

**Ayudantia:**  
**Pasos para realizar la aplicación**

Para ver que versión de django tienen instalada:

**python -m django --version**

Comando para crear un proyecto:

**django-admin startproject Proyecto**

**Nota:** evitar darle nombres a sus proyectos que sean iguales a los de otros componentes integrados de Python o Django. En particular, esto quiere decir que debe evitar usar nombres como django (que entrará en conflicto con Django mismo) o “test” (que entrará en conflicto con un paquete interno de Python).

Comando para correr el proyecto, antes del correr el proyecto meterse en la carpeta con el comando:

**cd Proyecto**

Luego:

**python manage.py runserver**

Comando para crear una aplicación dentro del proyecto:

**python manage.py startapp Aplicacion**

Configuración de la base de datos.

Abrir el archivo dentro de Proyecto “settings.py”.

Luego cambiar esto:

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

Por esto:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'HOST': 'localhost',  
        'NAME': 'app_fundamentos',  
        'USER': 'root',  
        'PASSWORD': 'root',  
    }  
}
```

```
}  
}
```

Crear la base de datos con el mismo nombre en phpmyadmin.

Luego:

Instalar mysqlclient:

**sudo pip install mysqlclient**

Para crear las tablas escriba el comando:

**python manage.py migrate**

Crear un super usuario para acceder a la administración de la aplicación

**python manage.py createsuperuser**

Luego crearemos los modelos:

Edite el archivo de la Aplicacion “models.py”

Para crear un modelo (las tablas), haremos el ejemplo con la tabla asignatura

Agregar al archivo:

```
class Asignatura(models.Model):  
    nombre=models.CharField(max_length=200)  
    codigo=models.CharField(max_length=20)
```

Luego en el archivo de Proyecto “settings.py”

Editar y agregar lo siguiente para activar la aplicación:

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

Luego ingresar el comando para crear las migraciones (se creará una por cada modelo)

**python manage.py makemigrations**

Despues ingresar el comando

**python manage.py migrate**

Este comando es para que se creen las tablas correspondientes en la base de datos

Para las otras tablas hacer lo mismo, lo modelos primero y luego correr el comando `python manage.py makemigrations`, todos los modelos se hacen en el mismo archivo y al final el comando `python manage.py migrate`

En el caso de que uno se equivoquen en hacer una migración tienen que eliminar el contenido en la carpeta “migrations” de su aplicación y volver a correr el comando **python manage.py makemigrations**.

Luego de haber creado los modelos, ingresar al archivo `admin.py` de la aplicación y editar los archivos

```
from .models import *  
admin.site.register(Asignatura)  
admin.site.register(Academico)  
admin.site.register(Carga)
```

y meterse como super usuario.

Las vistas y controladores serán explicadas en la ayudantia.