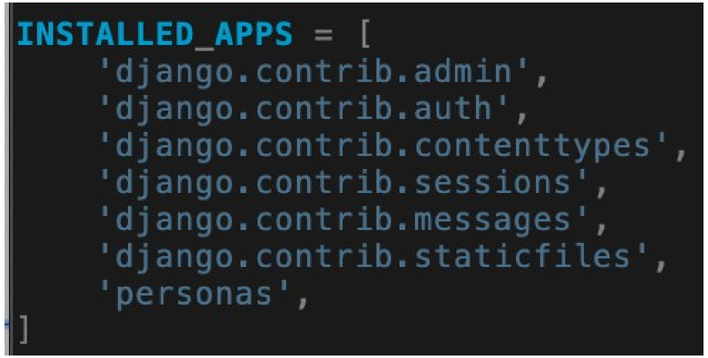
Cree sus propias APPs, no tenga temor de experimentar https://docs.djangoproject.com/es/3.2/intro/tutorial02/#creating-models

vim personas/models.py



```
from django.db import models

# Create your models here.
class Persona(models.Model):
  nombres   = models.TextField()
  apellidos = models.TextField()
  edad      = models.TextField()
```



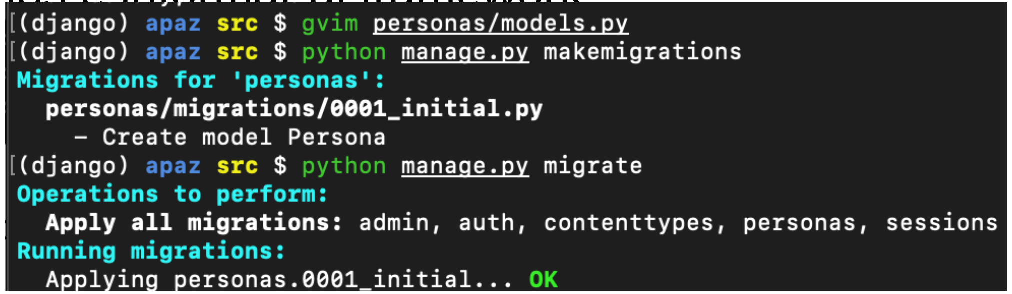
```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'personas',
]
```

Hay que actualizar los cambios hechos e informar al framework

python **manage.py** makemigrations

python **manage.py** migrate

¿Qué archivos se modificaron al hacer makemigrations y migrate?



```
(django) apaz src $ gvim personas/models.py
(django) apaz src $ python manage.py makemigrations
Migrations for 'personas':
  personas/migrations/0001_initial.py
    - Create model Persona
(django) apaz src $ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, personas, sessions
Running migrations:
  Applying personas.0001_initial... OK
```

https://docs.djangoproject.com/es/3.2/intro/tutorial02/#activating-models

The Django /admin site

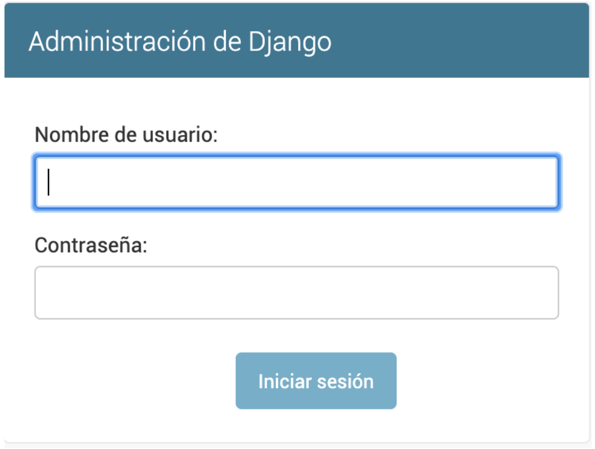
One of the most powerful parts of Django is the automatic admin interface. It reads metadata from your models to provide a quick, model-centric interface where trusted users can manage content on your site. **The admin's recommended use is limited to an organization's internal management tool. It's not intended for building your entire front end around.**

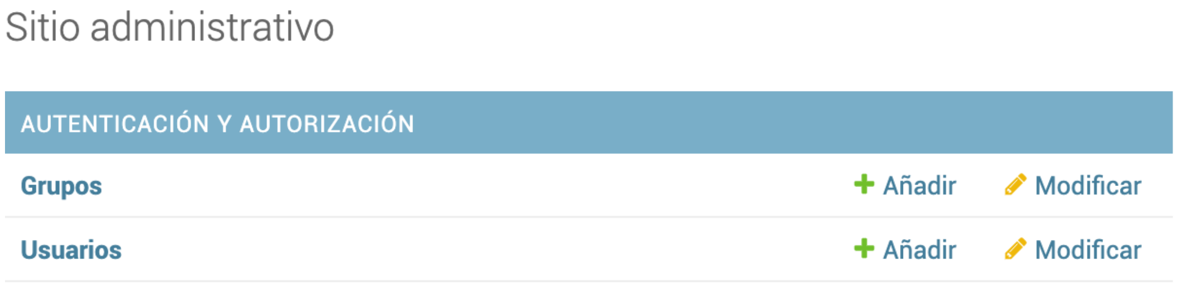
The admin has many hooks for customization, but beware of trying to use those hooks exclusively. If you need to provide a more process-centric interface that abstracts away the implementation details of database tables and fields, then it's probably time to write your own views.

https://docs.djangoproject.com/es/3.2/ref/contrib/admin/#:~:text=One%20of%20the%20most%20powerful,an%20organization's%20internal%20management%20tool.

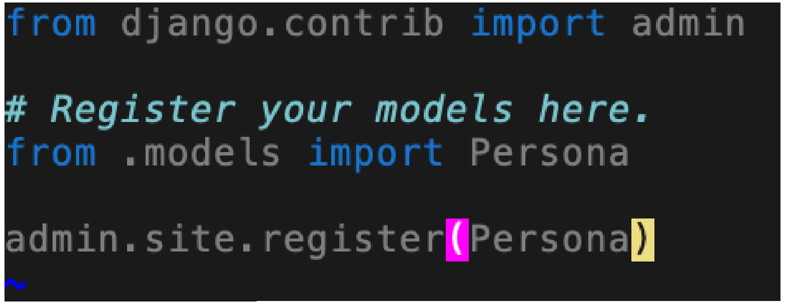
Creando nuevos usuarios

- ~~django-admin createsuperuser~~
- python **manage.py** createsuperuser
    - Crea un nuevo usuario con su contraseña con permisos de administrador





vim personas/admin.py

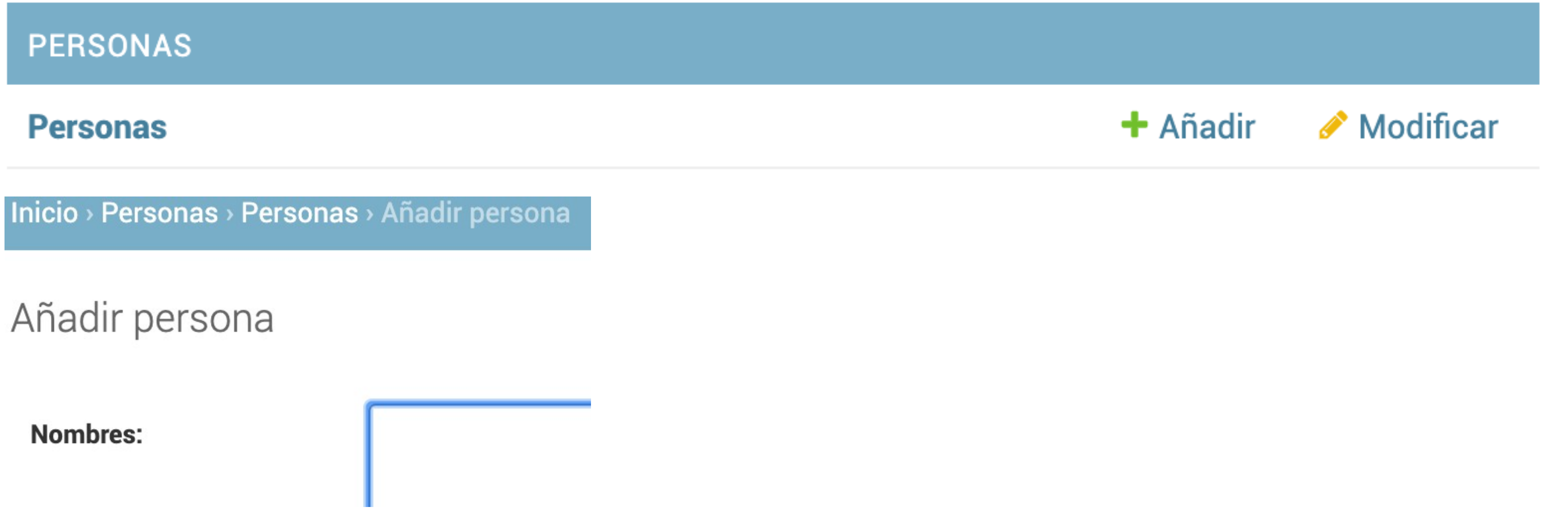


```
from django.contrib import admin

# Register your models here.
from .models import Persona

admin.site.register(Persona)
~
```

admin/



PERSONAS

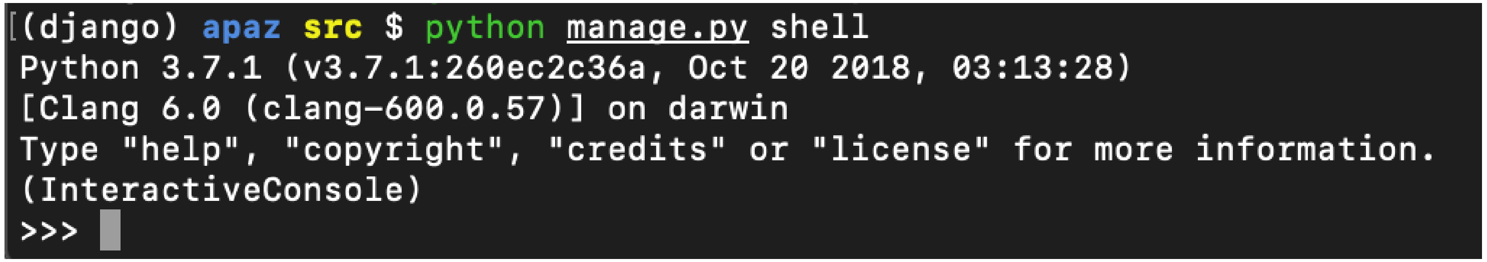
Personas ✚ Añadir ✎ Modificar

Inicio › Personas › Personas › Añadir persona

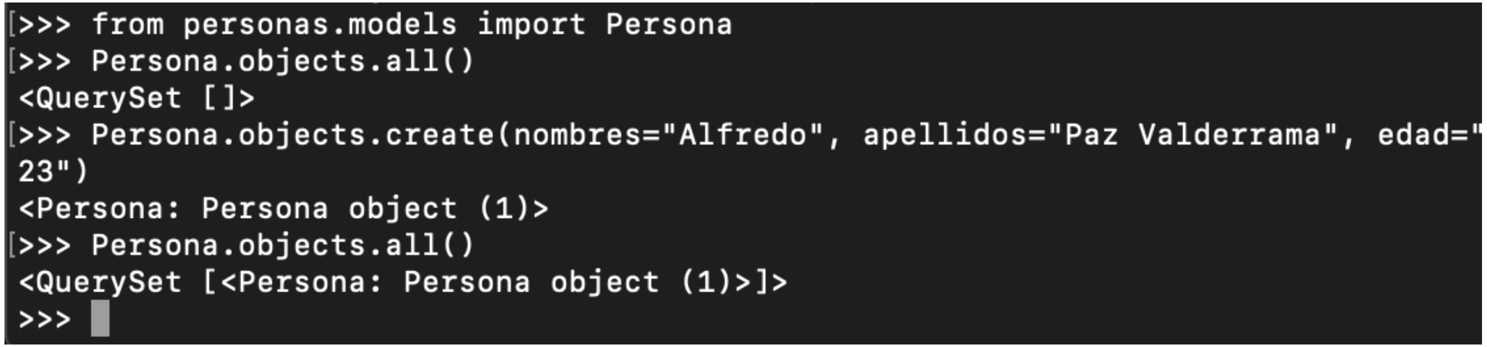
Añadir persona

Nombres:

python **manage.py** shell



```
[(django) apaz src $ python manage.py shell
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 03:13:28)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> 
```



```
[>>> from personas.models import Persona
[>>> Persona.objects.all()
<QuerySet []>
[>>> Persona.objects.create(nombres="Alfredo", apellidos="Paz Valderrama", edad="
23")
<Persona: Persona object (1)>
[>>> Persona.objects.all()
<QuerySet [<Persona: Persona object (1)>]>
>>> 
```

¿Qué archivos se modificaron al agregar personas?

Field types

https://docs.djangoproject.com/en/3.2/ref/models/fields/#field-types

- CharField
  - class CharField(max_length=None, **options)[source]
  - A string field, for small- to large-sized strings.
  - For large amounts of text, use TextField.
- TextField
  - class TextField(**options)[source]
  - A large text field. The default form widget for this field is a Textarea.
- IntegerField
  - class IntegerField(**options)[source]
  - An integer. Values from -2147483648 to 2147483647 are safe in all databases supported by Django.