

tercera parte

Proyecto: Centro de Idiomas

lenguaje: Python.

Framework: Django.

Editor: VS code.

Carpeta del Proyecto: UIII_Centro_de_Idiomas_0498

Proyecto:backend_Idiomas

aplicación: app_Idiomas

1 Aquí el modelo models.py que ya está creado

```
# =====
# MODELO: CLASE
# =====
```

```
class Clase(models.Model):
```

```
    nombre = models.CharField(max_length=150)
```

```
    descripcion = models.TextField(blank=True, null=True)
```

```
    capacidad = models.PositiveIntegerField()
```

```
    nivel = models.CharField(max_length=50,
```

```
        choices=[ ('Básico A1', 'Básico A1'), ('Básico A2', 'Básico A2'), ('Intermedio B1', 'Intermedio B1'), ('Intermedio B2', 'Intermedio B2'), ('Avanzado C1', 'Avanzado C1'), ('Avanzado C2', 'Avanzado C2') ])
```

```
    id_profesor = models.ForeignKey(Profesor, on_delete=models.CASCADE,
related_name="clases")
```

```
    idioma = models.ForeignKey(Idioma, on_delete=models.CASCADE,
related_name="clases_idioma")
```

```
def __str__(self):
```

```
    return f"{self.nombre} ({self.idioma.nombre})"
```

=====

2 Procedimiento para realizar las migraciones(makemigrations y migrate).

3 Ahora trabajamos con el MODELO: Clases

4 En view de app_Idiomas crear las funciones con sus códigos correspondientes (agregar_clase, actualizar_clase, realizar_actualizacion_clase borrar_clase)

5 En la opción de agregar_clase, la tabla clases está relacionada con Idiomas y Profesores, primero se captura los datos no relacionados de clases y luego en los campos de id_profesor debe aparecer un combobox con los profesores por su nombre y poder seleccionar una de ellas, en el campo idioma debe aparecer un combobox con los idiomas disponibles por su nombre y poder seleccionar uno de ellos

6 Modificar el archivo navbar.html que está dentro de templates, para actualizar la opción "clases" en submenu de Clases(Agregar clase, ver clases en forma de tabla, actualizar clase, borrar clase)

7 Crear la subcarpeta clases dentro de app_Idiomas\templates.

8 crear los archivos html con su código correspondientes de
(agregar_clase.html, ver_clases.html mostrar en tabla con
los botones ver, editar y borrar, actualizar_clase.html,
borrar_clase.html) dentro de app_Idioma\templates\clases

9 No utilizar forms.py.

10 procedimiento para agregar en urls.py en app_Idiomas con
el código correspondiente para acceder a las funciones de views.py
para operaciones de crud en película.

10 procedimiento para registrar los modelos en admin.py y volver a
realizar las migraciones.

11 trabajando con todos los modelos

12 Utilizar colores suaves, atractivos y modernos, el código de las
páginas web sencillas para clases.

13 No validar entrada de datos.

14 Al inicio crear la estructura completa de carpetas y archivos actualizados, donde se
muestra la carpeta películas dentro de templates con los archivos html correspondientes a
las operaciones crud clases.

15 proyecto totalmente funcional.

16 finalmente ejecutar servidor en el puerto puerto 8498.