

**Ayudantía:**  
**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',  
    }  
}
```

```
}  
}
```

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**

Después 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 migracion 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.

Como crear una vista

Crear archivo dentro de la aplicación con el nombre “`urls.py`”

Abra el archivo `view.py` de la aplicación

Inserte el siguiente código

```
from __future__ import unicode_literals
```

```
from django.shortcuts import render
```

```
# Create your views here.
```

```
from django.http import HttpResponse
```

```
def principal(request):  
    return HttpResponse("Vista principal")
```

Luego agregue la ruta de esta vista en la url de la aplicación, o sea en el archivo “`urls.py`”

El nombre de la url debe es la misma que el nombre de la función definida en la vista

·  
Creando ruta

```
from django.conf.urls import url
```

```
from . import views
```

```
urlpatterns = [  
    url(r'^$', views.principal, name='principal'),  
]  
]
```

Luego creamos el url en el proyecto, se ingresa el urls.py del proyecto y se agrega la ruta

```
from django.conf.urls import include, url  
from django.contrib import admin
```

```
urlpatterns = [  
    url(r'^principal/', include('Aplicacion.urls')),  
    url(r'^admin/', admin.site.urls),  
]  
]
```

Listo han creado su primera vista que solo imprime algo en pantalla

Vamos a usar el sistema de plantillas de Django para separar el diseño de Python mediante la creación de una plantilla que la vista pueda utilizar.

Primero, cree una carpeta llamada “templates” en su aplicación “polls”. Django buscará las plantillas allí.

Dentro del carpeta **templates** que acabas de crear, crea otra carpeta llamada Aplicacion` y dentro de este crea un archivo llamado `` principal.html.

Insertar el siguiente código dentro de principal.html

```
<!DOCTYPE html>  
  
{% load staticfiles %}  
  
<html>  
  
<head>  
  
    <title>APP</title>  
  
</head>  
  
<body>  
  
<h1>Academicos!</h1>  
  
<table style="width:100%">  
  
    <thead>
```

```

<th align="left">Rut</th>

<th align="left">Nombre</th>

<th align="left">Apellido</th>
</thead>

<tbody>

    {% for academico in var% }

    <tr>

        <td>{{ academico.rut }}</td><td>{{ academico.nombre }}</td><td>{{ academico.apellido }}</td>

    </tr>

    {% endfor% }

</tbody>

</table>

</body>

</html>

```

Ahora vamos a actualizar nuestra vista **principal** en el archivo **views.py** (dentro de la carpeta de su aplicación) para usar la plantilla:

```
# -*- coding: utf-8 -*-
```

```
from __future__ import unicode_literals
```

```
from django.shortcuts import render
```

```
# Create your views here.
```

```
from django.http import HttpResponse
```

```
from .models import Academico
```

```
def principal(request):
```

```
    return render(request, 'Aplicacion/principal.html',{'var':Academico.objects.all()})
```

Listo esta creada la vista de la lista de académicos.

Ahora crearemos la vista y el controlador para agregar un académico.

Primero deben entrar a la carpeta templates y luego a la de Aplicacion que está dentro de templates y luego crear un archivo con el nombre “nuevo\_academico.html”

Luego agregar a el archivo urls.py de la aplicación la ruta

```
from django.conf.urls import url
```

```
from . import views
```

```
urlpatterns = [  
    url(r'^$', views.principal, name='principal'),  
    url(r'^academicos/nuevo/$', views.nuevo_academico, name='nuevo_academico'),
```

Después el archivo nuevo\_academico.html se creara el formulario para agregar un academico.

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
    <title>Nuevo académico</title>  
</head>  
  
<body>  
<h1>Formulario nuevo académico</h1>  
{% if form.subject.errors %}  
    <ol>  
        {% for error in form.subject.errors %}  
            <li><strong>{{ error|escape }}</strong></li>  
        {% endfor %}  
    </ol>  
{% endif %}  
<form method="POST">  
    {% csrf_token %}
```

```
<label for="nombre">Nombre</label>
```

```
<input type="text" name="nombre" placeholder="Ej: Juan">
```

```
<label for="apellido">Apellido</label>
```

```
<input type="text" name="apellido" placeholder="Ej: Pérez">
```

```
<label for="rut">RUT</label>
```

```
<input type="text" name="rut">
```

```
<button>Registrar</button>
```

```
</form>
```

```
</body>
```

```
</html>
```

Después se creara el controlador en view.py de la aplicación

```
# -*- coding: utf-8 -*-
```

```
from __future__ import unicode_literals
```

```
from django.shortcuts import render
```

```
# Create your views here.
```

```
from django.http import HttpResponse
```

```
from .models import Academico
```

```
from django import forms
```

```
def principal(request):
```

```
    return render(request, 'Aplicacion/principal.html', {'var':Academico.objects.all()})
```

```

def nuevo_academico(request):
    if(request.method=='POST'):
        form=Form_nuevo_academico(request.POST)
        if form.is_valid():
            datos=form.cleaned_data
            nuevo=Academico()
            nuevo.rut=datos.get('rut')
            nuevo.nombre=datos.get('nombre')
            nuevo.apellido=datos.get('apellido')
            nuevo.save()
            return principal(request)
        else:
            return HttpResponseRedirect("datos invalidos");
    elif(request.method=='GET'):
        return render(request, 'Aplicacion/nuevo_academico.html')
    else:
        return HttpResponseRedirect('metodo no soportado')

```

```

class Form_nuevo_academico(forms.Form): #Note that it is not inheriting from forms.ModelForm
    nombre = forms.CharField(max_length=200)
    apellido = forms.CharField(max_length=200)
    #apellido = forms.CharField(max_length=200,required=False)
    rut = forms.CharField(max_length=20)

```

#All my attributes here

Por último se le agrega un link a la página principal para agregar un nuevo académico

```
<!DOCTYPE html>
```

```
{% load staticfiles %}
```

```
<html>
```



```

<head>

    <title>APP</title>

</head>

<body>

<h1>Académicos!</h1>

<table style="width:100%">

    <thead>

        <th align="left">Rut</th>

        <th align="left">Nombre</th>

        <th align="left">Apellido</th>

    </thead>

    <tbody>

        { %for academico in var% }

        <tr>

            <td>{{ academico.rut }}</td><td>{{ academico.nombre }}</td><td>{{ academico.apellido }}</td>

        </tr>

        { %endfor% }

    </tbody>

</table>

<a href="{ % url 'nuevo_academico' % }">Nuevo</a>

</body>

</html>

```

Listo ya está creada la función de agregar un académico