

Informe de Laboratorio 6

Tema: DJANGO

Nota

Estudiante	Escuela	Asignatura
Sergio Hanco Mullisaca shanccom@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: II Código:

Laboratorio	Tema	Duración
6	DJANGO	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 26 Mayo 2024	Al 29 Mayo 2024

1. Tarea

- Informe de laboratorio
- Video en Flip
- Ejercicios Propuestos

2. Equipos, materiales y temas utilizados

- VS
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.

3. URL de Repositorio Github

- URL del video en yt.
- <https://youtu.be/1L9XbhlCb8c>
- URL del video en flip.
- <https://flip.com/s/e5hcCChND88N>
- URL del GITHUB.
- https://github.com/shanccom/Programacion_Web_2.git

4. Actividades

4.1. EJERCICIO PROPUESTO

CREACION ENTORNO VIRTUAL CON GITIGNORE

```
PS C:\Sergio\SEMESTRE 2024A\PMEB\Programacion_Web_2\LAB_06_DJANGO\ejercicio_propuesto> ls

Directorio: C:\Sergio\SEMESTRE 2024A\PMEB\Programacion_Web_2\LAB_06_DJANGO\ejercicio_propuesto

Mode                LastWriteTime         Length Name
----                -
-a-----          28/05/2024   22:40             0 .gitignore.txt

PS C:\Sergio\SEMESTRE 2024A\PMEB\Programacion_Web_2\LAB_06_DJANGO\ejercicio_propuesto> python -m venv env
PS C:\Sergio\SEMESTRE 2024A\PMEB\Programacion_Web_2\LAB_06_DJANGO\ejercicio_propuesto> .\env\Scripts\activate
(env) PS C:\Sergio\SEMESTRE 2024A\PMEB\Programacion_Web_2\LAB_06_DJANGO\ejercicio_propuesto> pip install django
Collecting django
  Using cached Django-5.0.6-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.7.0 (from django)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.5.0-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from django)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached Django-5.0.6-py3-none-any.whl (8.2 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.0-py3-none-any.whl (43 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.0.6 sqlparse-0.5.0 tzdata-2024.1
(env) PS C:\Sergio\SEMESTRE 2024A\PMEB\Programacion_Web_2\LAB_06_DJANGO\ejercicio_propuesto> |
```

ARCHIVOS MODIFICADOS (app)

Listing 1: models.py

```
from django.db import models

# Creacion de modelos propios

class Alumno(models.Model):
    nombre = models.CharField(max_length = 50)
    apellido = models.CharField(max_length = 50)
    cui = models.CharField(max_length = 8, unique = True)

    def __str__(self):
        return f"{self.nombre} {self.apellido}"

class Curso(models.Model):
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField(blank=True)

    def __str__(self):
```

```
        return self.nombre

class NotasAlumno(models.Model):
    alumno = models.ForeignKey(Alumno, on_delete = models.CASCADE)
    curso = models.ForeignKey(Curso, on_delete = models.CASCADE)
    nota = models.DecimalField(max_digits = 5, decimal_places = 2)

    def __str__(self):
        return f"{self.alumno} - {self.curso}: {self.nota}"
```

Listing 2: admin.py

```
from django.contrib import admin
from .models import Alumno, Curso, NotasAlumno

admin.site.register(Alumno)
admin.site.register(Curso)
admin.site.register(NotasAlumno)
```

Listing 3: apps.py

```
from django.apps import AppConfig

class AdministracionConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'administracion'
```

Listing 4: forms.py

```
from django import forms
from .models import Alumno, Curso, NotasAlumno

class AlumnoForm(forms.ModelForm):
    class Meta:
        model = Alumno
        fields = ['nombre', 'apellido', 'cui']

class CursoForm(forms.ModelForm):
    class Meta:
        model = Curso
        fields = ['nombre']

class NotasAlumnosPorCursoForm(forms.ModelForm):
    class Meta:
        model = NotasAlumno
        fields = ['alumno', 'curso', 'nota']
```

Listing 5: urls.py

```
from django.urls import path
from .views import nuevo_alumno, nuevo_curso, nueva_nota, lista_alumnos, lista_cursos,
    lista_notas, eliminar_alumno, eliminar_curso, eliminar_nota

urlpatterns = [
```

```
path('', nuevo_alumno, name='inicio'),
path('nuevo_alumno/', nuevo_alumno, name='nuevo_alumno'),
path('nuevo_curso/', nuevo_curso, name='nuevo_curso'),
path('nueva_nota/', nueva_nota, name='nueva_nota'),
path('lista_alumnos/', lista_alumnos, name='lista_alumnos'),
path('lista_cursos/', lista_cursos, name='lista_cursos'),
path('lista_notas/', lista_notas, name='lista_notas'),
path('eliminar_alumno/<int:alumno_id>', eliminar_alumno, name='eliminar_alumno'),
path('eliminar_curso/<int:curso_id>', eliminar_curso, name='eliminar_curso'),
path('eliminar_nota/<int:nota_id>', eliminar_nota, name='eliminar_nota'),
]
```

Listing 6: views.py

```
from django.shortcuts import render, redirect, get_object_or_404
from .forms import AlumnoForm, CursoForm, NotasAlumnosPorCursoForm
from .models import Alumno, Curso, NotasAlumno

def nuevo_alumno(request):
    if request.method == "POST":
        form = AlumnoForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('lista_alumnos')
    else:
        form = AlumnoForm()
    return render(request, 'administracion/nuevo_alumno.html', {'form': form})

def nuevo_curso(request):
    if request.method == "POST":
        form = CursoForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('lista_cursos')
    else:
        form = CursoForm()
    return render(request, 'administracion/nuevo_curso.html', {'form': form})

def nueva_nota(request):
    if request.method == "POST":
        form = NotasAlumnosPorCursoForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('lista_notas')
    else:
        form = NotasAlumnosPorCursoForm()
    return render(request, 'administracion/nueva_nota.html', {'form': form})

def lista_alumnos(request):
    alumnos = Alumno.objects.all()
    return render(request, 'administracion/lista_alumnos.html', {'alumnos': alumnos})

def lista_cursos(request):
    cursos = Curso.objects.all()
    return render(request, 'administracion/lista_cursos.html', {'cursos': cursos})
```

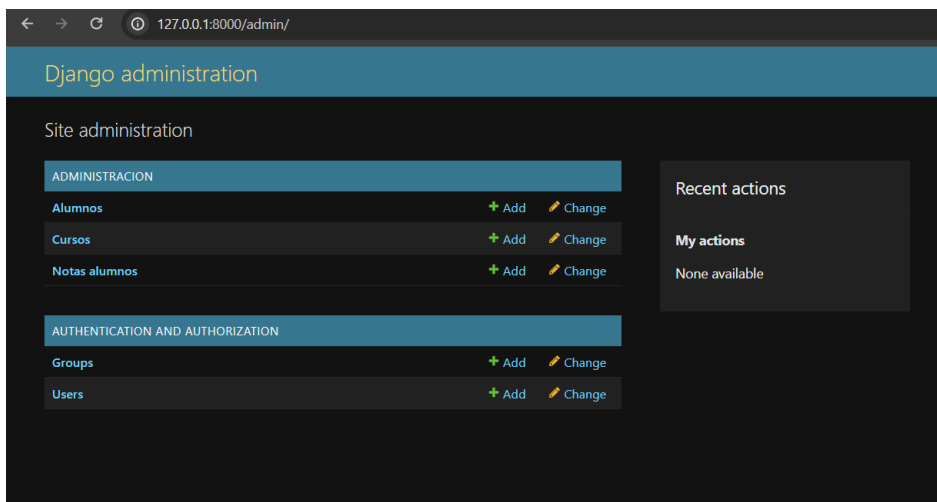
```
def lista_notas(request):
    notas = NotasAlumno.objects.all()
    return render(request, 'administracion/lista_notas.html', {'notas': notas})

def eliminar_alumno(request, alumno_id):
    alumno = Alumno.objects.get(pk=alumno_id)
    alumno.delete()
    return redirect('lista_alumnos')

def eliminar_curso(request, curso_id):
    curso = Curso.objects.get(pk=curso_id)
    curso.delete()
    return redirect('lista_cursos')

def eliminar_nota(request, nota_id):
    nota = NotasAlumno.objects.get(pk=nota_id)
    nota.delete()
    return redirect('lista_notas')
```

EJECUCION



ARCHIVOS MODIFICADOS (PROYECTO)

Listing 7: settings.py

```
"""
Django settings for ejercicioPropuesto project.

Generated by 'django-admin startproject' using Django 5.0.6.

For more information on this file, see
https://docs.djangoproject.com/en/5.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/5.0/ref/settings/
"""
```

```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-0d%raj871b-2=5vz#hep+1pfvd+e27)0!)wj!u6$!@ic10e5@u'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

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

    # APPS PROPIAS
    'administracion'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'ejercicioPropuesto.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(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',
            ]
        }
    }
]
```

```
        ],
    },
},
]

WSGI_APPLICATION = 'ejercicioPropuesto.wsgi.application'

# Database
# https://docs.djangoproject.com/en/5.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/5.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
```

```
# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field  
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Listing 8: urls.py

```
from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('administracion/', include('administracion.urls')),  
]
```

5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

Puntos	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
Total		20		17	