

Tipo : Guía de Laboratorio
Capítulo : Django Web Framework
Duración : 180 minutos

I. OBJETIVO

Desarrollar un web app sencillo usando Django.

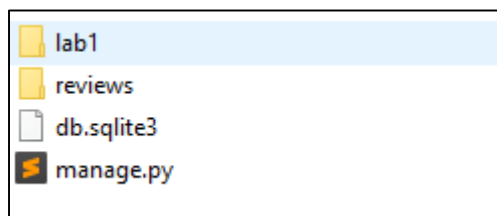
II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Instalar Anaconda en Windows
- Navegador web

III. EJECUCIÓN DEL LABORATORIO

- Ejercicio: Desarrollar un web app sencillo usando Django.
 - Abrir el command prompt - cmd.exe
 - Verificar instalación de Anaconda
 - a. `conda --version`
 - Crear entorno virtual
 - a. `conda create --name lab1 python=3.5`
 - Activar entorno virtual
 - a. `activate lab1`
 - Instalar django 1.11
 - a. `pip install django==1.11`
 - Ejecutar django
 - a. Crear la carpeta de trabajo (ej. LABS)
 - b. `django-admin startproject lab1`
 - c. `cd lab1`
 - d. `python manage.py runserver.`



1. Configuración inicial

- Instalar editor atom <https://atom.io/>
- Django-admin startapp reviews
- Editar lab1/settings.py

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

2. Modelo de datos

- Editar models.py

```
from django.db import models  
  
class Review(models.Model):  
    review = models.TextField(max_length=4000)
```

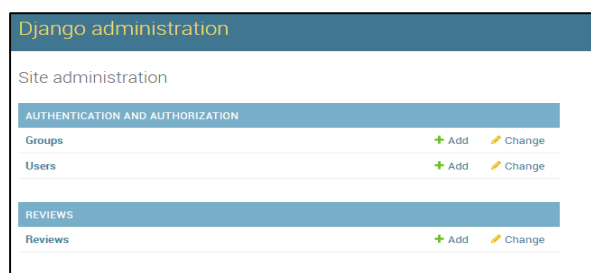
- python manage.py makemigrations
- python manage.py migrate

3. Django Admin

- python manage.py createsuperuser
- Editar admin.py

```
from django.contrib import admin  
from .models import Review  
  
admin.site.register(Review)
```

- 127.0.0.1/admin
- Login



4. Home Page

- a. Editar lab1/urls.py

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

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

- b. Crear archivo reviews/urls.py

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

from . views import index

urlpatterns = [
    url(r'', index, name="index"),
]
```

- c. Editar views.py

```
from django.shortcuts import render

from . models import Review

def index(request):
    params = {}
    return render(request, 'index.html', params)
```

- d. Crear archivo reviews/templates/index.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Bienvenido a REVIEW</title>
  </head>
  <body>
    Bienvenido a REVIEW
  </body>
</html>
```

5. Listar Reseñas

- a. Editar views.py

```
def list_review(request):
    reviews = Review.objects.all()
    params = {'reviews': reviews}
    return render(request, 'list_review.html', params)
```

- b. Editar reviews/urls.py

```
urlpatterns = [
    url(r'^$', index, name="index"),
    url(r'^list/$', list_review, name="list_review"),
]
```

- c. Editar index.html

```
<body>
    Bienvenido a REVIEW
    <p><a href="{% url 'list_review' %}">Listar reseñas</a></p>

</body>
```

- d. Crear reviews/templates/list_review.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>List Reviews</title>
  </head>
  <body>
    <table border=1 width=30%>
      <th>ID</th>
      <th>REVIEW</th>
      {% for r in reviews %}
      <tr>
        <td>{{r.id}}</td>
        <td>{{r.review}}</td>
      </tr>
      {% endfor %}
    </table>
    <p><a href="{% url 'index' %}">HOME</a></p>
  </body>
</html>
```

6. Crear Reseñas

a. Crear reviews/forms.py

```
from django.forms import ModelForm
from .models import Review

class ReviewForm(ModelForm):
    class Meta:
        model = Review
        fields = ['review']
```

b. Editar views.py

```
def add_review(request):
    if request.method == 'POST':
        form = ReviewForm(request.POST)
        if form.is_valid():
            form.save()
    else:
        form = ReviewForm()
    return render(request, 'add_review.html', {'form': form})
```

c. Crear reviews/templates/add_review.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>ADD REVIEW</title>
  </head>
  <body>
    Añadir una reseña
    <form method="post" novalidate>
      {% csrf_token %}
      {{ form }}
      <button type="submit">Submit</button>
    </form>

  </body>
</html>
```

- d. Editar index.html

```
<body>
  Bienvenido a REVIEW
  <p><a href="{% url 'list_review' %}">Listar reseñas</a></p>
  <p><a href="{% url 'add_review' %}">Agregar reseñas</a></p>

</body>
```

- e. Editar reviews/urls.py

```
urlpatterns = [
    url(r'^$', index, name="index"),
    url(r'^list/$', list_review, name="list_review"),
    url(r'^add/$', add_review, name="add_review"),
]
```

7. Pantallazos del app

Ejecutar en 127.0.0.1:8000

Bienvenido a REVIEW

[Listar reseñas](#)

[Agregar reseñas](#)

ID	REVIEW
1	Great stuff!!! Really great
2	Second review.
3	Impresionante!!! Bien hecho
4	NICE MOVIE. 5 stars

[HOME](#)

Añadir una reseña

Review:

[HOME](#)

IV. EVALUACIÓN

1. ¿Cómo se entiende el MVT en el contexto de esta aplicación?
 - a. **Respuesta:** El modelo está representado por el objeto review, que se puede visualizar por Django Admin y actualmente, almacenado en SQLite. Los views están representados por las rutas en urls.py, determinadas a través de patrones de texto, y la lógica establecida, mediante funciones en views.py. Finalmente, el template está representado por los archivos html que manejan la presentación en conjunto con Django Template Language.