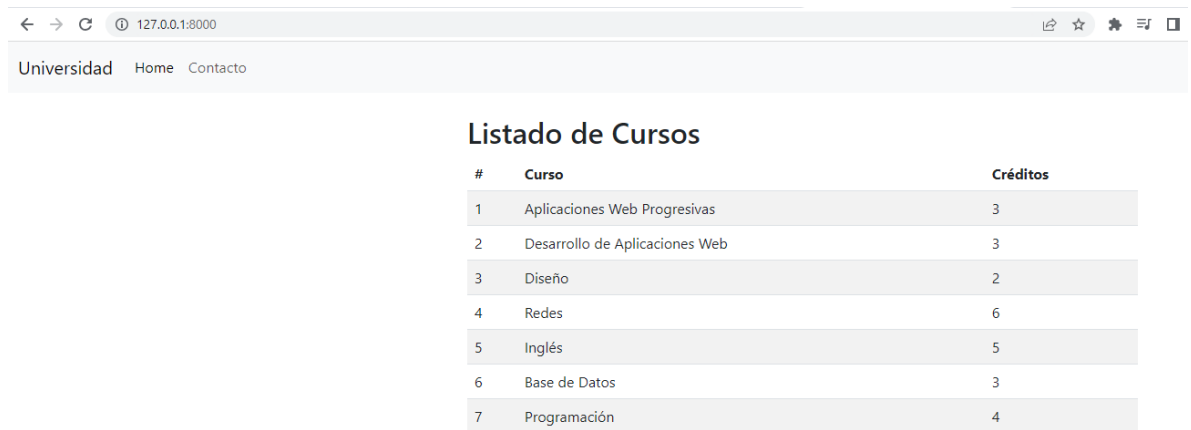


14- Django (CRUD: Agregar y Borrar)

1. Al momento principal, haciendo uso de nuestra plantilla, la página principal es la siguiente:



#	Curso	Créditos
1	Aplicaciones Web Progresivas	3
2	Desarrollo de Aplicaciones Web	3
3	Diseño	2
4	Redes	6
5	Inglés	5
6	Base de Datos	3
7	Programación	4

2. Se agregarán las opciones de Editar y Eliminar en la tabla. Ubica el archivo cursos.html y agrega las siguientes líneas:

- Al encabezado:

```
<th colspan="2">Opciones</th>
```

- Y al cuerpo de la tabla

```
<td><a href="edicionCurso/{{curso.id}}" class="btn btn-sm btn-block btn-info"><i class="fas fa-pencil-alt"></i> Editar</a></td>

<td><a href="eliminarCurso/{{curso.id}}" class="btn btn-sm btn-block btn-danger btnEliminacion"><i class="far fa-trash-alt"></i> Eliminar</a></td>
```

Se visualiza de la siguiente manera:

Listado de Cursos

#	Curso	Créditos	Opciones	
1	Aplicaciones Web Progresivas	3	Editar	Eliminar
2	Desarrollo de Aplicaciones Web	3	Editar	Eliminar
3	Diseño	2	Editar	Eliminar

Cuando se pasa el cursor sobre el botón, se visualiza el id del curso



ELIMINAR

En cada botón indicamos una ruta, entonces esas rutas deben definirse

```
<td>{{forloop.counter}}</td>
<td>{{curso.nombre}}</td>
<td>{{curso.credits}}</td>
<td><a href="edicionCurso/{{curso.id}}" class="btn btn-primary">Editar</a>
<td><a href="eliminacionCurso/{{curso.id}}" class="btn btn-danger">Eliminar</a>
```

Ubica el archivo urls agregamos esa ruta

```
path('eliminarCurso/<int:id>', eliminar_curso)
```

En el archivo views.py agrega la siguiente vista

- Recibimos el id y obtenemos el curso cuyo id corresponde
- Se llama a la función delete() para borrarlo

```
def eliminar_curso(request, id):
    curso = Curso.objects.get(id=id)
    curso.delete()

    return redirect('/')
```

Nos permite redireccionar a la ruta raíz, para poder usarla hay que importarla

```
1 from django.shortcuts import render, redirect
```

3. Realiza una prueba de eliminar un curso.

Listado de Cursos

#	Curso	Créditos	Opciones	
1	Aplicaciones Web Progresivas	3	<button>Editar</button>	<button>Eliminar</button>
2	Desarrollo de Aplicaciones Web	3	<button>Editar</button>	<button>Eliminar</button>
3	Diseño	2	<button>Editar</button>	<button>Eliminar</button>
4	Redes	6	<button>Editar</button>	<button>Eliminar</button>
5	Base de Datos	3	<button>Editar</button>	<button>Eliminar</button>
6	Programación	4	<button>Editar</button>	<button>Eliminar</button>

Se eliminó un curso, ahora solo quedan 7.

INSERTAR

4. Para insertar cursos agregaremos un formulario

- El formulario se enviará a la ruta '/registrarCurso' usando POST

```
<div class="col-md-4">
  <h2>Nuevo Curso</h2>
  <div class="card">
    <div class="card-body">
      <form id="formularioCurso" action="/registrarCurso/" method="POST">
        {% csrf_token %}
        <div class="form-group">
          <input type="text" id="txtNombre" name="txtNombre" class="form-control" minlength="3" maxlength="30" placeholder="Nombre" autofocus required />
        </div>
        <div class="form-group">
          <input type="number" id="numCreditos" name="numCreditos" class="form-control" min="1" max="10" step="1" value="1" required />
        </div>
        <div class="form-group">
          <button type="submit" class="btn btn-success btn-block text-white"><i class="far fa-save"></i> Guardar</button>
        </div>
      </form>
    </div>
  </div>
</div>
```

Se agregan 2 input, uno para el nombre del curso y otro para el número de créditos. Y el botón de tipo submit.

Se recomienda que el formulario contenga validaciones, en este ejemplo, el input tiene algunas validaciones:

Tiene un mínimo, un máximo(el mismo que la base de datos), autofocus para que se posicione en el input y required para que no se permitan valores vacíos.

```
class="form-control" minlength="3" maxlength="30" placeholder="Nombre" autofocus required />
```

Y en el campo de créditos el mínimo es 1, el máximo será 10, step es para indicar de cuanto en cuando aumentará el valor y también es required.

```
class="form-control" min="1" max="10" step="1" value="1" required />
```

Se visualiza de la siguiente manera:

Universidad Home Contacto

Nuevo Curso

Guardar

Listado de Cursos

#	Curso	Créditos	Opciones	
1	Aplicaciones Web Progresivas	3	Editar	Eliminar
2	Desarrollo de Aplicaciones Web	3	Editar	Eliminar
3	Diseño	2	Editar	Eliminar
4	Redes	6	Editar	Eliminar
5	Base de Datos	3	Editar	Eliminar
6	Programación	4	Editar	Eliminar

Para que funcione, debemos agregar la ruta en el archivo urls.py

```
path('registrarCurso/', registrar_curso)
```

Y debemos crear esa vista en el archivo de views.py

```
def registrar_curso(request):
    nombre = request.POST['txtNombre']
    credits = request.POST['numCredits']

    curso = Curso.objects.create(nombre=nombre, credits=credits)
    return redirect('/')
```

Esta vista recibe lo que el usuario envió en el formulario

Y haciendo uso del ORM guarda los datos en la Base de datos con el método create.

Finalmente redirecciona a la misma página

En el formulario se agregó esta línea

```
<div class="card-body">
    <form id="formularioCurso" action="/registrarCurso/" method="POST">
        {% csrf_token %}
        <div class="form-group">
            <input type="text" id="txtNombre" name="txtNombre" class="f
```

¿Qué es esa línea?

El ataque Cross-Site Request Forgery (CSRF) obliga a un usuario final a ejecutar acciones no deseadas en una aplicación web en la que se ha autenticado.

El atacante utiliza el estado autenticado del usuario a su favor al cambiar la solicitud del usuario, lo que solicita a los usuarios que realicen acciones que no tienen la intención de realizar. Si el ataque tiene éxito en una cuenta administrativa, puede comprometer toda la aplicación web.

CSRF es un ataque común, por lo que Django tiene una implementación muy simple para negar este ataque.

{% csrf_token %} etiqueta que se implementa para evitar ataques maliciosos. Genera un token en el lado del servidor al representar la página y se asegura de verificar este token para ver si hay solicitudes que regresen. Si las solicitudes entrantes no contienen el token, no se ejecutan.

5. Realiza pruebas y agrega varios cursos

Nuevo Curso

Guardar

Listado de Cursos

#	Curso	Créditos	Opciones	
1	Aplicaciones Web Progresivas	3	<button>Editar</button>	<button>Eliminar</button>
2	Desarrollo de Aplicaciones Web	3	<button>Editar</button>	<button>Eliminar</button>
3	Diseño	2	<button>Editar</button>	<button>Eliminar</button>
4	Base de Datos	3	<button>Editar</button>	<button>Eliminar</button>
5	Programación	4	<button>Editar</button>	<button>Eliminar</button>
6	Redes	6	<button>Editar</button>	<button>Eliminar</button>
7	Desarrollo Web Integral	4	<button>Editar</button>	<button>Eliminar</button>

¿Qué pasa si agregas un curso que el nombre sean 3 espacios en blanco?

Nuevo Curso

Guardar

Listado de Cursos

#	Curso	Créditos	Opciones	
1	Aplicaciones Web Progresivas	3	<button>Editar</button>	<button>Eliminar</button>
2	Desarrollo de Aplicaciones Web	3	<button>Editar</button>	<button>Eliminar</button>
3	Diseño	2	<button>Editar</button>	<button>Eliminar</button>
4	Base de Datos	3	<button>Editar</button>	<button>Eliminar</button>
5	Programación	4	<button>Editar</button>	<button>Eliminar</button>
6	Redes	6	<button>Editar</button>	<button>Eliminar</button>
7	Desarrollo Web Integral	4	<button>Editar</button>	<button>Eliminar</button>
8	Aprende Django	5	<button>Editar</button>	<button>Eliminar</button>
9		3	<button>Editar</button>	<button>Eliminar</button>



6. Para corregir este error, usaremos javascript, ahí mismo en la plantilla en los script agrega el siguiente:

```
<script src="{% static 'js/Cursos.js' %}"></script>
```

7. Crear ese archivo dentro de la carpeta js y contiene lo siguiente:

```
const $formularioCurso = document.getElementById("formularioCurso");
const $txtNombre = document.getElementById("txtNombre");
const $numCreditos = document.getElementById("numCreditos");

(function () {

    formularioCurso.addEventListener("submit", function (e) {
        let nombre = String(txtNombre.value).trim();
        if (nombre.length < 3) {
            alert("El nombre del curso no debe ir vacio");
            e.preventDefault();
        }
    });

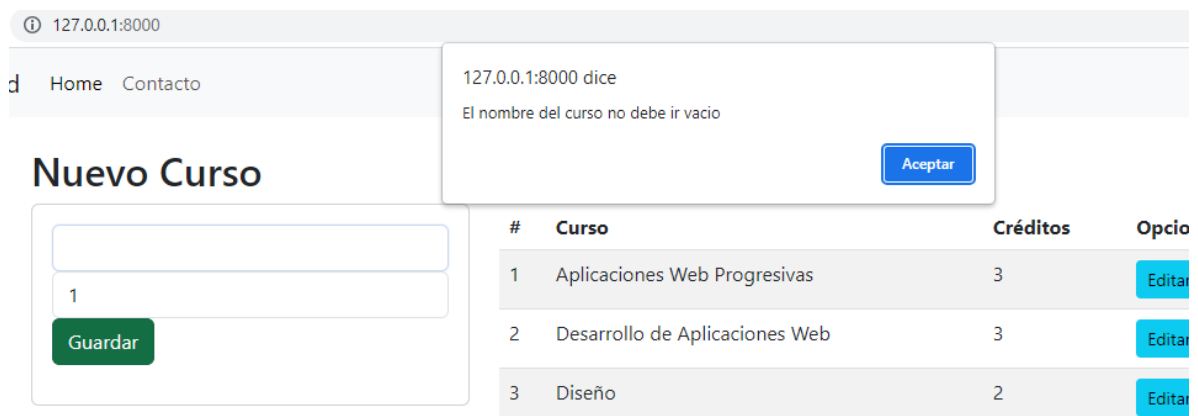
})();
```

Obtiene los elementos por id

El método trim() elimina los espacios en blanco en ambos extremos del string.

Revisa el txtNombre y su longitud, si es menor que 3 manda una alerta

8. Realiza una prueba con 3 espacios vacios:



The screenshot shows a web application interface. At the top, there's a navigation bar with 'Home' and 'Contacto'. Below it, a modal alert box is open, displaying the message '127.0.0.1:8000 dice' and 'El nombre del curso no debe ir vacio' with an 'Aceptar' button. The main content area is titled 'Nuevo Curso' and contains a form with a text input field, a number input field with the value '1', and a green 'Guardar' button. To the right of the form is a table with the following data:

#	Curso	Créditos	Opcio
1	Aplicaciones Web Progresivas	3	Editar
2	Desarrollo de Aplicaciones Web	3	Editar
3	Diseño	2	Editar

Y no permite insertar espacio ni el nombre vacio