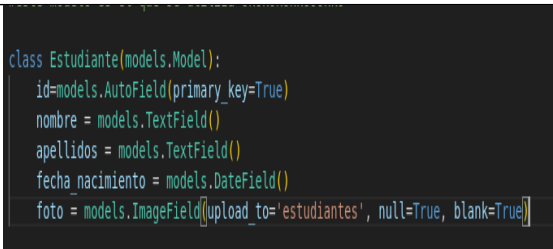


UNIVERSIDAD TÉCNICA DE COTOPAXI
MANUAL DE USUARIO
FACULTAD CIYA
CARRERA SISTEMAS DE INFORMACIÓN
DESARROLLO Y APLICACIONES WEB
MANUAL DE USUARIO
CRUD CON AJAX-FETCH-MODALES

1. Ingresamos a nuestro apartado de **models**. Realizaremos un nuevo models "Estudiante".

Código	Imagen
Models Estudiante	
<pre>class Estudiante(models.Model): id=models.AutoField(primary_key=True) nombre = models.TextField() apellidos = models.TextField() fecha_nacimiento = models.DateField() foto = models.ImageField(upload_to='estudiantes' , null=True, blank=True)</pre>	

2. Realizamos las migraciones.

Python manage.py makemigration

Python manage.py migrate

3. Ingresamos en los views.py para poder realizar las funciones del CRUD.

Código	Imágenes
Los de views estudiantes	

```
def estudiantes(request):
    return render(request, 'estudiantes.html')
```

```
def listadoEstudiantes(request):
    estudiantes =
list(Estudiante.objects.values())
    return JsonResponse({'estudiantes':
estudiantes})
```

```
def nuevoEstudiante(request):
    nombre = request.POST.get('nombre')
    apellidos = request.POST.get('apellidos')
    fecha_nacimiento =
request.POST.get('fecha_nacimiento')
    foto = request.FILES.get('foto')
```

```
    est = Estudiante.objects.create(
        nombre=nombre,
        apellidos=apellidos,
        fecha_nacimiento=fecha_nacimiento,
        foto=foto
    )
```

```
    return JsonResponse({'mensaje':
'Eudiante creado'})
```

```
def editarEstudiante(request, id):
    if request.method == 'POST':
        foto=request.FILES.get('foto')
        est = Estudiante.objects.get(id=id)
        est.nombre =
request.POST.get('nombre')
        est.apellidos =
request.POST.get('apellidos')
        est.fecha_nacimiento =
request.POST.get('fecha_nacimiento')
        if foto:
            if request.FILES.get('foto'):
                est.foto.delete(save=False)
                est.foto=foto
            est.save()
```

```
    return JsonResponse({'mensaje':
'Eudiante actualizado'})
```

```
def eliminarEstudiante(request, id):
    if request.method == 'POST':
        est = Estudiante.objects.get(id=id)
        if est.foto:
```

views.py X

Aplicaciones > Empresas > views.py > ...

```
101 """*CRUD DE AJAX ESTUDIANTES """
```

```
102
```

```
103 def estudiantes(request):
```

```
104 |     return render(request, 'estudiantes.html')
```

```
105
```

```
106
```

```
107 def listadoEstudiantes(request):
```

```
108 |     estudiantes = list(Estudiante.objects.values
```

```
109 |     return JsonResponse({'estudiantes': estudiant
```

```
110
```

```
111
```

```
112 def nuevoEstudiante(request):
```

```
113 |     nombre = request.POST.get('nombre')
```

```
114 |     apellidos = request.POST.get('apellidos')
```

```
115 |     fecha_nacimiento = request.POST.get('fecha_na
```

```
116 |     foto = request.FILES.get('foto')
```

```
117
```

```
118 |     est = Estudiante.objects.create(
```

```
119 |         nombre=nombre,
```

```
120 |         apellidos=apellidos,
```

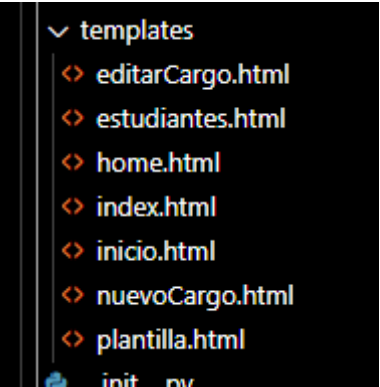
```
121 |         fecha_nacimiento=fecha_nacimiento,
```

```
122 |         foto=foto
```

```
123 |     )
```

<pre> est.foto.delete(save=False) est.delete() return JsonResponse({'mensaje': 'Estudiante eliminado'}) </pre>	

4. Dentro del templates vamos a ingresar un new file y crearemos uno nuevo llamado estudiantes.html



5. Dentro del index realizamos el llamado a la plantilla

Código	Imagen
<pre> {% extends './plantilla.html' %} {% block contenido %} {% endblock %} </pre>	

Dentro del bloque de contenido.

Código	Imagen
1. Visualización de datos	

<pre>

 <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> <title>Document</title> </head> <body> <h1 style="text-align:center">Listado de estudiantes </h1> <!--</a--> <button onclick="abrirModal()" class="btn btn-success">Registrar nuevo Estudiante</button> <div> <table id="tblEstudiantes"> <thead> <tr> <td>ID</td> <td>NOMBRES</td> <td>APELLIDOS</td> <td>FECHA NACIMIENTO</td> <td>FOTO</td> <td>ACCIONES</td> </tr> </thead> <tbody id="tablaEstudiantes"></tbody> </table> </div></pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <h1 style="text-align:center">Listado de estudiantes </h1> <!--</a--> <button onclick="abrirModal()" class="btn btn-success">Registrar nuevo </button> <div> <table id="tblEstudiantes"> <thead> <tr> <td>ID</td> <td>NOMBRES</td> <td>APELLIDOS</td> <td>FECHA NACIMIENTO</td> <td>FOTO</td></pre>
2. Crear un nuevo estudiante	primer modal

```

<div class="modal fade"
id="modalEstudiante" tabindex="-1">
  <div class="modal-dialog">

    <form id="formEstudiante"
class="modal-content"
enctype="multipart/form-data">
      <div class="modal-header">
        <h5 class="modal-
title">Persona</h5>
      </div>

      <div class="modal-body">
        <input type="hidden"
id="personaId" name="personaId">
        <div class="mb-3">
          <label for="nombre"
class="form-label">Nombre</label>
          <input type="text" id="nombre"
name="nombre" class="form-control" >
        </div>
        <div class="mb-3">
          <label for="apellidos"
class="form-label">Apellidos</label>
          <input type="text"
id="apellidos" name="apellidos" class="form-
control" >
        </div>
        <div class="mb-3">
          <label for="fecha_nacimiento"
class="form-label">Fecha de
Nacimiento</label>
          <input type="date"
id="fecha_nacimiento"
name="fecha_nacimiento" class="form-
control" >
        </div>
        <div class="mb-3">
          <label for="foto" class="form-
label">Foto</label>
          <input type="file" id="foto"
name="foto" class="form-control"
accept="image/*">
        </div>
      </div>


      <div class="modal-footer">
        <button type="submit" class="btn
btn-success">Guardar</button>
        <button type="button" class="btn
btn-secondary" data-bs-

```

```

index.html 7 X
Aplicaciones > Empresas > templates > index.html > div#modalPersona.modal.fade > div.modal-dialog
33 <!-- Modal -->
34 <div class="modal fade" id="modalPersona" tabindex="-1">
35   <div class="modal-dialog">
36     <form id="formPersona" class="modal-content" enctype="multipart/form-data">
37       <div class="modal-header">
38         <h5 class="modal-title" id="tituloModal">Nueva Persona</h5>
39       </div>
40       <div class="modal-body">
41         <input type="hidden" id="personaId" name="personaId">
42         <input type="hidden" id="accion" name="accion">
43         <div class="mb-3">
44           <label for="nombre" class="form-label">Nombre</label>
45           <input type="text" id="nombre" name="nombre" class="form-control">
46         </div>
47         <div class="mb-3">
48           <label for="apellidos" class="form-label">Apellidos</label>
49           <input type="text" id="apellidos" name="apellidos" class="form-control">
50         </div>
51         <div class="mb-3">
52           <label for="fecha_nacimiento" class="form-label">Fecha de nacimiento</label>
53           <input type="date" id="fecha_nacimiento" name="fecha_nacimiento" class="form-control">
54         </div>
55         <div class="mb-3">
56           <label for="foto" class="form-label">Foto</label>

```

<pre>dismiss="modal">Cancelar</button> </div> </form> </div> </div></pre>	
<p>3. Para los script</p> <pre><script> function getCookie(name) { let cookieValue = null; if (document.cookie && document.cookie !== '') { const cookies = document.cookie.split(';'); for (let i = 0; i < cookies.length; i++) { const cookie = cookies[i].trim(); // Coincide el nombre if (cookie.substring(0, name.length + 1) === (name + '=')) { cookieValue = decodeURIComponent(cookie.substring(nam e.length + 1)); break; } } return cookieValue; } const csrftoken = getCookie('csrftoken'); } </script> </body> </html></pre>	 <pre>estudiantes.html X Aplicaciones > Empresas > templates > estudiantes.html > html 5 <html lang="en"> 11 <body> 34 <div class="modal fade" id="modalEstudiante"> 35 <div class="modal-dialog"> 68 </div> 69 </div> 70 71 <script> 72 function getCookie(name) { 73 let cookieValue = null; 74 if (document.cookie && document.cookie !== '') { 75 const cookies = document.cookie.split(';'); 76 for (let i = 0; i < cookies.length; i++) { 77 const cookie = cookies[i].trim(); 78 // Coincide el nombre 79 if (cookie.substring(0, name.length + 1) === (name + '=')) { 80 cookieValue = decodeURIComponent(cookie.substring(name.length + 1)); 81 break; 82 } 83 } 84 return cookieValue; 85 } 86 const csrftoken = getCookie('csrftoken');</pre>
<p>4. Funcionalidades modales</p>	<ul style="list-style-type: none"> - Se abre el modal con campos vacíos, agregar - Usuario hace clic en "Editar" de una persona específica - Aparece una confirmación: "¿Seguro que deseas eliminar esta persona?" - El código incluye validaciones automáticas que verifican.

```

<script>

document.addEventListener('DOMContentLoaded', () => {
    listadoEstudiantes();

    // Listener para el formulario

document.getElementById('formEstudiante').
addEventListener('submit', function (e) {
    e.preventDefault();

    const id =
document.getElementById('personaId').value;
    const form = this;
    const formData = new
FormData(form);

    if (id) {
        // Si tiene id, actualizamos
        actualizarEstudiante(id, formData);
    } else {
        // Si no tiene id, creamos nuevo
        estudiante
        nuevoEstudiante(formData);
    }
});

});

function listadoEstudiantes() {
    fetch('/listadoEstudiantes')
    .then(response => response.json())
    .then(data => {
        const tabla =
document.getElementById('tablaEstudiantes')
;

        tabla.innerHTML = "";

        data.estudiantes.forEach(estu => {
            tabla.innerHTML += `
                <tr>
                    <td>${estu.id}</td>
                    <td>${estu.nombre}</td>
                    <td>${estu.apellidos}</td>
                    <td>${estu.fecha_nacimiento}
</td>
                    <td>${estu.foto ? `` : 'Sin
foto'}</td>

                    <td>

```

```

78
79 <script>
80     const modal = new bootstrap.Modal(document.getE
81
82     function abrirModalCrear() {
83         document.getElementById('formPersona').rese
84         document.getElementById('personaId').value
85         document.getElementById('accion').value = '
86         document.getElementById('tituloModal').text
87         modal.show();
88     }
89
90     function abrirModalEditar(id, nombre, apellidos
91         document.getElementById('personaId').value
92         document.getElementById('nombre').value = n
93         document.getElementById('apellidos').value
94         document.getElementById('fecha_nacimiento')
95         document.getElementById('foto').value = '';
96         document.getElementById('accion').value = '
97         document.getElementById('tituloModal').text
98         modal.show();
99     }

```

```

        <button class="btn btn-sm
btn-warning" onclick="editar($
{estu.id})">Editar</button>
        <button class="btn btn-sm
btn-danger" onclick="eliminar($
{estu.id})">Eliminar</button>
    </td>
</tr>
</tr>
    `;
    });
    });
}

function abrirModal() {
    const modal = new
bootstrap.Modal(document.getElementById('
modalEstudiante'));
    modal.show();
}

function nuevoEstudiante(formData) {
    fetch('/nuevoEstudiante/', {
        method: 'POST',
        headers: {
            'X-CSRFToken': csrftoken
        },
        body: formData
    })
    .then(res => res.json())
    .then(data => {
        console.log(data.mensaje);
        cerrarModal();
    });
}

function editar(id) {
    fetch(`/listadoEstudiantes`)
    .then(res => res.json())
    .then(data => {
        const estudiante =
data.estudiantes.find(e => e.id === id);
        if (estudiante) {

document.getElementById('personalId').value
= estudiante.id;

document.getElementById('nombre').value =
estudiante.nombre;

document.getElementById('apellidos').value =
estudiante.apellidos;

```



```

document.getElementById('fecha_nacimiento').value = estudiante.fecha_nacimiento;
    abrirModal();
    }
    });
}

function actualizarEstudiante(id, formData)
{
    fetch(`/editarEstudiante/${id}/`, {
        method: 'POST',
        headers: {
            'X-CSRFToken': csrftoken
        },
        body: formData
    })
    .then(res => res.json())
    .then(data => {
        console.log(data.mensaje);
        cerrarModal();
    });
}

function cerrarModal() {
    listadoEstudiantes();
    const modal =
bootstrap.Modal.getInstance(document.getEl
ementById('modalEstudiante'));
    modal.hide();

document.getElementById('formEstudiante').
reset();

document.getElementById('personalId').value
= "";
}

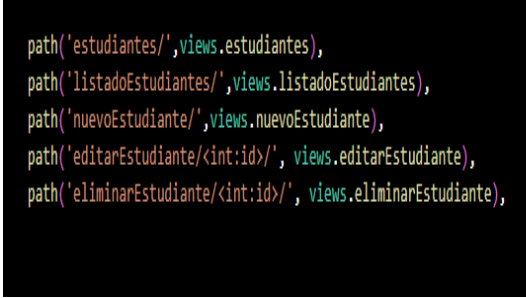
function eliminar(id) {
    if (confirm('¿Seguro que deseas eliminar
este estudiante?')) {
        fetch(`/eliminarEstudiante/${id}/`, {
            method: 'POST',
            headers: {
                'X-CSRFToken': csrftoken
            }
        })
        .then(res => res.json())
        .then(data => {
            console.log(data.mensaje);

```

<pre> listadoEstudiantes(); }); } } </script> </pre>	
5. Manejo de data tables	
<pre> <script> let table = new DataTable('#tblEstudiantes', { layout: { topStart: { buttons: ['copy', 'csv', 'excel', 'pdf', 'print'] } } }); </script> </pre>	<pre> <script> \$("#foto").fileinput({ language: "es", allowedFileExtensions: ["png jpg jpeg"], showCaption: false, dropZoneEnabled: true, showClose: false }); </script> </pre>
6. Manejo de imágenes	
<pre> <script> \$("#foto").fileinput({ language: "es", allowedFileExtensions: ["png jpg jpeg"], showCaption: false, dropZoneEnabled: true, showClose: false }); </script> </pre>	
7. Validaciones de campos	
<pre> <script> \$("#formEstudiante").validate({ rules: { "nombre": { required: true, minlength: 3 }, "apellidos": { required: true, minlength: 3 }, "fecha_nacimiento": { required: true, }, "foto": { required: true, } }, messages: { "nombre": { </pre>	<pre> </script> <script> \$("#formEstudiante").validate rules: { "nombre": { required: true, minlength: 3 }, "apellidos": { required: true, minlength: 3 }, "fecha_nacimiento": { required: true, }, "foto": { required: true, } } </pre>

<pre> required: "Por favor, ingresa el nombre", minlength: "Debe tener al menos 3 caracteres" }, "apellidos": { required: "Por favor, ingresa los apellidos", minlength: "Debe tener al menos 3 caracteres" }, "fecha_nacimiento": { required: "Selecciona la fecha de nacimiento", }, "foto": { required: "Selecciona una imagen", } } }); </script> </pre>	
--	--

6. Ingresamos a hacer el llamado a las urls.py

codigo	Imagen
<pre> path('estudiantes/',views.estudiantes), path('listadoEstudiantes/',views.listadoEstudia ntes), path('nuevoEstudiante/',views.nuevoEstudiant e), path('editarEstudiante/<int:id>/', views.editarEstudiante), path('eliminarEstudiante/<int:id>/', views.eliminarEstudiante), </pre>	 <pre> path('estudiantes/',views.estudiantes), path('listadoEstudiantes/',views.listadoEstudiantes), path('nuevoEstudiante/',views.nuevoEstudiante), path('editarEstudiante/<int:id>/', views.editarEstudiante), path('eliminarEstudiante/<int:id>/', views.eliminarEstudiante), </pre>

Ventajas de usar AJAX en Django

- La página no se recarga completamente.
- Es más rápido y fluido para el usuario.
- Solo se actualiza la parte que cambio.