

UIII Act 4 Tres tablas del negocio Investech

Segunda parte

- **Proyecto:** Investech
 - **Lenguaje:** Python
 - **Framework:** Django
 - **Editor:** VS Code
 - **Carpeta del Proyecto:** UIII_investech_0640
 - **Proyecto:** backend_investech
 - **Aplicación:** app_investech
-

Procedimientos

1. Aquí el modelo **models.py** que ya está creado

```
class Portafolio(models.Model):
    usuario = models.ForeignKey(Usuario, on_delete=models.CASCADE, related_name="portafolios")
    # 1 a muchos
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField(blank=True)
    valor_total = models.DecimalField(max_digits=12, decimal_places=2, default=0.00)
    fecha_creacion = models.DateTimeField(auto_now_add=True)
    riesgo = models.CharField(max_length=50, choices=[('bajo', 'Bajo'), ('medio', 'Medio'), ('alto', 'Alto')])
    activos = models.ManyToManyField('Activo', related_name="portafolios") # muchos a muchos

    def __str__(self):
        return f'{self.nombre} ({self.usuario.nombre})'
```

2. Procedimiento para realizar las **migraciones** (makemigrations y migrate)
3. Ahora trabajamos con el **MODELO: Portafolio**
4. En **views.py** de **app_investech**, crear las funciones con sus códigos correspondientes:
 - `agregar_portafolio`
 - `actualizar_portafolio`
 - `realizar_actualizacion_portafolio`
 - `borrar_portafolio`
5. Modificar el archivo **navbar.html** (dentro de *templates*) para actualizar la opción **"Portafolio"**, incluyendo el submenú:
 - Agregar portafolio
 - Ver portafolio
 - Actualizar portafolio
 - Borrar portafolio

6. Crear la subcarpeta **portafolio** dentro de *app_investech\templates*
7. Dentro de *app_investech\templates\sala*, crear los archivos HTML con su código correspondiente:
 - agregar_portafolio.html
 - ver_portafolio.html (mostrar en tabla con los botones ver, editar y borrar)
 - actualizar_portafolio.html
 - borrar_portafolio.html
8. **No utilizar forms.py**
9. Procedimiento para agregar en **urls.py** de *app_investech* el código correspondiente para acceder a las funciones de *views.py* (operaciones CRUD en salas)
10. Procedimiento para **registrar los modelos en admin.py** y volver a realizar las migraciones
11. Por lo pronto, solo trabajar con “**portafolio**”; dejar pendiente el **MODELO: Activo**
12. Utilizar **colores suaves, atractivos y modernos** en el código de las páginas web sencillas para *salas*
13. **No validar entrada de datos**
14. Al inicio, crear la **estructura completa de carpetas y archivos actualizados**, mostrando la carpeta *salas* dentro de *templates* con los archivos HTML correspondientes a las operaciones CRUD de salas
15. **Proyecto totalmente funcional**
16. Finalmente, **ejecutar el servidor en el puerto 8640**