Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación
Introducción al Desarrollo de Páginas Web
Taller Django

Profesora:
Ericka Solano Fernández

Estudiantes:
Arlem Gabriel Brenes Aguilar

Verano, 2019

Experiencia de Instalación

Para efectos de la instalación, el instructivo fue lo suficientemente claro para realizar cada uno de los pasos correspondientes, además de que la tecnología no ameritaba realizar exhaustivas configuraciones por tanto la experiencia fue excelente. Además, los compañeros aportaron el servidor ya configurado lo que permitió agilidad en el proceso del taller.

Aspectos Relevantes de la Tecnología

Creado por Adrian Holovaty y Simon Willison, creado en LJWORLD.COM, con el tiempo Djando pasó a ser de dominio público, su nombre se debe en honor a un guitarrista. Django es opensource, es compatible prácticamente con todo como bases de datos PostgresSQL, Maria DB, ORACLE,MySQL. Su comunidad es open source es bastante grande y activa, esta herramienta tiene share nothing que consiste en que se pueden asignar recursos a cada capa conforme se vayan usando lo que ayuda a la escalabilidad. Cuenta con un sistema de cache más eficiente, guarda las páginas en el caché.
Con respecto a su arquitectura es un ORM Object Relational Model, pues toma los datos de un lenguaje orientado a objetos y guarda esos datos para dar estructura a los datos en la base.

- Model: estructura de cómo se guardaran los dato.
- Forms: Creación de forms usa ModelForm, se pasa el modelo a esta clase para después guardarlo automáticamente.
- Template: es la capa de UI que perciben los usuarios
- View: es el controlador

Ahora bien, como todo este cuenta con sus respectivas ventajas y desventajas, de las cuales podemos destacar las siguientes:

Ventajas: Es muy completo porque tiene bastante documentación y completa, es portable porque está programado en Python lo que lo hace multiplataforma. Se caracteriza por el desarrollo ágil, es reutilizable y es open source.

Desventajas: Como existe tanta documentación podría llegar a ser confuso.
Aplicaciones que usan DJANGO: Instagram, Dropbox, Bitbucket, Youtube, Nasa, The New York Times

Descripción del Ejercicio

Para efectos del taller el ejercicio consistió en realizar una lista de tareas haciendo uso de Django. El ejercicio permite agregar, editar, tachar y eliminar tareas, para ello se debía agregar el código pertinente que permitiera la navegación desde el template hasta el model y registrar los datos.

Evidencias Visuales

Código:

forms.py

```
from django import forms
from .models import List

class ListForm ( forms . ModelForm ):
    class Meta :
        model = List
        fields = [ "item" , "completed" ]
```

models.py

```
from django.db import models

class List ( models . Model ):

    item = models.CharField( max_length = 200 )
    completed = models.BooleanField( default = False )

    def __str__ ( self ):
        return self .item + ' | ' + str ( self .completed)
```

urls.py

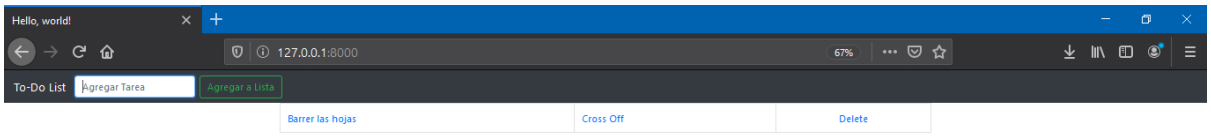
```
from django.urls import path
from . import views
urlpatterns = [
    path( '' , views.home, name = "home" ),
    path( 'delete/<list_id>' , views.delete, name = "delete" ),
    path( 'cross_off/<list_id>' , views.cross_off, name = "cross_off" ),
    path( 'uncross/<list_id>' , views.uncross, name = "uncross" ),
    path( 'edit/<list_id>' , views.edit, name = "edit" ),
]
```

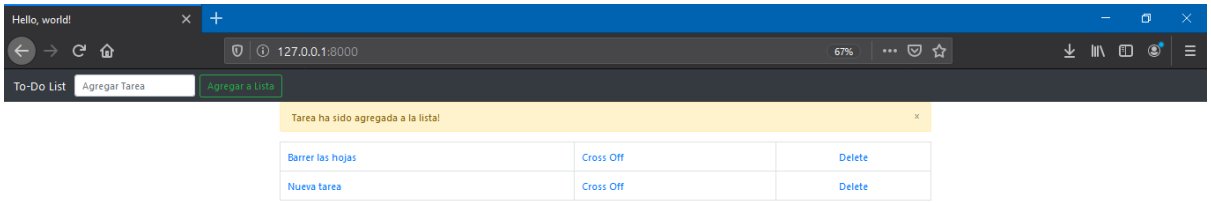
views.py

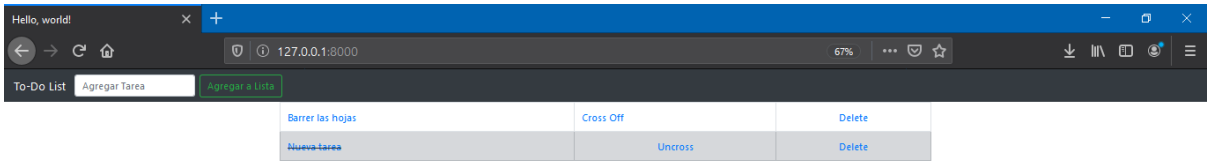
```
from django.shortcuts import render, redirect
from .models import List
from .forms import ListForm
from django.contrib import messages

def home ( request ):
    if request.method == 'POST':
        form = ListForm(request.POST or None )
        if form.is_valid():
            form.save()
            all_items = List.objects.all
```

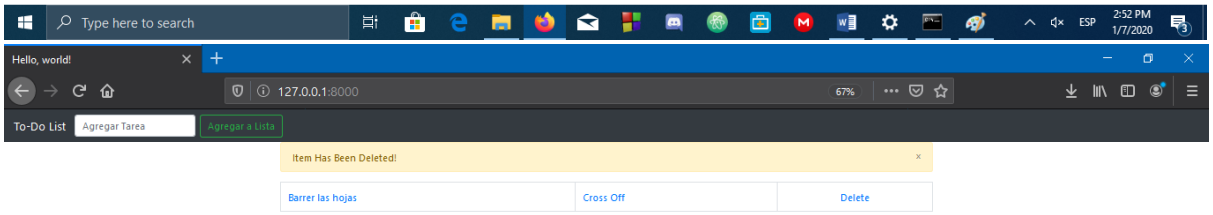
```
            messages.success(request, ( 'Tarea ha sido agregada a la lista!' )
)
            return render(request, "home.html" ,{ 'all_items' :all_items})
    else:
        all_items = List.objects.all
        return render(request, 'home.html' , { 'all_items' :all_items})

def delete ( request , list_id ):
    item = List.objects.get( pk =list_id)
    item.delete()
    messages.success(request, ( 'Item Has Been Deleted!' ))
    return redirect( 'home' )

def cross_off ( request , list_id ):
    item = List.objects.get( pk =list_id)
    item.completed = True
    item.save()
    return redirect( 'home' )

def uncross ( request , list_id ):
    item = List.objects.get( pk =list_id)
    item.completed = False
    item.save()
    return redirect( 'home' )

def edit ( request , list_id ):
    if request.method == 'POST' :
        item = List.objects.get( pk =list_id)
        form = ListForm(request.POST or None , instance =item)
        if form.is_valid():
            form.save()
            messages.success(request,( 'Tarea ha sido editada!' ))
            return redirect( 'home' )
    else:
        item = List.objects.get( pk =list_id)
        return render(request, 'edit.html' , { 'item' :item})
```

Resultado Obtenido:






Hello, world!
127.0.0.1:8000
67%
To-Do List
Agregar Tarea
Agregar a Lista
Barrer las hojas
Cross Off
Delete
Nueva tarea
Uncross
Delete


Type here to search
ESP
2:52 PM
1/7/2020
Hello, world!
127.0.0.1:8000
67%
To-Do List
Agregar Tarea
Agregar a Lista
Item Has Been Deleted!
Barrer las hojas
Cross Off
Delete


Type here to search
ESP
2:53 PM
1/7/2020