

Ing. Computación

Instituto Tecnológico de Costa Rica



Introducción al desarrollo de aplicaciones web

Apunte Django

Profesora:

Ericka Solano Fernández

Realizado por:

Gabriel Quirós Obando

Experiencia con la instalación

La instalación del entorno es simple y rápida, en menos de 10 minutos ya se cuenta con el ambiente para hacer uso de Django. Sin embargo, antes de instalar Django se debe contar con Python.

Aspectos más relevantes

Fue creado por una empresa dedicada a un sitio de noticias, pero a petición de sus clientes, desarrollaron Django para satisfacer sus necesidades y posteriormente fue liberado al público.

Django es open source, permitiendo hacer conexiones a cualquier base de datos.

Share-nothing: entre sus capas no existe ninguna relación, asignando recursos a la capa que más lo necesite.

Tiene una característica llamada “Explícito es mejor que lo implícito”, donde el programador debe programar de forma que todo quede claro para los próximos programadores.

Arquitectura

ORM Object Relational Model, para convertir datos y almacenar los objetos en la base de datos.

MTV Model Template View, similar al patrón de diseño MVC.

Ventajas

Existe documentación extensa y completa

Al ser programado en Python, lo hace portable

Desventajas

Es más difícil que otros frameworks, y por su extensa documentación puede ser confuso.

Algunas aplicaciones que utilizan Django

Instagram, YouTube

Breve descripción del ejercicio a realizar en el taller

Aplicación de lista de quehaceres

Evidencias visuales

```
models.py × forms.py views.py urls.py
todo_list > models.py > ...
1  from django.db import models
2
3  class List(models.Model):
4      item = models.CharField(max_length=200)
5      completed = models.BooleanField(default=False)
6
7      def __str__(self):
8          return self.item + ' | ' + str(self.completed)
```

```
models.py forms.py × views.py urls.py
todo_list > forms.py > ListForm
1  from django import forms
2  from .models import List
3
4  class ListForm(forms.ModelForm):
5      class Meta:
6          model = List
7          fields = ["item", "completed"]
```

models.py forms.py views.py X urls.py

todo_list > views.py > home

```
1  from django.shortcuts import render, redirect
2  from .models import List
3  from .forms import ListForm
4  from django.contrib import messages
5
6  def home(request):
7      if request.method == 'POST':
8          form = ListForm(request.POST or None)
9
10         if form.is_valid():
11             form.save()
12             all_items = List.objects.all
13             messages.success(request, ('Tarea ha sido agregada a la lista!'))
14             return render(request, "home.html", {'all_items':all_items})
15         else:
16             all_items = List.objects.all
17             return render(request, 'home.html', {'all_items':all_items})
18
19  def delete(request, list_id):
20      item = List.objects.get(pk=list_id)
21      item.delete()
22      messages.success(request, ('Item has been deleted!'))
23      return redirect('home')
24
25  def cross_off(request, list_id):
26      item = List.objects.get(pk=list_id)
27      item.completed = True
28      item.save()
29      return redirect('home')
30
31  def uncross(request, list_id):
32      item = List.objects.get(pk=list_id)
33      item.completed = False
34      item.save()
35      return redirect('home')
36
37  def edit(request, list_id):
38      if request.method == 'POST':
39          item = List.objects.get(pk=list_id)
40          form = ListForm(request.POST or None, instance=item)
41          if form.is_valid():
42              form.save()
43              messages.success(request, ('Tarea ha sido editada!'))
44              return redirect('home')
45          else:
46              item = List.objects.get(pk=list_id)
47              return render(request, 'edit.html', {'item':item})
48
```

```
models.py  forms.py  views.py  urls.py  x
todo_list > urls.py > ...
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('', views.home, name="home"),
6      path('delete/<list_id>', views.delete, name="delete"),
7      path('cross_off/<list_id>', views.cross_off, name="cross_off"),
8      path('uncross/<list_id>', views.uncross, name="uncross"),
9      path('edit/<list_id>', views.edit, name="edit"),
10 ]
```

Hello, world! x +

localhost:8000

To-Do List

Tarea ha sido agregada a la lista! x

Enviar apunte Django	Cross Off	Delete
--------------------------------------	---------------------------	------------------------

localhost:8000

To-Do List

Tarea ha sido agregada a la lista! x

Enviar apunte Django	Uncross	Delete
Almorzar	Cross Off	Delete