



unl

Universidad
Nacional
de Loja





Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera Computación

Desarrollo Basado en Plataformas

Profesor: Ing. Roberth Figueroa Díaz M.Sc.

roberth.figueroa@unl.edu.ec



Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

Agenda

1. Tecnologías del lado del Servidor o Backend

2. Instalando Framework Django

- Instalar Django
- Identificar comandos claves
- Realizar un ejemplo con Django
- Conclusiones y recomendaciones en Django

INSTALACIÓN DJANGO

3.2.3

Pasos a seguir

AMBIENTE DESARROLLO

1. Instalando PIP en Python
 - Usar comando → **sudo easy_install pip**
2. Instalando virtual environment
 - Usar comando → **sudo pip install virtualenv**
3. Crear el entorno virtual deseado
 - Usar comando → **virtualenv envDjango**
 - O definir la versión de python al entorno → **virtualenv -p python3 .**
 - Se crean carpeta **bin**, **lib** con archivos Python necesarios
4. Instalar Django
 - Comando para activar virtualenv → **source bin/activate**
 - Comando para instalar → **pip install Django==3.2.3**
 - Comando para actualizar la version django → **pip install django -- upgrade**
 - Comando para ver la versión de django instalada → **pip freeze**
5. Crear la aplicación **mi_proyecto** dentro del environment
 - Comando → **django-admin startproject mi_proyecto**
6. Se genera la base de datos (SQLite):
 - Comando → **python manage.py migrate**
7. Se ejecuta el servidor de aplicaciones web:
 - Comando → **python manage.py runserver**

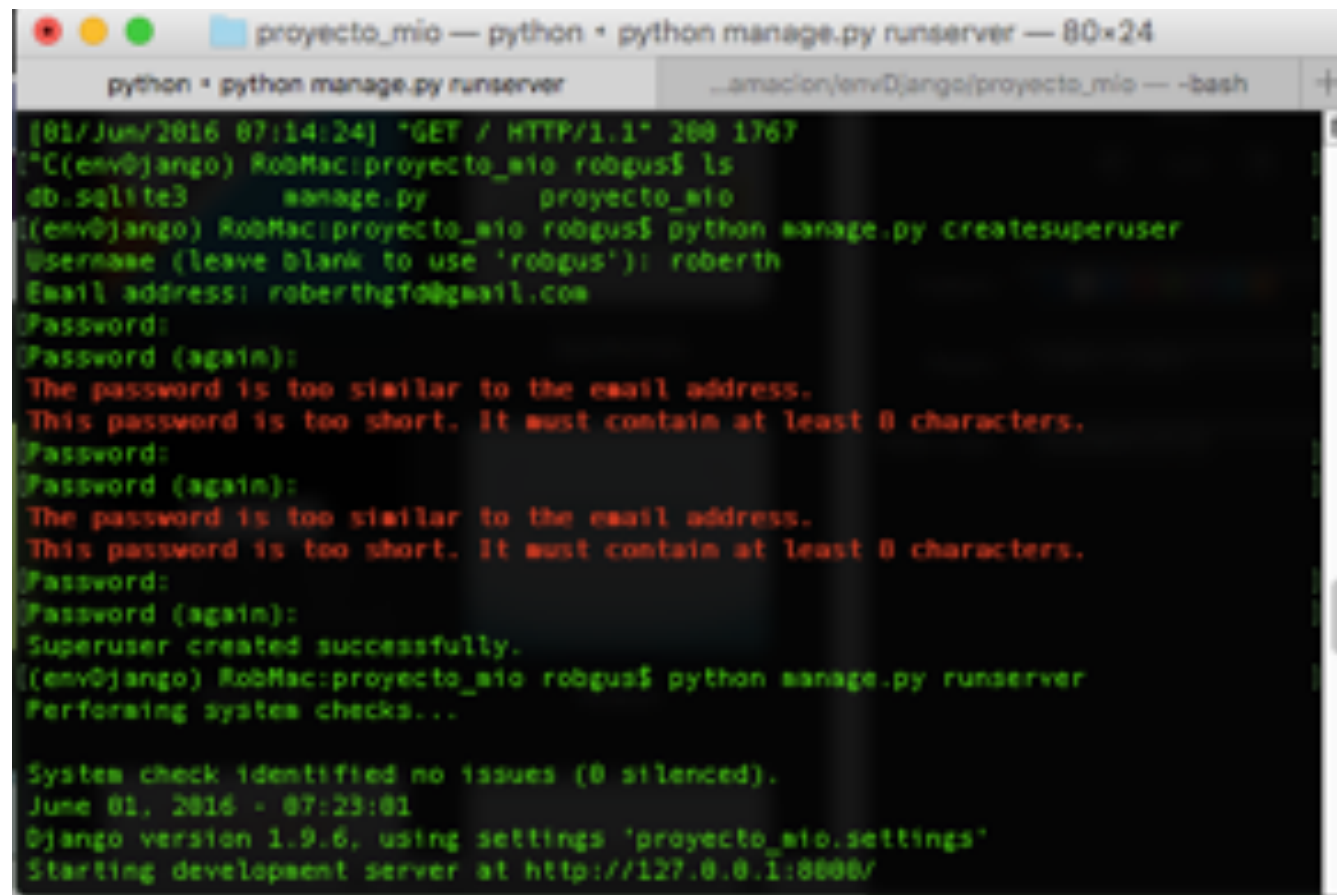
Proceso

Instalación de Django 3.2.3

```
macrob:Developer roberth$ ls
PythonLearn  envBorrar    tryDjango
macrob:Developer roberth$ rm -rf envBorrar/
macrob:Developer roberth$ ls
PythonLearn  tryDjango
macrob:Developer roberth$ cd tryDjango/
macrob:tryDjango roberth$ ls
envDjango
macrob:tryDjango roberth$ pip install Django==3.2.3
Collecting Django==3.2.3
  Downloading Django-3.2.3-py3-none-any.whl (7.9 MB)
    |#####| 7.9 MB 3.2 MB/s
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.3.4-py3-none-any.whl (22 kB)
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.1-py3-none-any.whl (42 kB)
    |#####| 42 kB 1.2 MB/s
Collecting pytz
  Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
    |#####| 510 kB 3.0 MB/s
Installing collected packages: sqlparse, pytz, asgiref, Django
Successfully installed Django-3.2.3 asgiref-3.3.4 pytz-2021.1 sqlparse-0.4.1
macrob:tryDjango roberth$
```

Crear el administrador Django

- Crear el usuario
 - Usar comando → **python manage.py createsuperuser**



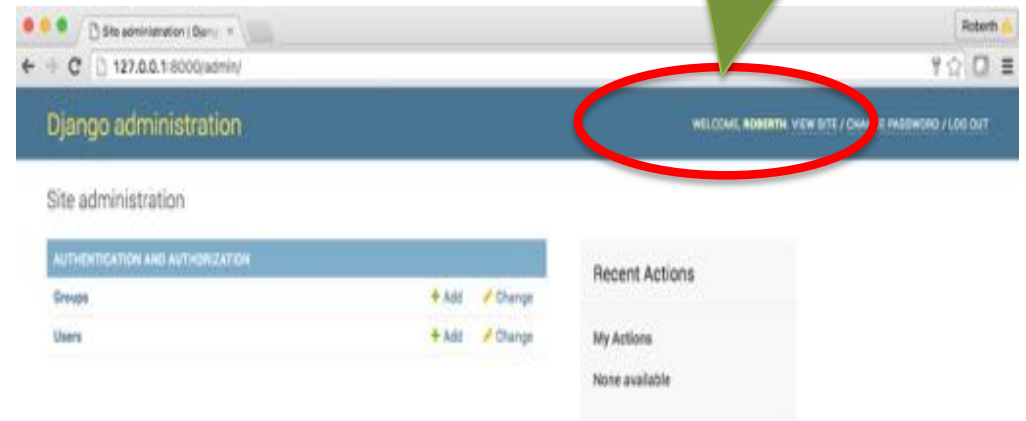
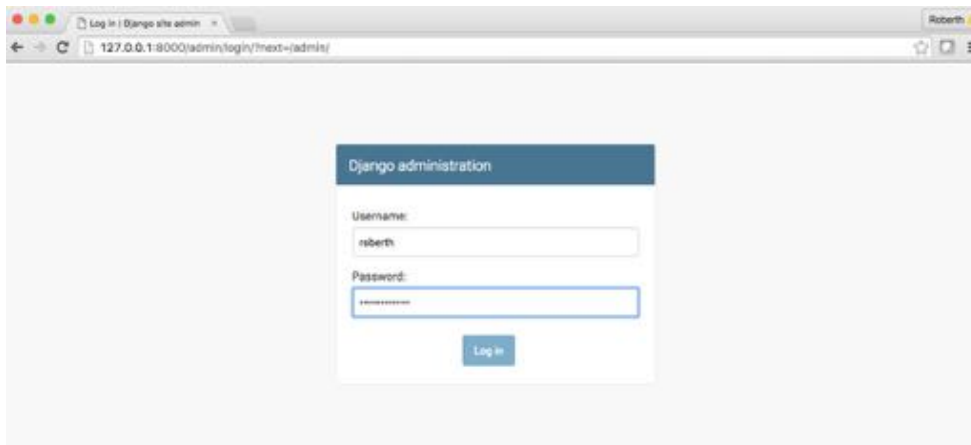
```
projecto_mio — python * python manage.py runserver — 80x24
python * python manage.py runserver  ...mac/venvDjango/projecto_mio — -bash

[01/Jun/2016 07:14:24] "GET / HTTP/1.1" 200 1767
((envDjango) RobMac:projecto_mio robgu$ ls
db.sqlite3      manage.py      proyecto_mio
((envDjango) RobMac:projecto_mio robgu$ python manage.py createsuperuser
Username (leave blank to use 'robgu'): robertg
Email address: robertgfo@gmail.com
Password:
Password (again):
The password is too similar to the email address.
This password is too short. It must contain at least 8 characters.
Password:
Password (again):
The password is too similar to the email address.
This password is too short. It must contain at least 8 characters.
Password:
Password (again):
Superuser created successfully.
((envDjango) RobMac:projecto_mio robgu$ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
June 01, 2016 - 07:23:01
Django version 1.9.6, using settings 'projecto_mio.settings'
Starting development server at http://127.0.0.1:8000/
```

Crear el administrador Django

- Se ejecuta el servidor de aplicaciones :
 - Comando → **python manage.py runserver**
- Ir al navegador a:
 - `http://127.0.0.1:8000/admin/`



Creación de aplicaciones

- Crear el usuario
 - Usar comando → **python manage.py startapp appUno**
 - Agregar la appUno al settings.py → **sudo vim settings.py**

settings.py

```
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30 # Application definition
31
32 INSTALLED_APPS = [
33     'django.contrib.admin',
34     'django.contrib.auth',
35     'django.contrib.contenttypes',
36     'django.contrib.sessions',
37     'django.contrib.messages',
38     'django.contrib.staticfiles',
39     'appUno',
40 ]
41
42
43 MIDDLEWARE_CLASSES = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
47     'django.middleware.csrf.CsrfViewMiddleware',
48     'django.contrib.auth.middleware.AuthenticationMiddleware',
49     'django.contrib.auth.middleware.SessionAuthenticationMiddleware',
50     'django.contrib.messages.middleware.MessageMiddleware',
51     'django.middleware.clickjacking.XFrameOptionsMiddleware',
52 ]
53
54 ROOT_URLCONF = 'proyecto_mio.urls'
55
56 TEMPLATES = [
57     {
58         'BACKEND': 'django.template.backends.django.DjangoTemplates',
59         'DIRS': [],
```

CREACIÓN DE MÁS APLICACIONES EN DJANGO

Creación de Aplicaciones Django

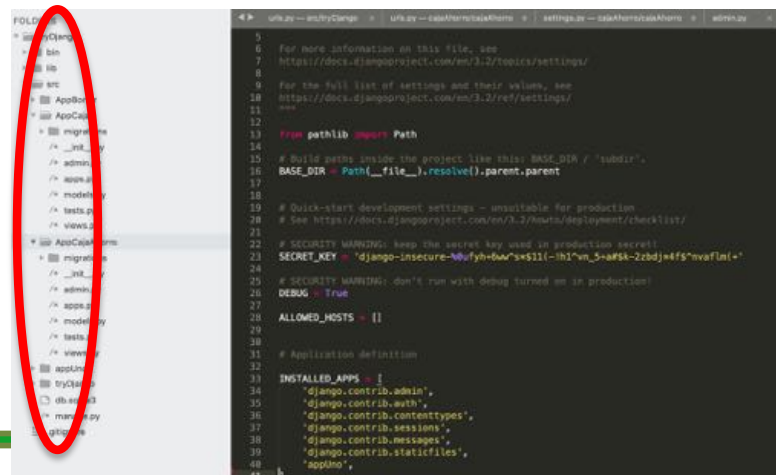
- Ubicarse en el directorio de instalación de Django

En nuestro caso:

→ /Users/roberth/Developer/tryDjango/src

```
src — -bash — 80x24
bin      cajaAhorro  lib      pyvenv.cfg  src
(tryDjango) macrob:tryDjango roberth$ ls
bin      cajaAhorro  lib      pyvenv.cfg  src
(tryDjango) macrob:tryDjango roberth$ cd ..
(tryDjango) macrob:Developer roberth$ ls
PythonLearn  tryDjango
(tryDjango) macrob:Developer roberth$ pwd
/Users/roberth/Developer
(tryDjango) macrob:Developer roberth$ cd tryDjango/
(tryDjango) macrob:tryDjango roberth$ ls
bin      cajaAhorro  lib      pyvenv.cfg  src
(tryDjango) macrob:tryDjango roberth$ cd src/
(tryDjango) macrob:src roberth$ ls
Banco    appUno    db.sqlite3  manage.py  tryDjango
(tryDjango) macrob:src roberth$ python manage.py startapp AppBorrar
(tryDjango) macrob:src roberth$ python manage.py startapp AppCaja
(tryDjango) macrob:src roberth$ python manage.py startapp AppCajaAhorro
(tryDjango) macrob:src roberth$ python manage.py startapp AppCajaAhorro2
(tryDjango) macrob:src roberth$ python manage.py startproject MiProyecto
(tryDjango) macrob:src roberth$ pwd
/Users/roberth/Developer/tryDjango/src
(tryDjango) macrob:src roberth$
```

- Identificar carpetas y archivos principales de cada aplicación



Creación de aplicación con **startapp**

- Crear la aplicación CajaAhorro
 - Usar comando → `python manage.py startapp appCajaAhorro`
 - Agregar la appCajaAhorro al `settings.py` → `sudo vim settings.py`

settings.py

```
31
32 # Application definition
33
34 INSTALLED_APPS = [
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41     'appUno',
42     'CajaAhorro',
43 ]
44
45 MIDDLEWARE = [
46     'django.middleware.security.SecurityMiddleware',
47     'django.contrib.sessions.middleware.SessionMiddleware',
48     'django.middleware.common.CommonMiddleware',
49     'django.middleware.csrf.CsrfViewMiddleware',
50     'django.contrib.auth.middleware.AuthenticationMiddleware',
51     'django.contrib.messages.middleware.MessageMiddleware',
52     'django.middleware.clickjacking.XFrameOptionsMiddleware',
53 ]
54
55 ROOT_URLCONF = 'DemoDjango.urls'
56
57 TEMPLATES = [
58     {
59         'BACKEND': 'django.template.backends.django.DjangoTemplates',
60         'DIRS': [BASE_DIR, "templates"],
61         'APP_DIRS': True,
62         'OPTIONS': {
```

Comando makemigrations

- Codificar el **models.py**, con la clase Contacto:

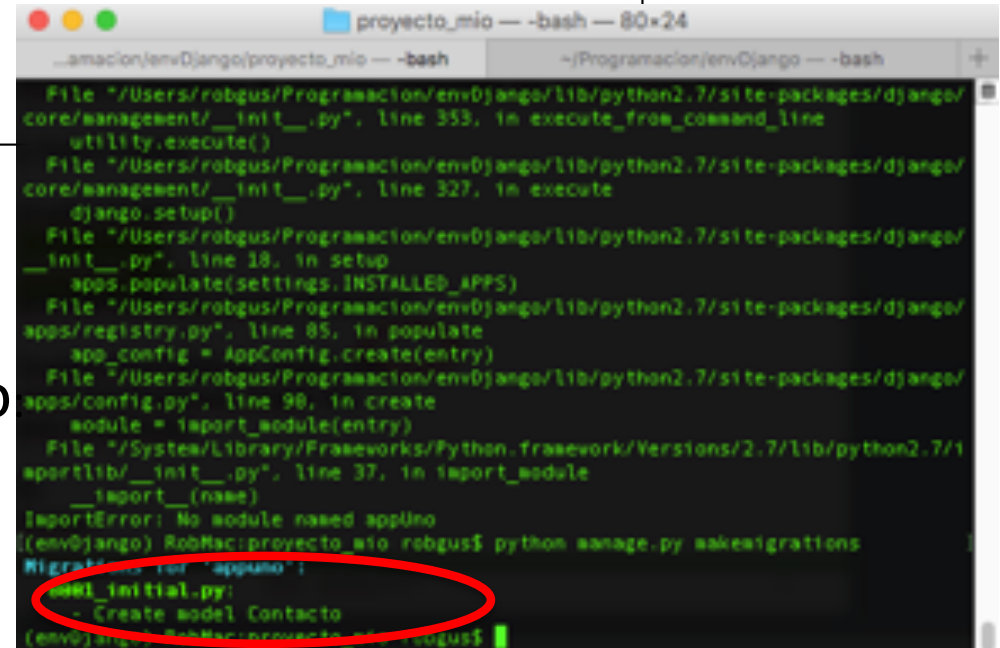
```
from django.db import models

class Contacto(models.Model):

    nombres=models.CharField(max_length=30,blank=True)
    apellidos=models.CharField(max_length=30)
    cedula=models.CharField(max_length=30)
    email=models.EmailField()

def __str__(self):
    return self.email
```

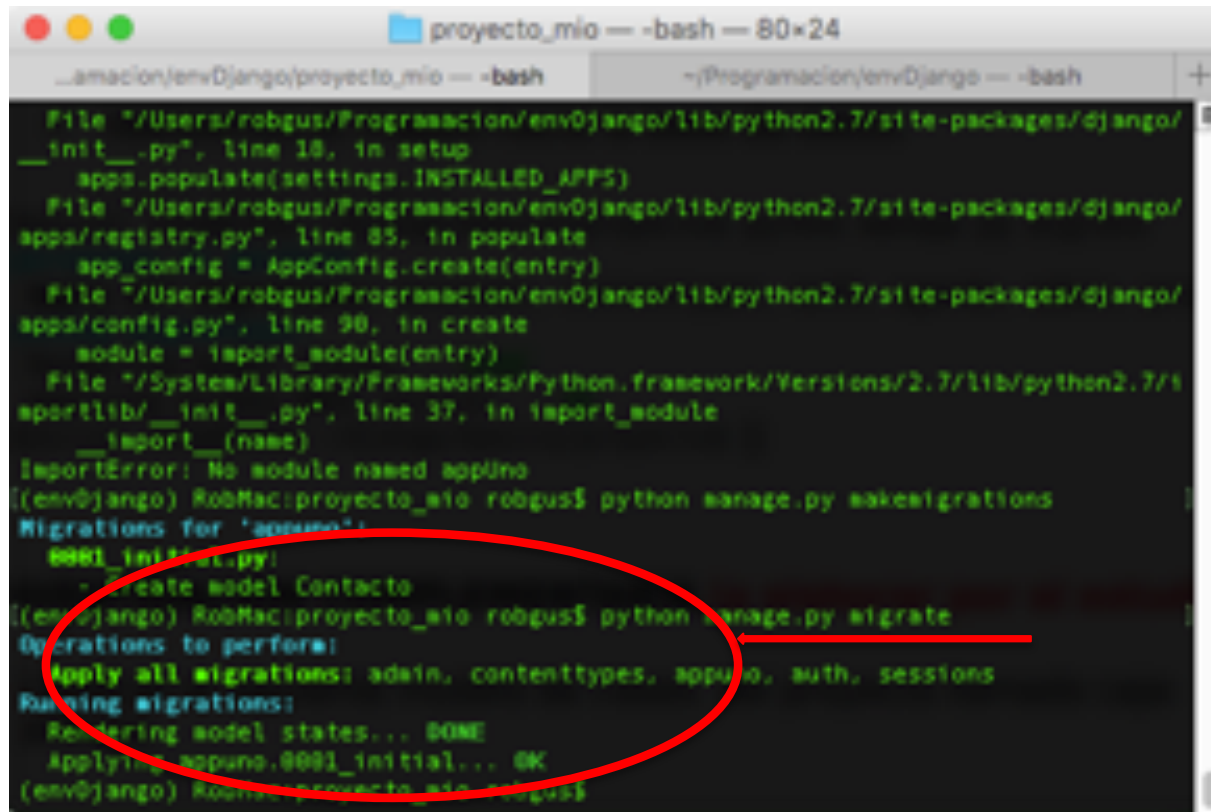
- Aplicar el modelo al proyecto
usar → **python manage.py makemigrations**



```
proyecto_mio --bash -- 80x24
~/Programacion/envDjango/proyecto_mio --bash
~/Programacion/envDjango --bash
File ~/Users/robgus/Programacion/envDjango/lib/python2.7/site-packages/django/
core/management/__init__.py, line 353, in execute_from_command_line
utility.execute()
File ~/Users/robgus/Programacion/envDjango/lib/python2.7/site-packages/django/
core/management/__init__.py, line 327, in execute
django.setup()
File ~/Users/robgus/Programacion/envDjango/lib/python2.7/site-packages/django/
__init__.py, line 18, in setup
apps.populate(settings.INSTALLED_APPS)
File ~/Users/robgus/Programacion/envDjango/lib/python2.7/site-packages/django/
apps/registry.py, line 85, in populate
app_config = AppConfig.create(entry)
File ~/Users/robgus/Programacion/envDjango/lib/python2.7/site-packages/django/
apps/config.py, line 90, in create
module = import_module(entry)
File ~/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/i
mportlib/__init__.py, line 37, in import_module
__import__(name)
ImportError: No module named appuno
((envDjango) RobMac:proyecto_mio robgus$ python manage.py makemigrations
Migrations for 'appuno':
model_initial.py:
- Create model Contacto
((envDjango) RobMac:proyecto_mio robgus$
```

Comando migrate

- Mapear hacia la base de datos el modelo
 - usar → `python manage.py migrate`

A terminal window titled 'proyecto_mio' showing the execution of Django management commands. The user runs 'python manage.py makemigrations', which creates a new migration file '0001_initial.py' for the 'appuno' app. Then, the user runs 'python manage.py migrate', which applies the migration. The output shows the migration being applied successfully. A red circle highlights the 'python manage.py migrate' command and its output.

```
File "/Users/robguis/Programacion/envDjango/lib/python2.7/site-packages/django/
__init__.py", line 18, in setup
    apps.populate(settings.INSTALLED_APPS)
File "/Users/robguis/Programacion/envDjango/lib/python2.7/site-packages/django/
apps/registry.py", line 85, in populate
    app_config = AppConfig.create(entry)
File "/Users/robguis/Programacion/envDjango/lib/python2.7/site-packages/django/
apps/config.py", line 98, in create
    module = import_module(entry)
File "/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/s
importlib/__init__.py", line 37, in import_module
    __import__(name)
ImportError: No module named appuno
(envDjango) RobMac:proyecto_mio robguis$ python manage.py makemigrations
Migrations for 'appuno':
  0001_initial.py:
    - Create model Contacto
(envDjango) RobMac:proyecto_mio robguis$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, contenttypes, appuno, auth, sessions
Running migrations:
  Rendering model states... DONE
  Applying appuno.0001_initial... OK
(envDjango) RobMac:proyecto_mio robguis$
```

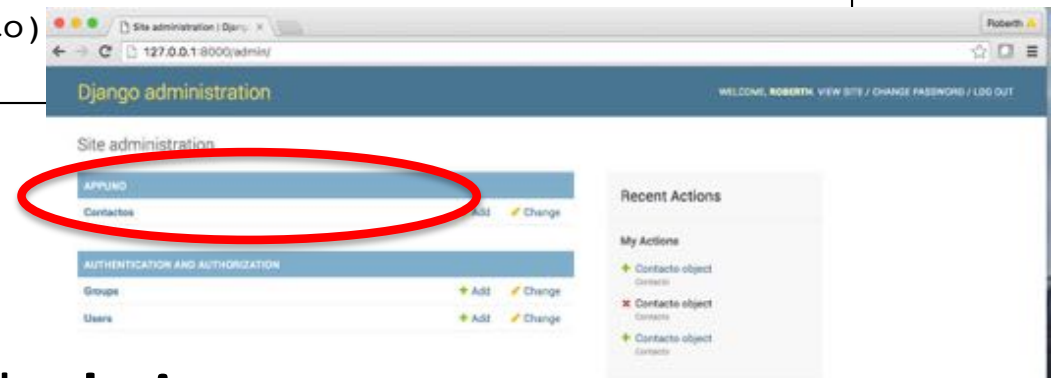

Anclar el modelo al admin.py

- Realizar cambios al admin.py

```
from django.contrib import admin
from .models import Contacto

class AdminContacto(admin.ModelAdmin):
    list_display = ["__str__", "nombres", "apellidos", "email"]
    class Meta(object):
        model = Contacto

admin.site.register(Contacto, AdminContacto)
```



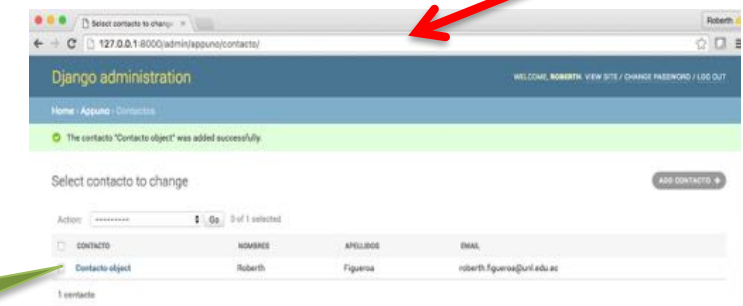
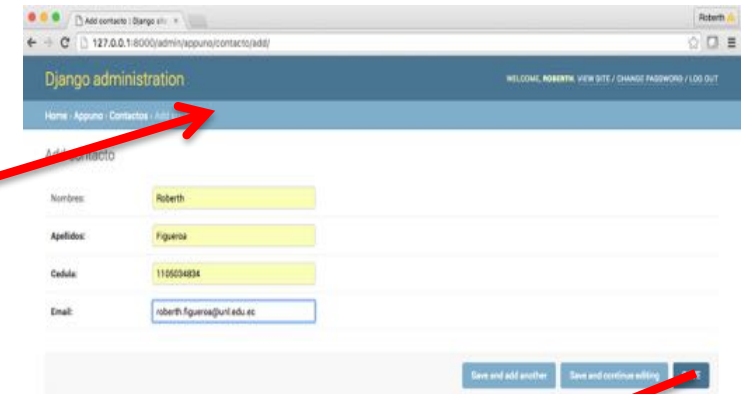
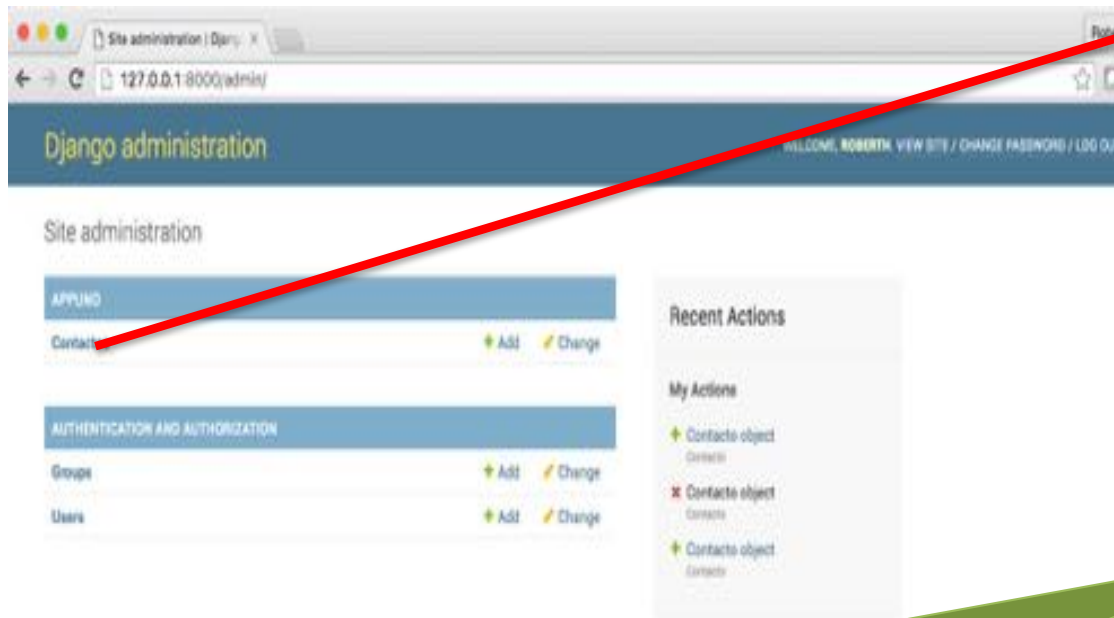
- Aplicar el anclaje al admin.py

usar → **python manage.py migrate**

Levantar el servidor de aplicaciones → **python manage.py runserver**

Resultado luego anclaje

- Pantalla modelo creado en código python.



el `__str__` hará de índice en la tabla y el objeto

Uso de Views

Modificar View

- Realizar cambios al **view.py**

```
#from django.http import HttpResponse
from django.shortcuts import render
```

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

```
def home_view(request, *args, **kwargs):
    print (args, kwargs)
    print (request)
    print (request.user)
    return render(request, "home.html", {})
    #return HttpResponse("<h1>Hola Bienvenido al Sistema appUno</h1>")
```

```
def about_view(request):
    return render(request, "about.html", {})
```

```
def contact_view(request,):
    return render(request, "contact.html", {})
```

Habilitar para enviar HTML directamente en el return

```
1 #from django.http import HttpResponse
2 from django.shortcuts import render
3
4 # Create your views here.
5
6 def home_view(request, *args, **kwargs):
7     print (args, kwargs)
8     print (request)
9     print (request.user)
10    return render(request, "home.html", {})
11    #return HttpResponse("<h1>Hola Bienvenido al Sistema appUno</h1>")
12
13 def about_view(request):
14     return render(request, "about.html", {})
15
16 def contact_view(request,):
17     return render(request, "contact.html", {})
18
19 def demo_view(request,):
20     return render(request, "demo.html", {})
```

Agregar un elemento nuevo a la lista urlpatterns en **urls.py**

- Agregar la url deseada archivo **urls.py**
`path('about/', about_view),`
- **about_view** : es la vista donde se ubica la función en **views.py**

```
from django.contrib import admin
from django.urls import path
from appUno.views import home_view, contact_view, about_view

urlpatterns = [
    path('', home_view, name='home'),
    path('about/', about_view),
    path('contact/', contact_view),
    path('admin/', admin.site.urls),
]
```

Acceder a las nuevas url's



Uso de Templates

Configurar los **templates** en **settings.py**

- Directorio de templates:

modificar a la ubicación → 'DIRS': [],

- Quedando así: **'DIRS'**

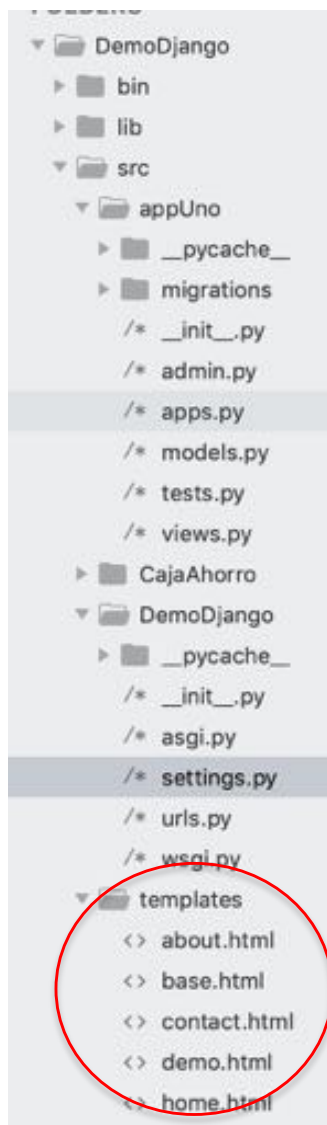
```
56
57 TEMPLATES = [
58     {
59         'BACKEND': 'django.template.backends.django.DjangoTemplates',
60         'DIRS': [BASE_DIR, "templates"],
61         'APP_DIRS': True,
62         'OPTIONS': {
63             'context_processors': [
64                 'django.template.context_processors.debug',
65                 'django.template.context_processors.request',
66                 'django.contrib.auth.context_processors.auth',
67                 'django.contrib.messages.context_processors.messages',
68             ],
69         },
70     ],
71 ]
72
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR, "templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

Modifique las **views** para cada **templates**

```
1 from django.http import HttpResponse
2 from django.shortcuts import render
3
4
5 def home_view(request):
6
7     return HttpResponse("<h1>Hola Bienvenido Señor@s Estudiantes 5A</h1>")
8
9 def about_view(request):
10     return render(request, "about.html", {})
11
12 def contact_view(request):
13     return render(request, "contact.html", {})
14
15 def demo_view(request):
16     return render(request, "demo.html", {})
17
18
19
```

Jerarquía de carpetas para templates



```
12 |
13 | from pathlib import Path
14 |
15 | # Build paths inside the project like this: BASE_DIR / 'subdir'.
16 | BASE_DIR = Path(__file__).resolve().parent.parent
17 | #/Users/roberth/Developer/DemoDjango/src
18 | print(BASE_DIR)
19 |
20 | # Quick-start development settings - unsuitable for production
21 | # See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/
22 |
23 | # SECURITY WARNING: keep the secret key used in production secret!
24 | SECRET_KEY = 'django-insecure-!w*p%4!*3q6zefyue0k@50&z3583ncweem(v7)3v1&^rgat*'
25 |
26 | # SECURITY WARNING: don't run with debug turned on in production!
27 | DEBUG = True
28 |
29 | ALLOWED_HOSTS = []
30 |
```

Hay que recordar que la ruta de su proyecto de Django la puede ver mediante la variable **BASE_DIR**, que se ubica en el archivo **settings.py**

Creamos el archivo html

contact.html

Se guarda el html en el directorio de **templates**:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Conctacto pagina</title>
6  </head>
7  <body>
8  <h1>Pagina de contacto con plantilla DBP</h1>
9  </body>
10 </html>
```

Se crean más archivos html en el directorio de **templates**, según la necesidad

Agregar un elemento nuevo a la lista urlpatterns en url.py de ser necesario

```
"""DemoDjango URL Configuration
```

```
The `urlpatterns` list routes URLs to views. For more information please see:
```

```
https://docs.djangoproject.com/en/3.2/topics/http/urls/
```

```
Examples:
```

```
Function views
```

1. Add an import: `from my_app import views`
2. Add a URL to urlpatterns: `path('', views.home, name='home')`

```
Class-based views
```

1. Add an import: `from other_app.views import Home`
2. Add a URL to urlpatterns: `path('', Home.as_view(), name='home')`

```
Including another URLconf
```

1. Import the `include()` function: `from django.urls import include, path`
2. Add a URL to urlpatterns: `path('blog/', include('blog.urls'))`

```
"""
```

```
from django.contrib import admin
```

```
from django.urls import path
```

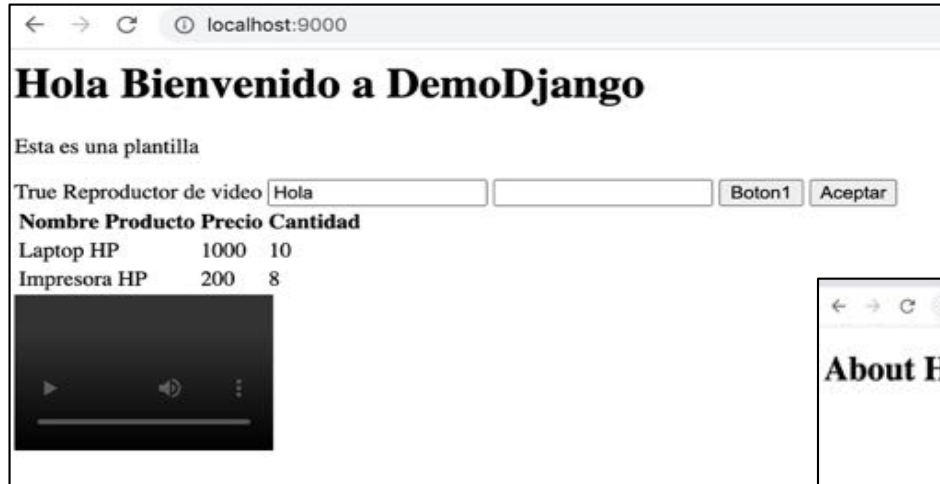
```
from appUno.views import home_view, contact_view, about_view, demo_view
```

```
urlpatterns = [  
    path('', home_view, name='home'),  
    path('about/', about_view),  
    path('demo/', demo_view),  
    path('contact/', contact_view),  
    path('admin/', admin.site.urls),  
]
```

importamos las vistas

```
1 """DemoDjango URL Configuration
2
3 The `urlpatterns` list routes URLs to views. For more information please see:
4   https://docs.djangoproject.com/en/3.2/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15
16 from django.contrib import admin
17 from django.urls import path
18 from appUno.views import home_view, contact_view, about_view, demo_view
19
20 urlpatterns = [
21     path('', home_view, name='home'),
22     path('about/', about_view),
23     path('demo/', demo_view),
24     path('contact/', contact_view),
25     path('admin/', admin.site.urls),
26 ]
27
```


Ver el resultado de uso de templates



Agregando código

Django versión 3.2.3

Agregar código python en templates

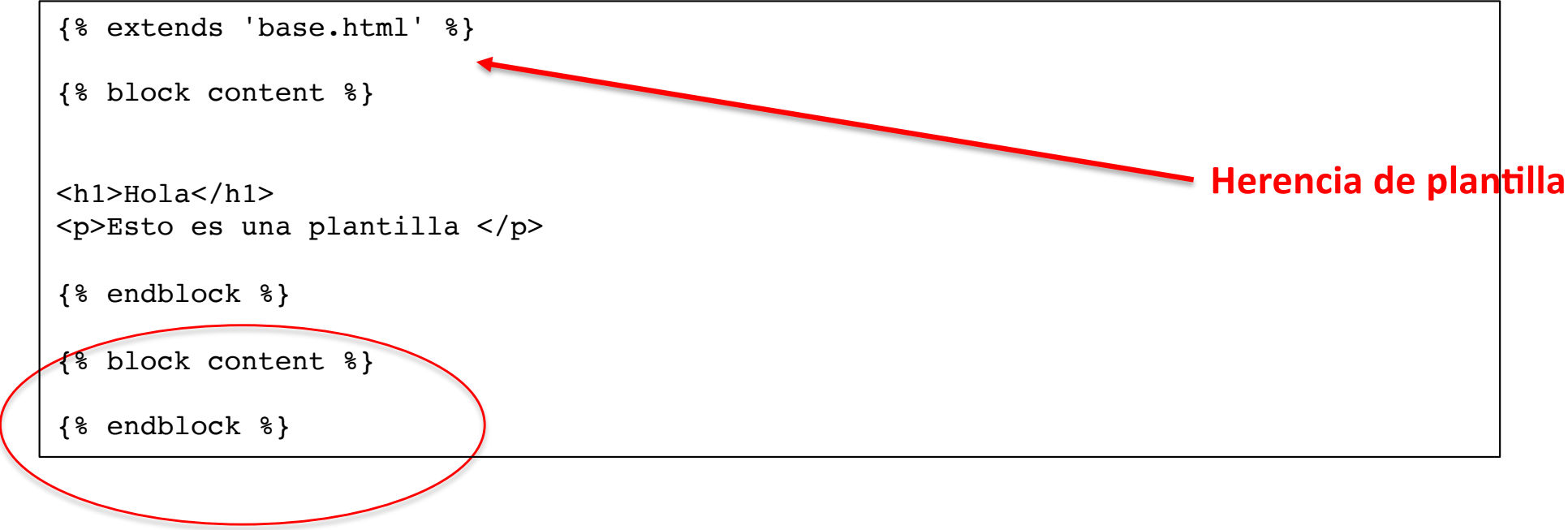
```
{% extends 'base.html' %}

{% block content %}

<h1>Hola</h1>
<p>Esto es una plantilla </p>

{% endblock %}

{% block content %}
{% endblock %}
```



Herencia de plantilla

Código para Verificar el usuario activo:

```
{{request.user.is_authenticated}}
```

Modificando contact.html

```
{% include "base.html" %}
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Conctacto pagina</title>
</head>
<body>
<h1>Pagina de contacto con plantilla DBP</h1>

  {{request.user.is_authenticated}}

</body>
</html>
```

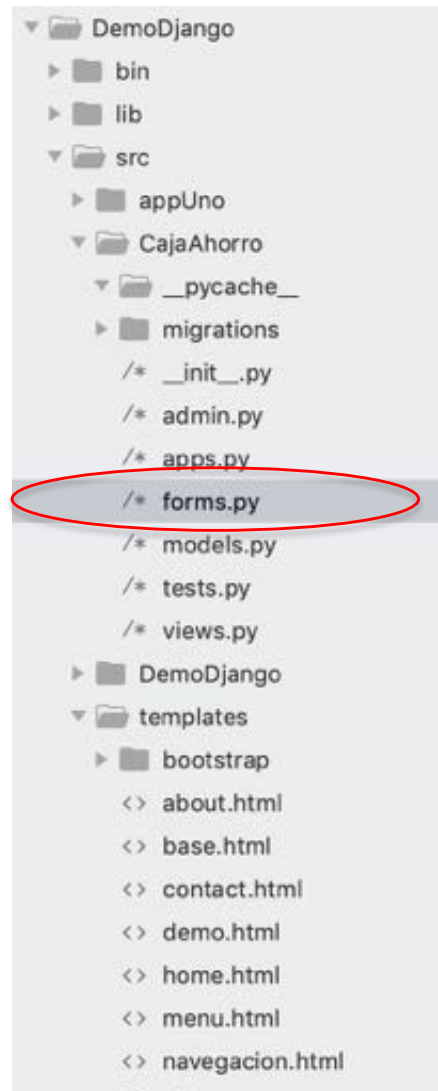
Se importa base.html



Agregando formularios

Django versión 3.2.3

Creamos el archivo **forms.py** dentro de la app



Definir el formulario forms.py

```
from django import forms
from .models import Contacto

class Formulario(forms.ModelForm):
    class Meta:
        model = Contacto
        fields=["nombres", "apellidos","cedula","email"]
```

```
1  from django import forms
2  from .models import Contacto
3
4  class Formulario(forms.ModelForm):
5      class Meta:
6          model = Contacto
7          fields=["nombres", "apellidos","cedula","email"]
```

Modificamos el archivo `view.py`

```
from django.shortcuts import render
from django.shortcuts import redirect
from .forms import Formulario
from .models import Contacto
```

```
def home_view(request):
    f = Formulario(request.POST or None)
    if request.method == 'POST':
        if f.is_valid():
            datos = f.cleaned_data
            c = Contacto()
            c.nombres = datos.get("nombres")
            c.apellidos = datos.get("apellidos")
            c.cedula = datos.get("cedula")
            c.email = datos.get("email")
            if c.save() != True:
                print('Imprimo en pantalla y guardo data en BD')
                print(f.cleaned_data)
            return redirect(home_view)

    context = {
        "form": f,
    }
    return render(request, "home.html", context)

def contact_view(request):
    return render(request, "contact.html", {})

def about_view(request):
    return render(request, "about.html", {})
```

f es un objeto de la Petición POST

Validar los campos de entradas:

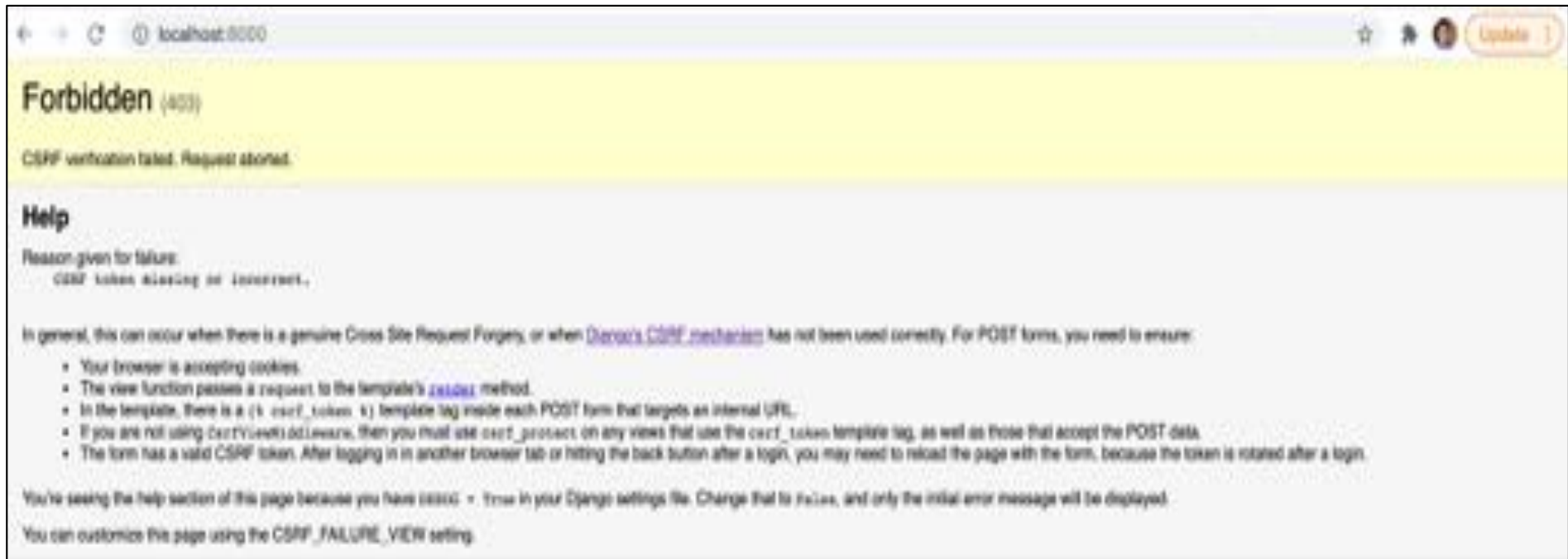
- Agregamos los requisitos de validación de POST
 - `request.POST`
- Agregamos None para que no se visualice los mensajes de
 - `request.POST or None`

El view.py modificado

```
1  from django.shortcuts import render
2  from django.shortcuts import redirect
3  from .forms import Formulario
4  from .models import Contacto
5
6  def home_view(request):
7      f = Formulario(request.POST or None)
8      if request.method == 'POST':
9          if f.is_valid():
10             datos = f.cleaned_data
11             c = Contacto()
12             c.nombres = datos.get("nombres")
13             c.apellidos = datos.get("apellidos")
14             c.cedula = datos.get("cedula")
15             c.email = datos.get("email")
16             if c.save() != True:
17                 print('Imprimo en pantalla y guardo data en BD')
18                 print(f.cleaned_data)
19                 return redirect(home_view)
20             context = {
21                 "form":f,
22             }
23             return render(request,"home.html",context)
24
25  def contact_view(request):
26      return render(request,"contact.html",{})
27
28  def about_view(request):
29      return render(request,"about.html",{})
30
31  def demo_view(request):
32      return render(request,"demo.html",{})
```

Vulnerabilidad Cross Site Request Forgery

CSRF en peticiones **POST**



Modificamos la plantilla `home.html`

- `{{form}}` es la variable context enviada por view a la plantilla.
- `form.as_p` → muestra el formulario en párrafo en html usando la etiqueta `<p>` por defecto.

```
<h1>Pagina de Inicio</h1>
<form method="post" action=""> {% csrf_token %}
    {{form.as_p}}
    <input type="submit" value="Aceptar">
</form>
<h1>Despues del formulario</h1>
```

- Agregamos una seguridad web en nuestro formulario `{% csrf_token %}`

Nota: Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legítimas provenientes de la víctima.

El archivo home.html

```
1 <h1>Pagina de Inicio</h1>
2 <form method="post" action=""> {% csrf_token %}
3     {{form.as_p}}
4     <input type="submit" value="Aceptar">
5 </form>
6 <h1>Despues del formulario</h1>
```

Reiniciar el Servidor Web wsgi

python manage.py runserver



← → ↻ ⓘ localhost:8000

Pagina de Inicio

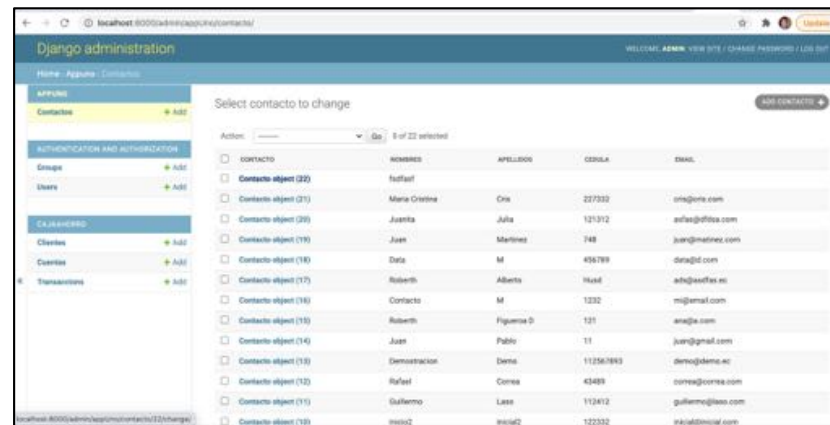
Nombres:

Apellidos:

Cedula:

Email:

Despues del formulario



Django administration

Home | Apparea | Contactos

CONTACTO

SELECT CONTACTO TO CHANGE

Actions: 8 of 22 selected

<input type="checkbox"/>	CONTACTO	NOMBRES	APELLIDOS	CEDULA	EMAIL
<input type="checkbox"/>	Contacto object (22)	Roberth	Figueroa		
<input type="checkbox"/>	Contacto object (21)	Maria Cristina	Cris	227332	crista@unl.com
<input type="checkbox"/>	Contacto object (20)	Julia	Julia	121312	esra@ofisa.com
<input type="checkbox"/>	Contacto object (19)	Juan	Marines	748	juan@marines.com
<input type="checkbox"/>	Contacto object (18)	Dana	M	456789	dana@id.com
<input type="checkbox"/>	Contacto object (17)	Roberth	Alberto	1648	rob@axofas.ec
<input type="checkbox"/>	Contacto object (16)	Contacto	M	1232	m@jemail.com
<input type="checkbox"/>	Contacto object (15)	Roberth	Figueroa D	121	esra@e.com
<input type="checkbox"/>	Contacto object (14)	Juan	Pablo	11	juan@gmail.com
<input type="checkbox"/>	Contacto object (13)	Demonstracion	Demo	112367893	demo@demo.ec
<input type="checkbox"/>	Contacto object (12)	Rafael	Correa	43489	correa@correa.com
<input type="checkbox"/>	Contacto object (11)	Guillermo	Lara	112412	guillermo@lao.com
<input type="checkbox"/>	Contacto object (10)	Woolf	Woolf	125332	wol@wolmail.com

Verificar que los datos se guarden en la base de datos de la aplicación

Redireccionar una App específica modificando las urls

Django versión 3.2.3

Accediendo desde otro Host externo o computador a mi aplicación en Django

Django versión 3.2.3

Modificar el settings.py

```
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.2/ref/settings/
11 """
12
13 from pathlib import Path
14
15 # Build paths inside the project like this: BASE_DIR / 'subdir'.
16 BASE_DIR = Path(__file__).resolve().parent.parent
17 #/Users/roberth/Developer/DemoDjango/src
18 print(BASE_DIR)
19
20 # Quick-start development settings - unsuitable for production
21 # See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/
22
23 # SECURITY WARNING: keep the secret key used in production secret!
24 SECRET_KEY = 'django-insecure-!w*p%4!*3q6zefyue0k@50&z3583ncweem(v7)3v1&^rgat*'
25
26 # SECURITY WARNING: don't run with debug turned on in production!
27 DEBUG = True
28
29 #Para permitir acceder desde host externos '*'
30 ALLOWED_HOSTS = ['*']
31
32
33 # Application definition
34
35 INSTALLED_APPS = [
36     'django.contrib.admin',
37     'django.contrib.auth',
38     'django.contrib.contenttypes',
39     'django.contrib.sessions',
40     'django.contrib.messages',
41     'django.contrib.staticfiles',
42     'appUno',
43     'CajaAhorro',
44 ]
45
```

ALLOWED_HOST = ['*'] → Permite acceso a otros equipos

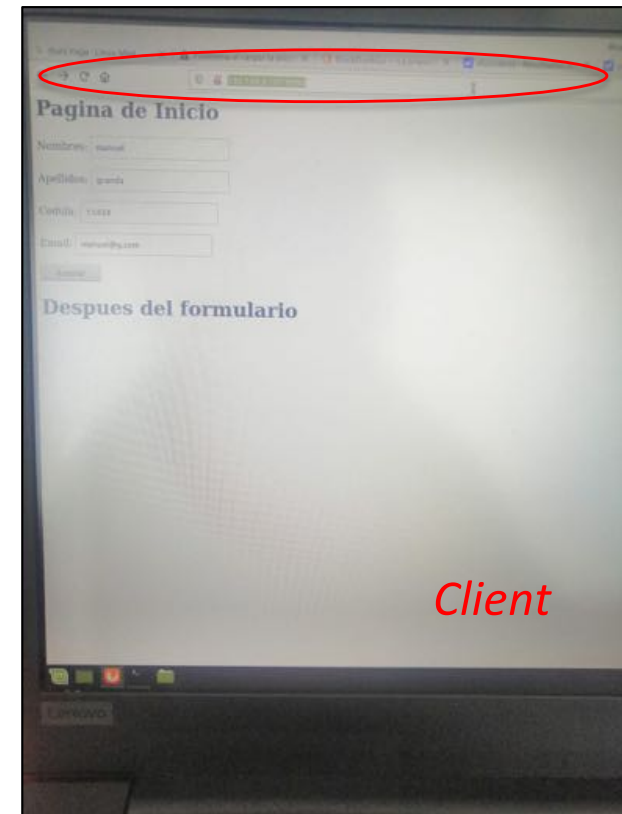
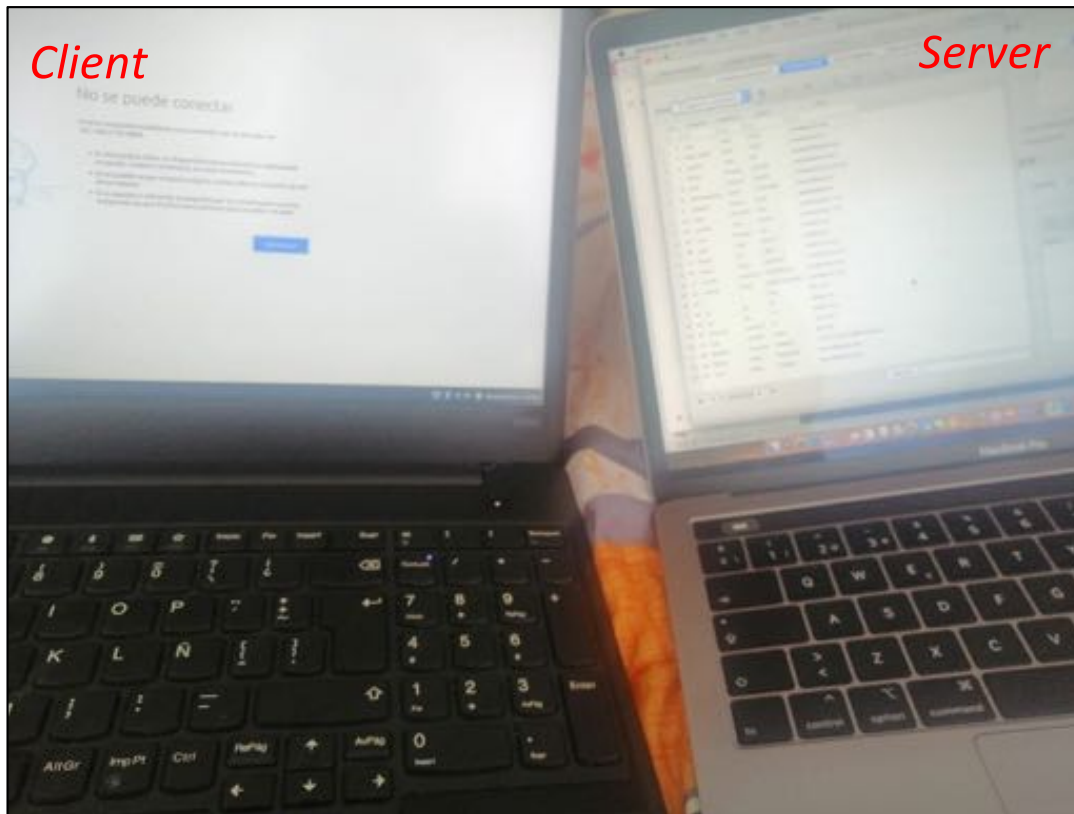
Reiniciar el Servidor Web WSGI 0:[puerto]

```
python manage.py runserver 0:8088
```

```
src — Python • Python manage.py runserver 0:8088 — 103x30
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
July 04, 2021 - 14:03:14
Django version 3.2.3, using settings 'DemoDjango.settings'
Starting development server at http://0:8000/
Quit the server with CONTROL-C.
^[[A[04/Jul/2021 14:04:50] "GET / HTTP/1.1" 200 746
[04/Jul/2021 14:05:03] "GET / HTTP/1.1" 200 746
^[[A[04/Jul/2021 14:05:03] "GET / HTTP/1.1" 200 746
^[[A[04/Jul/2021 14:05:03] "GET / HTTP/1.1" 200 746
(DemoDjango) macrob:src roberth$ python manage.py runserver 0:8088
/Users/roberth/Developer/DemoDjango/src
/Users/roberth/Developer/DemoDjango/src
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
July 04, 2021 - 14:07:56
Django version 3.2.3, using settings 'DemoDjango.settings'
Starting development server at http://0:8088/
Quit the server with CONTROL-C.
[04/Jul/2021 14:08:09] "GET / HTTP/1.1" 200 746
Not Found: /favicon.ico
[04/Jul/2021 14:08:09] "GET /favicon.ico HTTP/1.1" 404 2550
Imprimo en pantalla y guardo data en BD
{'nombres': 'manuel', 'apellidos': 'granda', 'cedula': '11434', 'email': 'manuel@g.com'}
[04/Jul/2021 14:08:57] "POST / HTTP/1.1" 302 0
[04/Jul/2021 14:08:57] "GET /about/ HTTP/1.1" 200 313
```

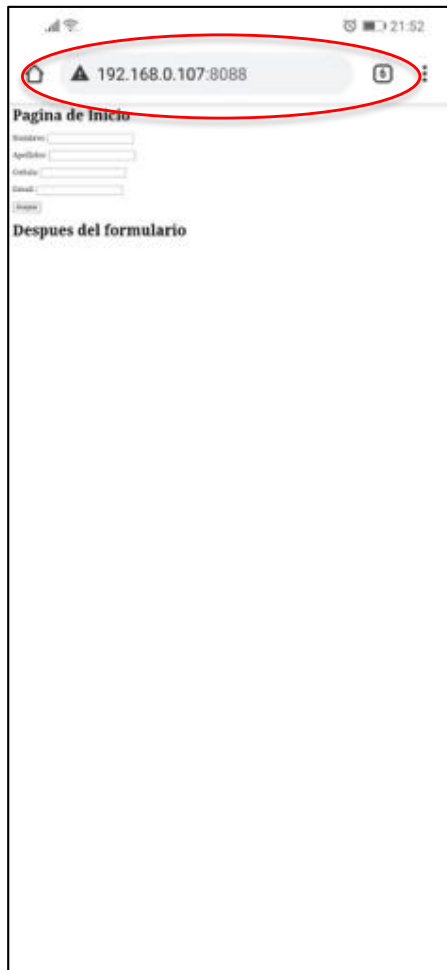
Invocar en el **host remoto** o equipo cliente

http://192.168.0.111:8087/



Ejecutar en el **host remoto** o celular

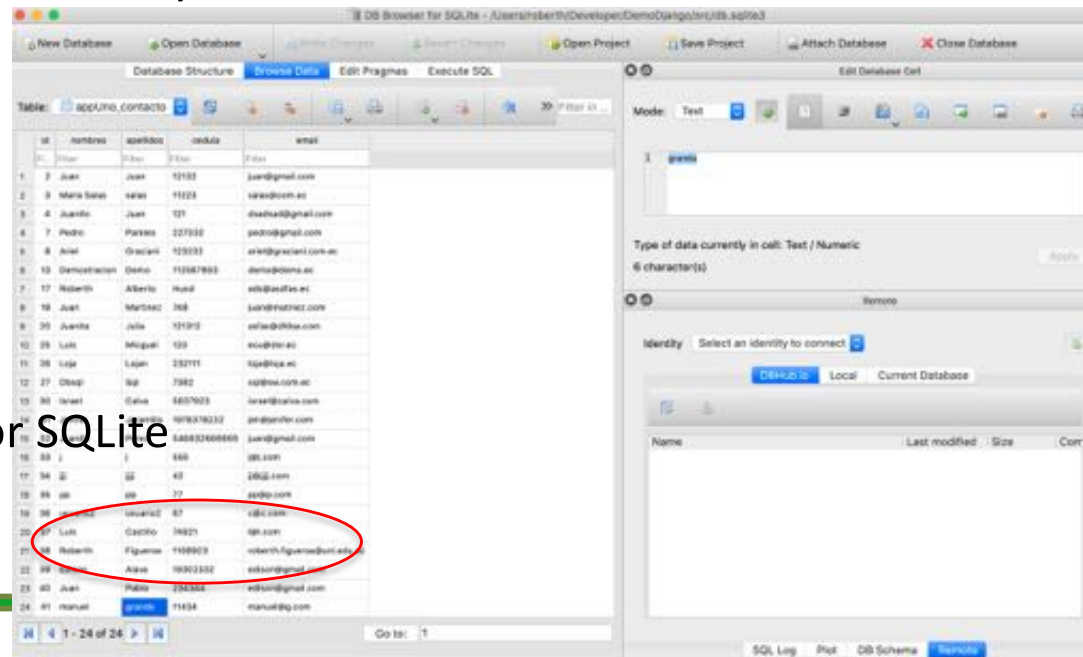
Colocar en el móvil → <http://192.168.0.107:8088/>



Ver en equipo servidor

```
Performing system checks... 207 ms 4,592 ms
System check identified no issues (0 silenced).
July 04, 2021 - 14:07:56
Django version 3.2.3, using settings 'DemoDjango.settings'
Starting development server at http://0:8088/
Quit the server with CONTROL-C.
[04/Jul/2021 14:08:09] "GET / HTTP/1.1" 200 746
Not Found: /favicon.ico
[04/Jul/2021 14:08:09] "GET /favicon.ico HTTP/1.1" 404 2550
Imprimo en pantalla y guardo data en BD
{'nombres': 'manuel', 'apellidos': 'granda', 'cedula': '11434', 'email': 'manuel@g.com'}
[04/Jul/2021 14:08:57] "POST / HTTP/1.1" 302 0
[04/Jul/2021 14:08:57] "GET /about/ HTTP/1.1" 200 313
```

Ejecutar desde el otro Host y verificar en el servidor los nuevos datos almacenados



id	nombres	apellidos	cedula	email
1	Juan	Juan	11101	juan@gmail.com
2	Maria	Sanz	11102	maria@com.ec
3	Juanito	Juan	121	juanito@gmail.com
4	Pedro	Perez	127330	pedro@gmail.com
5	Artal	Graciani	129331	artal@graciani.com.ec
6	Demonstracion	Demio	115947890	demo@demo.ec
7	Roberto	Alberto	muad	roberto@ec.ec
8	Juan	Martinez	368	juanmartinez.com
9	Juanita	Julia	121913	juanita@chka.com
10	Luis	Miguel	199	luis@ec.ec
11	Luis	Luis	23711	luis@ec.ec
12	Olga	Sup	7382	olga@com.ec
13	Israel	Olivia	6057601	israel@olivia.com
14	Roberto	Roberto	109318232	roberto@com
15	Juan	Juan	10032600000	juan@gmail.com
16	Juan	Juan	1	juan.com
17	Juan	Juan	43	juan.com
18	Juan	Juan	27	juan.com
19	Juan	Juan	67	juan.com
20	Luis	Capello	34821	luis.com
21	Roberto	Figueras	1108903	roberto@figueras.com.ec
22	Artal	Artal	109318232	artal@gmail.com
23	Juan	Pablo	236364	juan@gmail.com
24	manuel	granda	11434	manuel@g.com

Ver la DB en Browser for SQLite

Contacto



Universidad
Nacional
de Loja



Ing. Roberth Figueroa Díaz M.Sc.



Carrera Computación



roberth.figueroa@unl.edu.ec



www.unl.edu.ec



Universidad
Nacional
de Loja

Gracias por su atención

REDES  OFICIALES



Universidad Nacional De Loja - UNL



@UnOficial



UnOficial



UnOficial