**Tarea AE4 grupal, módulo 6**

**Elabore una pequeña guía que sintetice los pasos necesarios para levantar un formulario desde Django con Forms y Models.**

**ENTORNO VIRTUAL**

python --version

virtualenv -p python311 "nombre del proyecto"

.\"nombre del proyecto"\Scripts\activate

**VAMOS CON DJANGO**

pip install django

python -m django --version

django-admin startproject "nombre del proyecto"

django-admin startproject "nombre del proyecto" . (para evitar un subproyecto con el mismo nombre)

cd "nombre del proyecto"

python manage.py migrate

python manage.py runserver

python manage.py startapp "nombre aplicación"

**EN APLICACION**

Crear carpeta "templates", donde irán los templates HTML

Crear carpeta "static", dónde iran subdirectorios de estilos CSS e imagenes

**EN ARCHIVO SETTINGS.PY DEL PROYECTO**

Incluir la o las aplicaciones de nuestro proyecto en:

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'aplicacion\_2'

]

**EN ARCHIVO MODELS.PY DE LA APLICACIÓN**

Crear clases bajo el esquema models.Model de Django, ejemplo:

from django.db import models

class registroUsuarios(models.Model):

    rut = models.CharField(max\_length=10, unique=True)

    nombre = models.CharField(max\_length=20)

    apellido = models.CharField(max\_length=20)

    def \_\_str\_\_(self):

        return "Usuario RUT %s y nombre %s %s" %(self.rut, self.nombre, self.apellido)

**EN ARCHIVO URLS.PY DEL PROYECTO**

Acá creamos las direcciones o enrutamientos de las vistas. Por ejemplo:

from django.contrib import admin

from django.urls import path

from aplicacion\_2.views import primeraFuncion

from aplicacion\_2.views import formularioContacto

from aplicacion\_2.views import informeUsuarios

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', primeraFuncion, name = 'index'),

    path('registro/', formularioContacto, name = 'registro'),

    path('salida/', informeUsuarios, name = 'salida'),

]

**EN ARCHIVO FORMS.PY DE LA APLICACIÓN**

Creamos los formularios de datos, provenientes de la base de datos django. Se enlazarán a los HTML de salida, mostrando información de las tablas de la base. Por ejemplo:

from django import forms

from .models import registroUsuarios, Proveedores

class registroForm(forms.ModelForm):

    class Meta:

        model = registroUsuarios

        #fields = ['rut', 'nombre', 'apellido'] /// esto si quiero algunos campos. Alternativa, "all"

        fields = '\_\_all\_\_'

**EN ARCHIVO VIEWS.PY DE LA APLICACIÓN**

Crear vistas a través de funciones python, las cuales renderizan los templates HTML, de acuerdo a cada requerimiento. También se renderizan los formularios creados en forms.py.

from django.shortcuts import render

from aplicacion\_2.models import registroUsuarios,

from .forms import registroForm

def primeraFuncion(request):

    return render(request, 'landing.html')

def formularioContacto(request):

    data = {

        'form': registroForm()

    }

    if request.method == 'POST':

        formulario = registroForm(data=request.POST)

        if formulario.is\_valid():

            formulario.save()

            data["mensaje"] = "USUARIO REGISTRADO !!!"

        else:

            data["form"] = formulario

    return render(request, 'registro.html', data)

def informeUsuarios(request):

     #'usuarios' es una instancia de la clase "registroUsuarios" definida en models.py

    usuarios = registroUsuarios.objects.all()

    return render(request, 'salida.html', {'usuarios': usuarios})

**EN ARCHIVO ADMIN.PY DE LA APLICACIÓN**

Le decimos al sistema administrador de base de datos de Django, las clase que queremos trabajar y que debemos mostrar de cada tabla.

from django.contrib import admin

from aplicacion\_2.models import registroUsuarios

class UsuariosAdmin(admin.ModelAdmin):

    list\_display = ('nombre', 'apellido', 'direccion', 'email', 'ciudad')

admin.site.register(registroUsuarios, UsuariosAdmin)

**PASOS FINALES**

Una vez hecho lo anterior, volvemos a migrar y usar la base de datos Django, con las siguientes instrucciones:

* python manage.py makemigrations
* python manage.py migrate
* python manage.py runserver. Es aquí nos aparecerá la dirección de nuestro sistema web.