

Informe de Laboratorio 08

Tema: Django: Uno a muchos, muchos a muchos, impresión de pdf y emails

Nota

Estudiante	Escuela	Asignatura
- David Alfredo Huamani Ollachica - Marco Antonio Suarez Huamani - Rafael Diego Nina Caliza-ya - Angel Paul Apaza Nazareth	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 1702122

Laboratorio	Tema	Duración
08	Django: Uno a muchos, muchos a muchos, impresión de pdf y emails	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 20 Junio 2024	Al 24 Junio 2024

1. Tarea

- URL GitHub del Proyecto Django https://github.com/DrN25/pw2_lab08.git

2. Ejercicios Propuestos

Se trabajó un proyecto Django en el cual se trabaja con las relaciones de uno a muchos, de muchos a muchos, impresión de pdfs y envío de emails. Para ello, tenemos los siguientes modelos para cada uno de estos puntos.

2.1. Modelos de Relación: De Uno a Muchos

- El código define dos modelos en Django para gestionar información de proyectos y tareas creadas usando la relación de Uno a Muchos. A continuación, se describen brevemente los modelos:
- **Modelo Proyecto:**
 - nombre: El nombre del proyecto.

- **descripcion:** La descripción del proyecto.
- **fecha_inicio:** La fecha de inicio del proyecto.
- **fecha_fin:** La fecha de fin del proyecto.

■ **Modelo Tarea:**

- **proyecto:** Una clave foránea que referencia al modelo Proyecto.
- **nombre:** El nombre de la tarea.
- **descripcion:** La descripción de la tarea.
- **fecha_creacion:** La fecha de creación de la tarea.
- **fecha_vencimiento:** La fecha de vencimiento de la tarea.
- **estado:** El estado de la tarea, con opciones para elegir.

Cada modelo también tiene un método que devuelve una representación en cadena de texto del objeto.

Listing 1: models.py

```
class Proyecto(models.Model):
    nombre = models.CharField(max_length=255)
    descripcion = models.TextField()
    fecha_inicio = models.DateField()
    fecha_fin = models.DateField()

    def __str__(self):
        return self.nombre

class Tarea(models.Model):
    proyecto = models.ForeignKey(Proyecto, related_name='tareas',
                                on_delete=models.CASCADE)
    nombre = models.CharField(max_length=255)
    descripcion = models.TextField()
    fecha_creacion = models.DateField(auto_now_add=True)
    fecha_vencimiento = models.DateField()
    estado = models.CharField(max_length=50, choices=[('pendiente', 'Pendiente'),
                                                    ('en_progreso', 'En Progreso'), ('completada', 'Completada')])

    def __str__(self):
        return self.nombre
```

2.2. Modelos de Relación: De Muchos a Muchos

- El código define dos modelos en Django para gestionar información de trabajadores y tareas-trabajador creadas usando la relación de Muchos a Muchos. A continuación, se describen brevemente los modelos:
- **Modelo Trabajador:**
 - **nombre:** El nombre del trabajador.
 - **correo:** El correo del trabajador.
- **Modelo TareaTrabajador:**
 - **tarea:** Una clave foránea que referencia al modelo Tarea.

- trabajador: Una clave foránea que referencia al modelo Trabajador.

Cada modelo también tiene un método que devuelve una representación en cadena de texto del objeto.

Listing 2: models.py

```
class Trabajador(models.Model):
    nombre = models.CharField(max_length=100)
    correo = models.EmailField(unique=True)

    def _str_(self):
        return self.nombre

class TareaTrabajador(models.Model):
    tarea = models.ForeignKey(Tarea, related_name='trabajadores',
                             on_delete=models.CASCADE)
    trabajador = models.ForeignKey(Trabajador, related_name='tareas_asignadas',
                                   on_delete=models.CASCADE)

    class Meta:
        unique_together = ('tarea', 'trabajador')

    def _str_(self):
        return f'Tarea: {self.tarea.nombre} - Trabajador: {self.trabajador.nombre}'
```

3. Diseño de la Página Web

3.1. Capturas de pantalla de las tablas y la página web:

Gestión de Proyectos [Inicio](#) [Tareas](#) [Trabajadores](#) [Tareas de Trabajadores](#)

Proyectos

[Agregar Nuevo](#)
[Enviar Correo a Trabajadores](#)
[Generar PDF](#)

ID	Nombre	Descripción	Fecha de Inicio	Fecha de Fin	Acciones
1	Muebles	hacer muebles de todo tipo	June 10, 2024	June 29, 2024	Eliminar
37	Marco Antonio	sdsdsd	June 12, 2024	June 29, 2024	Eliminar

Figura 1: Página de los proyectos creados.

Gestión de Proyectos [Inicio](#) [Tareas](#) [Trabajadores](#) [Tareas de Trabajadores](#)

Tareas

hacer un ropero

Figura 2: Página de las tareas.

Gestión de Proyectos [Inicio](#) [Tareas](#) [Trabajadores](#) [Tareas de Trabajadores](#)

Trabajadores

Marco Antonio

Figura 3: Página de los trabajadores creados.

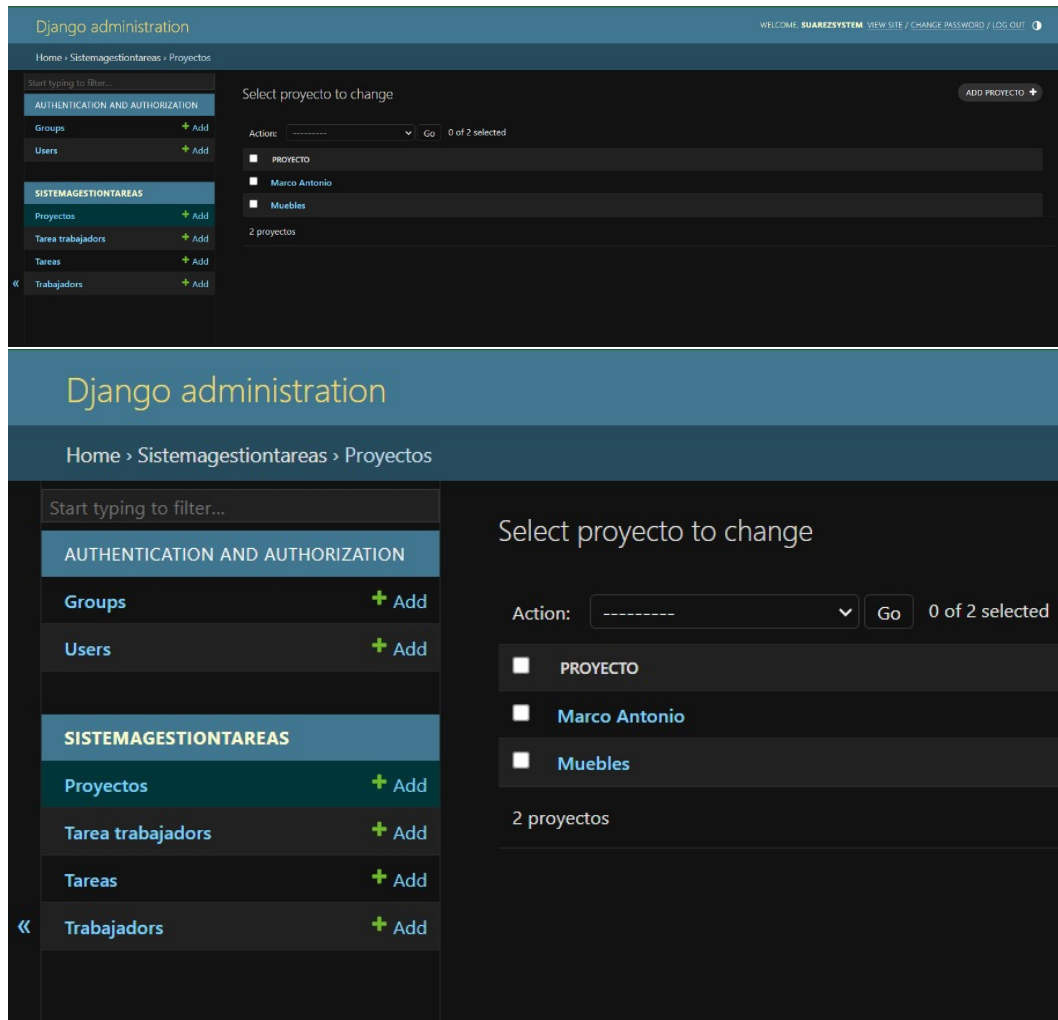


Figura 4: Página admin de las tablas creadas.

4. Rúbricas

4.1. Sobre el 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.

4.2. 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	1	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	1	
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	2	
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 estan dentro de los plazos de fecha de entrega establecidos.	2	X	1.5	
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	3	
Total		20		14.5	

5. Referencias

- https://docs.djangoproject.com/en/5.0/topics/db/examples/many_to_one/
- <https://docs.djangoproject.com/>
- <https://www.w3schools.com/python/>
- <https://www.w3schools.com/django/>