

## Informe de Laboratorio 07

**Tema: Django – Modelos, Vistas y Templates**

Nota

Integrantes	Escuela	Asignatura
Miguel Angel Alvarez Choque 20230477 Rodrigo Alexander Fernández Huarca 20230465 Eduardo Joel Cuno Salazar 20231497 Jose Maria Ticona Saure 20233482	Escuela Profesional de Ingeniería de Sistemas	Programación Web II Semestre: I

Laboratorio	Tema	Duración
07	Django – Modelos, Vistas y Templates	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 12 de junio 2024	Al 14 de junio 2024

## RESULTADOS Y PRUEBAS

### 1. EJERCICIOS RESUELTOS:

#### Ejercicio propuesto:

- *Implementa un Sistema en Django que maneje una tabla de Alumnos, una de Cursos y una de NotasAlumnosPorCurso y que permita ingresar a nuevos alumnos, nuevos cursos y finalmente permita ingresar las notas por curso.*

#### 1.1. Parte 1:

- En esta parte se realizó la creación del proyecto que se llamará **colegio**.
- Se creó también una aplicación llamada **gestion**
- Se añadió la aplicación **gestion** al proyecto.
- Se creó los modelos **Alumno**, **Curso** y **NotaAlumnosPorCurso**
- Se realizaron las migraciones para crear las tablas en la base de datos
- Finalmente se añadió los modelos al panel de administración

### 1.1.1. evidencias:

```
C:\Users\joset\OneDrive\Escritorio\Django\Pweb-6\Pweb-6\Laboratorio_7>django-admin startproject colegio
C:\Users\joset\OneDrive\Escritorio\Django\Pweb-6\Pweb-6\Laboratorio_7>cd colegio
C:\Users\joset\OneDrive\Escritorio\Django\Pweb-6\Pweb-6\Laboratorio_7\colegio>python manage.py startapp gestion
C:\Users\joset\OneDrive\Escritorio\Django\Pweb-6\Pweb-6\Laboratorio_7\colegio>python manage.py makemigrations
Migrations for 'gestion':
  gestion\migrations\0001_initial.py
    - Create model Alumno
    - Create model Curso
    - Create model NotaAlumnosPorCurso
C:\Users\joset\OneDrive\Escritorio\Django\Pweb-6\Pweb-6\Laboratorio_7\colegio>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, gestion, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying gestion.0001_initial... OK
  Applying sessions.0001_initial... OK
```

## 1.2. Parte 2:

- En esta segunda parte se realizó la creación de los formularios para nuestros modelos. Estos formularios se guardaron en *gestion/forms.py*
- Se añadieron vistas para poder manejar las operaciones CRUD. se añadieron en *gestion/views.py*.
- Luego se procedió a configurar las url para las vistas antes implementadas. Esto se configuró en *gestion/urls.py*

### 1.2.1. evidencias:

## 1.3. Parte 3:

- En esta parte se realizó todo lo relacionado con las plantillas para las vistas antes creadas. Para ello se creo la carpeta *gestion/templates/gestion*.
- Los htmls implementados son los siguientes:

### 1.3.1. Evidencias:

- agregar-alumno y lista-alumnos

← → ↻ 127.0.0.1:8000/gestion/agregar\_alumno/

#### Agregar Alumno

Nombre:

Apellido:

Email:

[Volver a la lista](#)

← → ↻ 127.0.0.1:8000/gestion/listar\_alumnos/

#### Lista de Alumnos

- eduardo cuno

[Agregar Alumno](#)

- agregar-curso y lista-cursos

← → ↻ 127.0.0.1:8000/gestion/agregar\_curso/

#### Agregar Curso

Nombre:

Se conoce como cálculo a todo proceso matemático que tiene como fin obtener el resultado de una expresión numérica. EL cálculo busca desarrollar las habilidades de razonamiento y solución de problemas a través de la interpretación lógico-matemática de expresiones simbólicas y numéricas.

Descripcion:

[Volver a la lista de cursos](#)

← → ↻ 127.0.0.1:8000/gestion/listar\_cursos/

#### Lista de Cursos

- Calculo

[Agregar Curso](#)

- agregar-nota y lista-notas

← → ↻ 127.0.0.1:8000/gestion/agregar\_nota/

#### Agregar Nota

Alumno:

Curso:

Nota:

[Volver a la lista de notas](#)

## CONCLUSIONES

- Aprender Django te equipa con las habilidades necesarias para desarrollar aplicaciones web modernas y eficientes. Su enfoque en la simplicidad, la rapidez de desarrollo y la seguridad lo convierte en una excelente elección tanto para principiantes como para desarrolladores experimentados. Además, la comunidad y la documentación de Django proporcionan un soporte invaluable durante el proceso de aprendizaje y desarrollo. Con Django, puedes construir aplicaciones web de manera rápida, segura y escalable, lo que te permite centrarte en la creación de valor a través de tu código
- Finalmente, la combinación de modelos, vistas y plantillas en Django permite construir aplicaciones robustas y escalables, enfocándose en la creación de valor y en la resolución de problemas complejos de manera efectiva.

## METODOLOGÍA DE TRABAJO

- Para el presente trabajo se repartió cada parte del trabajo a un integrante del equipo. Los roles principales fueron:
  - Desarrollo de las actividades y ejercicios del laboratorio:
    - José Maria - Parte 1
    - Rodrigo - parte 2
    - Eduardo - parte 3
  - Elaboración del informe: Miguel

## REFERENCIAS Y BIBLIOGRAFÍA

Para el trabajo se ocuparon fuentes básicas para entender el uso de Django. Se procede a compartir las fuentes bibliograficas:

- <https://docs.djangoproject.com/es/3.2/>
- <https://docs.djangoproject.com/es/3.2/ref/models/fields/#field-types>
- [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial\\_local\\_library\\_website](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website)

### 1.5. URL'S del repositorio:

- URL del Repositorio GitHub donde se elaboró el trabajo del laboratorio.
- <https://github.com/ELGRANn/Pweb-6.git>
- URL personal de cada integrante del grupo.

- Miguel Angel Alvarez Choque:  
• [https://github.com/miguelnodjan/pw2\\_24a.git](https://github.com/miguelnodjan/pw2_24a.git)
- Eduardo Joel Cuno Salazar:  
• <https://github.com/ELGRANn/pw2-24a.git>
- Rodrigo Alexander Fernández Huarca:  
• <https://github.com/RdrigoFH/pw2-24a.git>
- Jose Maria Ticona Saure:  
• <https://github.com/joseticonasaure/pw2-24a.git>

## 1.6. Estructura de laboratorio 7

```
+-----laboratorio_7
|   |- db.sqlite3
|   |- manage.py
|
|-----laboratorio_7
|   |- asgi.py
|   |- settings.py
|   |- urls.py
|   |- wsgi.py
|   |- __init__.py
|   |-
|       -----__pycache__
|
|
|
|
|-----notas
|   |- admin.py
|   |- apps.py
|   |- models.py
|   |- tests.py
|   |- views.py
|   |- __init__.py
|   |-
|       -----migrations
|
|
|-----__pycache__
```

## 2. Rúbricas

### 2.1. Entregable Informe

Tabla 2: Tipo de Informe

<b>Informe</b>		
<b>Latex</b>	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.	<b>Nota</b>
<b>Observaciones</b>	Respetar la estructura de organización para la ubicación de los entregables. Por cada observación dentro del informe se le descontará puntos. Se debe incluir el código fuente latex del informe	

### 2.2. 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 3: Niveles de desempeño

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

Tabla 4: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
<b>1. GitHub</b>	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	1	
<b>2. Commits</b>	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
<b>3. Código fuente</b>	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
<b>4. Ejecución</b>	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
<b>5. Pregunta</b>	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
<b>6. Fechas</b>	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
<b>7. Ortografía</b>	El documento no muestra errores ortográficos.	2	X	1	
<b>8. Madurez</b>	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	2	
<b>Total</b>		20		12	