

Informe de Laboratorio 08

Tema: Django: ORM, PDFs y correos electrónicos

Nota

Estudiante(s)	Escuela	Asignatura
Diego Alejandro Carbajal Gonzales, Jhonatan David Arias Quispe, Ricardo Mauricio Chambilla Perca, Mariel Alisson Jara Mamani mjarama@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: I Código: 1702122

Laboratorio	Tema	Duración
08	Django: ORM, PDFs y correos electrónicos	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 B	Del 19 Junio 2024	Al 23 Junio 2024
















































Laboratorio 08

1 Tarea	3
2 Commits	4
3 Equipos y materiales utilizados	5
4 Ejercicio 1: Relación: Uno a Muchos y Muchos a Muchos	6
4.1 Código	6
4.2 Ejecución	6
4.2.1 Relación de uno a muchos	6
4.2.2 Relación de muchos a muchos	7
4.3 Base de Datos	7
4.3.1 Relación de uno a muchos	7
4.3.2 Relación de muchos a muchos	7
5 Ejercicio 2: Envío de correos electrónicos	8
5.1 Código	8
5.2 Prueba	8
6 Ejercicio 3: Impresión de PDFs	8
6.1 Código	8
6.2 Prueba	11
7 Ejercicio Propuesto: Sistema de Gestión de Cursos	11
7.1 Código	11
7.1.1 Modelos	11
7.1.2 Formularios	11
7.1.3 Vistas	12
7.2 Archivos Proyecto	15
7.2.1 Urls	15
7.2.2 Settings	15
7.3 Archivos Estáticos	18
7.3.1 HTML	18
7.3.2 CSS	23
7.4 Prueba	27
7.4.1 Registro de Estudiantes	27
7.4.2 Registro de Cursos	27
7.4.3 Lista de Estudiantes	28
7.4.4 Lista de Cursos	28
7.4.5 Envío de Mensajes	28
7.4.6 Generación de Certificados	29
8 URL del repositorio en GitHub	29
9 Estructura de laboratorio 08	30
10 Rúbrica	31
11 Referencias	31

1 Tarea

- Deberán replicar la actividad de los videos donde se trabaja con Relacion de uno a muchos, de muchos a muchos, impresión de pdfs y envio de emails; y adecuarla a un proyecto en blanco Django.
- Para ello crear una carpeta dentro del proyecto github colaborativo con el docente, e informar el link donde se encuentra.
- Eres libre de agregar CSS para decorar tu trabajo.
- **Observaciones:**
 - Ya sabes que el trabajo con Git es obligatorio. Revisa los videos entregados.
 - Cada commit debe ser realizado con un mensaje descriptivo que estuvo siguiendo. Si los mensajes no son claros respecto al código “commiteado” tendrá menos calificación.

2 Commits

Correccion main jhonatanDczel committed 4 hours ago	bf86a1d	 
Se añade la vista para generar un certificado y mandarlo por correo Alan20 committed 4 hours ago	8d7c52f	 
Estilos de todas las vistas jhonatanDczel committed 5 hours ago	e3f83cb	 
Merge branch 'main' of github.com:Alan20/pw2-24a jhonatanDczel committed 5 hours ago	e7c248b	 
Estilos generales jhonatanDczel committed 5 hours ago	759bdda	 
Se creo la vista para para mandar un mensaje a un estudiante Alan20 committed 5 hours ago	6c7ace3	 
Se creo las vistas que listan a los estudiantes y a los cursos Alan20 committed 6 hours ago	34773d9	 
Se crean la pagina principal y las plantillas para crear estudiantes y cursos Alan20 committed 6 hours ago	9bb8bb8	 
Se crea los formularios para crear Estudiantes y Cursos Alan20 committed 7 hours ago	24ffc57	 
Se crea los modelos Student y Course Alan20 committed 7 hours ago	799bdf6	 
Inicializando proyecto Gestión de cursos Alan20 committed 7 hours ago	583766a	 
pdf download Alan20 committed 7 hours ago	7c5ba17	 
prueba inicial pdf Alan20 committed 7 hours ago	5859dcb	 
Se conectan las plantillas al renderizador y se realiza la prueba inicial Alan20 committed 7 hours ago	e761ff7	 
Commits on Jun 23, 2024		
Se crea la vista para ver y descargar el pdf Alan20 committed yesterday	4cc447f	 
Se crea la funcion para renderizar paginas como pdf Alan20 committed yesterday	8c82d26	 
Inicializando el proyecto impresión de pdfs Alan20 committed yesterday	bab64be	 
enviar un email con correos de gmail Alan20 committed yesterday	2f81b9c	 
Index.html (pagina de inicio) y configuracion para el email Alan20 committed yesterday	cb5ea73	 
Creando la app send_email, y configurando las uri's Alan20 committed yesterday	28c6935	 
Inicializando el proyecto sending email Alan20 committed yesterday	af19bds	 
Ejemplo de caso para conexión con otra SBD Alan20 committed yesterday	80553de	 
Relación muchos a muchos: Realizando consultas con filter, set.all, y get, cambios en la BD Alan20 committed yesterday	1edda3c	 
Relacion muchos a muchos: Creando objetos de los modelos Movie y Character, cambios en BD Alan20 committed yesterday	2cf2166	 

Lista de commits.

3 Equipos y materiales utilizados

- Cuenta en GitHub con el correo institucional.
- Sistema Operativo Microsoft Windows 10
- Visual Studio Code
- Git
- Windows PowerShell
- Python
- Navegador Mozilla Firefox

4 Ejercicio 1: Relación: Uno a Muchos y Muchos a Muchos

4.1 Código

```
1 from django.db import models
2
3 # One To Many
4 class Language(models.Model):
5     name = models.CharField(max_length=10)
6
7     def __str__(self):
8         return self.name
9
10 class Framework(models.Model):
11     name = models.CharField(max_length=10)
12     language = models.ForeignKey(Language, on_delete=models.CASCADE)
13
14     def __str__(self):
15         return self.name
16
17 # Many to Many
18 class Movie(models.Model):
19     name = models.CharField(max_length=10)
20
21     def __str__(self):
22         return self.name
23
24 class Character(models.Model):
25     name = models.CharField(max_length=10)
26     movies = models.ManyToManyField(Movie)
27
28     def __str__(self):
29         return self.name
```

Archivo models.py

4.2 Ejecución

4.2.1 Relación de uno a muchos

Se muestra las operaciones realizar con los modelos Language y Framework.

```
>>> Framework.objects.filter(language__name='Python')
<QuerySet [<Framework: Django>, <Framework: Flask>, <Framework: Bottle>]>
>>> Framework.objects.filter(language__name='Java')
<QuerySet [<Framework: Spring>]>
>>> Framework.objects.filter(language__name__startswith='Ja')
<QuerySet [<Framework: Spring>, <Framework: Angular>]>
>>> Framework.objects.filter(language__name__startswith='Javw')
<QuerySet []>
>>> Language.objects.filter(framework__name='Spring')
<QuerySet [<Language: Java>]>
>>> Language.objects.filter(framework__name='Angular')
<QuerySet [<Language: JavaScript>]>
>>> 
```

Resultado de la relación de uno a muchos.

4.2.2 Relación de muchos a muchos

Se muestra las operaciones realizar con los modelos Movie y Character.

```
>>> avengers.save()
>>> captain_america = Character(name='Captain America')
>>> captain_america.save()
>>> captain_america.movies.add(avengers)
>>> civil_war = Movie(name='Civil War')
>>> thor = Movie(name='Thor: Dark World')
>>> thor_character = Character(name='Thor')
>>> civil_war.save()
>>> thor.save()
>>> thor_character.save()
>>> captain_america.movies.add(civil_war)
>>> thor_character.movies.add(avengers)
>>> thor_character.movies.add(thor)
>>> captain_america.movies.create(name='Winter Soldier')
<Movie: Winter Soldier>
```

Resultado de la relación de muchos a muchos.

4.3 Base de Datos

4.3.1 Relación de uno a muchos

Se muestra la BD de los modelos Language y Framework.

Table:

	id	name ▼ ¹
	Filter	Filter
1	4	Java
2	3	Python

Tabla Language.

Table:

	id	name	language_id
	Filter	Filter	Filter
1	1	Django	3
2	2	Flask	3
3	3	Bottle	3
4	4	Spring	4

Tabla Framework.

4.3.2 Relación de muchos a muchos

Se muestra la BD de los modelos Movie y Character.

Table: example_movie

	id	name
	Filter	Filter
1	1	Avengers
2	2	Civil War
3	3	Thor: Dark World
4	4	Winter Soldier

Tabla Language.

Table: example_character

	id	name
	Filter	Filter
1	1	Captain America
2	2	Thor

Tabla Framework.

Table: example_character

	id	name
	Filter	Filter
1	1	Captain America
2	2	Thor

Tabla Framework.

5 Ejercicio 2: Envío de correos electrónicos

5.1 Código

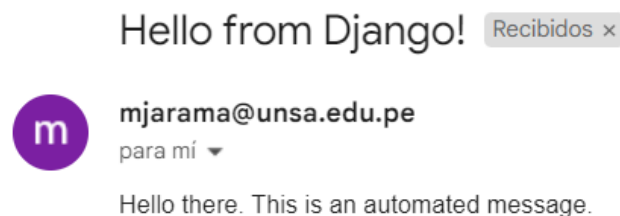
Se muestra el archivo view.py para el envío del correo.

```

1 from django.shortcuts import render
2 from django.core.mail import send_mail
3
4 def index(request):
5     send_mail('Hello from Django!',
6             'Hello there. This is an automated message.', 'mjarama@unsa.edu.pe', ['marieljara656@gmail.com'],
7             fail_silently=False)
8     return render(request, 'send/index.html')
```

Archivo views.py

5.2 Prueba



Prueba del envío de email.

6 Ejercicio 3: Impresión de PDFs

6.1 Código

Para este ejercicio instalar la dependencia xhtml2pdf.

```
py -m pip install xhtml2pdf pip --upgrade
```


- **renderers.py:** Contiene la función `render_to_pdf` que utiliza `pisa` para convertir una plantilla HTML y un diccionario de contexto en un documento PDF. Devuelve una respuesta HTTP con el PDF generado.

```
1 from io import BytesIO
2 from django.http import HttpResponse
3 from django.template.loader import get_template
4
5 from xhtml2pdf import pisa
6
7 def render_to_pdf(template_src, context_dict={}):
8     template = get_template(template_src)
9     html = template.render(context_dict)
10    result = BytesIO()
11    pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")), result)
12    if pdf.err:
13        return HttpResponse("Invalid PDF", status_code=400, content_type='text/plain')
14    return HttpResponse(result.getvalue(), content_type='application/pdf')
```

Archivo `renderers.py`

- **views.py:** Contiene la vista que renderiza la plantilla `invoice.html` y la convierte en un PDF.

```
1 from django.http import HttpResponse
2 from django.views.generic import View
3 from django.template.loader import get_template
4 from .renderers import render_to_pdf
5
6 class GeneratePDF(View):
7     def get(self, request, *args, **kwargs):
8         template = get_template('pdf/invoice.html')
9         context = {
10             'invoice_number': 123,
11             'customer_name': 'John Cooper',
12             'amount': 1399.99,
13             'date': 'Today',
14         }
15         html = template.render(context)
16         pdf = render_to_pdf('pdf/invoice.html', context)
17         if pdf:
18             response = HttpResponse(pdf, content_type='application/pdf')
19             filename = "Invoice_%s.pdf" %("121212")
20             content = "inline; filename='%s'" %(filename)
21             response['Content-Disposition'] = content
22             download = request.GET.get("download")
23             if download:
24                 content = "attachment; filename='%s'" %(filename)
25             return response
26         return HttpResponse(pdf, content_type='application/pdf')
```

Archivo `views.py`

- **invoice.html:** Contiene la plantilla HTML que se convertirá en un PDF.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2 <html>
3     <head>
4         <title>First PDF</title>
```

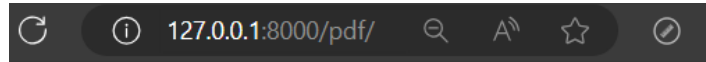
```

5      <style type="text/css">
6          body {
7              font-weight: 200;
8              font-size: 14px;
9          }
10         .header {
11             font-size: 20px;
12             font-weight: 100;
13             text-align: center;
14             color: #007cae;
15         }
16         .title {
17             font-size: 22px;
18             font-weight: 100;
19             /* text-align: right; */
20             padding: 10px 20px 0px 20px;
21         }
22         .title span {
23             color: #007cae;
24         }
25         .details {
26             padding: 10px 20px 0px 20px;
27             text-align: left !important;
28             /*margin-left: 40%;*/
29         }
30         .hrItem {
31             border: none;
32             height: 1px;
33             /* Set the hr color */
34             color: #333; /* old IE */
35             background-color: #fff; /* Modern Browsers */
36         }
37     </style>
38 </head>
39 <body>
40     <div class="wrapper">
41         <div class="header">
42             <p class="title">Invoice #{ invoice_number }</p>
43         </div>
44     </div>
45     <div class="details">
46         Bill to: { customer_name }<br />
47         Amount: { amount } <br />
48         Date: { date }
49         <hr class="hrItem" />
50     </div>
51 </body>
52 </html>

```

Archivo invoice.html

6.2 Prueba



Invoice #123

Bill to: John Cooper
Amount: 1399.99
Date: Today

Prueba de la impresión de PDFs.

7 Ejercicio Propuesto: Sistema de Gestión de Cursos

Se propone un sistema de gestión de cursos que permita a los estudiantes registrarse, registrar cursos junto con sus estudiantes, enviar mensajes por correo electrónico a los estudiantes y generar certificados en formato PDF para los estudiantes.

7.1 Código

7.1.1 Modelos

Define dos modelos de Django: `Student` y `Course`.

- `Student` tiene campos `name` y `email`.
- `Course` tiene campos `name`, `teacher`, y una relación many-to-many con `Student`.

```
1 from django.db import models
2
3 # Create your models here.
4 class Student(models.Model):
5     name = models.CharField(max_length=100)
6     email = models.EmailField(unique=True)
7
8     def __str__(self):
9         return self.name
10
11
12 class Course(models.Model):
13     name = models.CharField(max_length=100)
14     teacher = models.CharField(max_length=100)
15     students = models.ManyToManyField(Student)
16     def __str__(self):
17         return self.name
```

Archivo `models.py`

7.1.2 Formularios

- `StudentForm`: Formulario para registrar a un estudiante, basado en el modelo `Student` con campos `name` y `email`.

- **CourseForm**: Formulario para registrar un curso, basado en el modelo **Course** con campos **name**, **teacher** y **students** (usando **CheckboxSelectMultiple** para seleccionar varios estudiantes).
- **MessageForm**: Formulario para mandar un email a un estudiante **student** (**ModelChoiceField**), **course** (**ModelChoiceField**) y **message** (**CharField**).
- **CertifiedForm**: Formulario que debemos llenar para generar un certificado **student** (**ModelChoiceField**), **course** (**ModelChoiceField**) y **message** (**CharField**).

```
1 from django import forms
2 from .models import Student, Course
3
4 class StudentForm(forms.ModelForm):
5     class Meta:
6         model = Student
7         fields = ['name', 'email']
8         labels = {
9             'name': 'Nombre del Estudiante',
10            'email': 'Correo Electrónico'
11        }
12
13 class CourseForm(forms.ModelForm):
14     class Meta:
15         model = Course
16         fields = ['name', 'teacher', 'students']
17         widgets = {
18             'students': forms.CheckboxSelectMultiple
19         }
20         labels = {
21             'name': 'Nombre del Curso',
22             'teacher': 'Profesor',
23             'students': 'Estudiantes'
24         }
25
26 class MessageForm(forms.Form):
27     student = forms.ModelChoiceField(queryset=Student.objects.all(), label="Estudiante")
28     course = forms.ModelChoiceField(queryset=Course.objects.all(), label="Curso")
29     message = forms.CharField(widget=forms.Textarea, label="Mensaje")
30
31 class CertifiedForm(forms.Form):
32     student = forms.ModelChoiceField(queryset=Student.objects.all(), label="Estudiante")
33     course = forms.ModelChoiceField(queryset=Course.objects.all(), label="Curso")
34     message = forms.CharField(widget=forms.Textarea, label="Mensaje")
```

Archivo forms.py

7.1.3 Vistas

- **index(request)**: Renderiza la página principal.
- **register_student(request)**: Permite registrar nuevos estudiantes mediante un formulario.
- **register_course(request)**: Permite registrar nuevos cursos mediante un formulario.
- **list_students(request)**: Muestra todos los estudiantes registrados.

- `list_courses(request)`: Muestra todos los cursos registrados.
- `send_message(request)`: Envía un mensaje a un estudiante específico de un curso.
- `generate_certified(request)`: Genera un certificado en formato PDF para un estudiante de un curso específico, lo adjunta a un correo electrónico y lo envía al estudiante.

```
1 from django.shortcuts import render, redirect
2 from .models import Course, Student
3 from .forms import CourseForm, StudentForm, MessageForm, CertifiedForm
4 from django.core.mail import send_mail, EmailMessage
5
6 #Render PDF
7 from django.http import HttpResponse
8 from django.template.loader import get_template
9 from xhtml2pdf import pisa
10 import os
11
12
13 def index(request):
14     return render(request, 'index.html')
15
16 def register_student(request):
17     if request.method == 'POST':
18         form = StudentForm(request.POST)
19         if form.is_valid():
20             form.save()
21             return redirect('index')
22         else:
23             context = {'form': StudentForm()}
24             return render(request, 'register_student.html', context)
25     else:
26         context = {'form': StudentForm()}
27         return render(request, 'register_student.html', context)
28
29
30 def register_course(request):
31     if request.method == 'POST':
32         form = CourseForm(request.POST)
33         if form.is_valid():
34             form.save()
35             return redirect('index')
36         else:
37             context = {'form': CourseForm()}
38             return render(request, 'register_course.html', context)
39     else:
40         context = {'form': CourseForm()}
41         return render(request, 'register_course.html', context)
42
43
44 def list_students(request):
45     students = Student.objects.all()
46     context = {'students': students}
47     return render(request, 'list_students.html', context)
48
49 def list_courses(request):
```

```
50     courses = Course.objects.all()
51     context = {'courses': courses}
52     return render(request, 'list_courses.html', context)
53
54
55 emailRoot = 'mjarama@unsa.edu.pe'
56
57 def send_message(request):
58     if request.method == 'POST':
59         form = MessageForm(request.POST)
60         if form.is_valid():
61             student = form.cleaned_data['student']
62             course = form.cleaned_data['course']
63             message = form.cleaned_data['message']
64             subject = f"Mensaje para el curso {course.name}"
65             email = student.email
66             send_mail(subject, message, emailRoot, [email], fail_silently=False)
67             return redirect('index')
68         else:
69             context = {'form': MessageForm()}
70             return render(request, 'send_message.html', context)
71
72
73 def generate_certified(request):
74     if request.method == 'POST':
75         form = CertifiedForm(request.POST)
76         if form.is_valid():
77             student = form.cleaned_data['student']
78             course = form.cleaned_data['course']
79             message = form.cleaned_data['message']
80
81             # Render the PDF
82             template = get_template('certified.html')
83             context = {
84                 'student': student,
85                 'course': course,
86                 'message': message,
87                 'teacher': course.teacher,
88             }
89             html = template.render(context)
90
91             # Save PDF to a file
92             pdf_path = f"{student.name}_diploma.pdf"
93             with open(pdf_path, "w+b") as result_file:
94                 pisa_status = pisa.CreatePDF(html, dest=result_file)
95
96             if pisa_status.err:
97                 return HttpResponse("Error al generar el PDF")
98
99             # Send email with PDF attachment
100             subject = f"Certificado del curso {course.name}"
101             body = message
102             email = EmailMessage(
103                 subject=subject,
104                 body=body,
105                 from_email=emailRoot,
```

```
106         to=[student.email],
107     )
108     email.attach_file(pdf_path)
109     email.send()
110     os.remove(pdf_path)
111     return redirect('index')
112 else:
113     form = CertifiedForm()
114     return render(request, 'generate_certified.html', {'form': form})
```

Archivo views.py

7.2 Archivos Proyecto

7.2.1 Urls

- `index`: Página principal.
- `register_student`: Para registrar estudiantes.
- `register_course`: Para registrar cursos.
- `list_students`: Para listar todos los estudiantes.
- `list_courses`: Para listar todos los cursos.
- `send_message`: Para enviar mensajes a estudiantes de cursos específicos.
- `generate_certified`: Para generar certificados en formato PDF y enviarlos por correo electrónico.

```
1 from django.contrib import admin
2 from django.urls import path, include
3 from . import views
4
5 urlpatterns = [
6     path('', views.index, name='index'),
7     path('register_student/', views.register_student, name='register_student'),
8     path('register_course/', views.register_course, name='register_course'),
9     path('list_students/', views.list_students, name='list_students'),
10    path('list_courses/', views.list_courses, name='list_courses'),
11    path('send_message/', views.send_message, name='send_message'),
12    path('certified/', views.generate_certified, name='certified'),
13 ]
```

Archivo urls.py

7.2.2 Settings

- `INSTALLED_APPS`: Se agregan la app `systemM`.
- `EMAIL_HOST`, `EMAIL_PORT`, `EMAIL_HOST_USER`, `EMAIL_HOST_PASSWORD`: Configuración para enviar correos electrónicos.
- `STATICFILES_DIRS`: Se agrega la carpeta `static` para almacenar archivos estáticos.
- `EMAIL_USE_TLS`: Habilita el uso de TLS.

- EMAIL_USE_SSL: Habilita el uso de SSL.
- EMAIL_BACKEND: Configuración para enviar correos. electrónicos.

```
1  """
2  Django settings for system project.
3
4  Generated by 'django-admin startproject' using Django 5.0.6.
5
6  For more information on this file, see
7  https://docs.djangoproject.com/en/5.0/topics/settings/
8
9  For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/5.0/ref/settings/
11 """
12
13 from pathlib import Path
14 import json
15
16 # Build paths inside the project like this: BASE_DIR / 'subdir'.
17 BASE_DIR = Path(__file__).resolve().parent.parent
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/5.0/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'django-insecure-4x^c#o-i1_n1#m+bmkg#59k8yw%&h%qvm+=ch-pc2ot)o%7w7n'
24 STATIC_URL = '/static/'
25 EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
26 EMAIL_HOST = 'smtp.gmail.com'
27 EMAIL_PORT = 587
28 with open(BASE_DIR/'system'/ 'credentials.json') as f:
29     data = json.load(f)
30     EMAIL_HOST_USER = data['user']
31     EMAIL_HOST_PASSWORD = data['password']
32 EMAIL_USE_TLS = True
33 EMAIL_USE_SSL = False
34
35 # SECURITY WARNING: don't run with debug turned on in production!
36 DEBUG = True
37
38 ALLOWED_HOSTS = []
39
40 # Application definition
41
42 INSTALLED_APPS = [
43     'django.contrib.admin', 'django.contrib.auth',
44     'django.contrib.contenttypes', 'django.contrib.sessions',
45     'django.contrib.messages', 'django.contrib.staticfiles', 'systemM'
46 ]
47
48 MIDDLEWARE = [
49     'django.middleware.security.SecurityMiddleware',
50     'django.contrib.sessions.middleware.SessionMiddleware',
51     'django.middleware.common.CommonMiddleware',
52     'django.middleware.csrf.CsrfViewMiddleware',
```



```
53     'django.contrib.auth.middleware.AuthenticationMiddleware',
54     'django.contrib.messages.middleware.MessageMiddleware',
55     'django.middleware.clickjacking.XFrameOptionsMiddleware',
56 ]
57
58 ROOT_URLCONF = 'system.urls'
59
60 TEMPLATES = [
61     {
62         'BACKEND': 'django.template.backends.django.DjangoTemplates',
63         'DIRS': [],
64         'APP_DIRS': True,
65         'OPTIONS': {
66             'context_processors': [
67                 'django.template.context_processors.debug',
68                 'django.template.context_processors.request',
69                 'django.contrib.auth.context_processors.auth',
70                 'django.contrib.messages.context_processors.messages',
71             ],
72         },
73     },
74 ]
75
76 WSGI_APPLICATION = 'system.wsgi.application'
77
78 # Database
79 # https://docs.djangoproject.com/en/5.0/ref/settings/#databases
80
81 DATABASES = {
82     'default': {
83         'ENGINE': 'django.db.backends.sqlite3',
84         'NAME': BASE_DIR / 'db.sqlite3',
85     }
86 }
87
88 # Password validation
89 # https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-validators
90
91 AUTH_PASSWORD_VALIDATORS = [
92     {
93         'NAME':
94         'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
95     },
96     {
97         'NAME':
98         'django.contrib.auth.password_validation.MinimumLengthValidator',
99     },
100    {
101        'NAME':
102        'django.contrib.auth.password_validation.CommonPasswordValidator',
103    },
104    {
105        'NAME':
106        'django.contrib.auth.password_validation.NumericPasswordValidator',
107    },
108 ]
```

```
109
110 # Internationalization
111 # https://docs.djangoproject.com/en/5.0/topics/i18n/
112
113 LANGUAGE_CODE = 'en-us'
114
115 TIME_ZONE = 'UTC'
116
117 USE_I18N = True
118
119 USE_TZ = True
120
121 # Static files (CSS, JavaScript, Images)
122 # https://docs.djangoproject.com/en/5.0/howto/static-files/
123
124 STATIC_URL = 'static/'
125 STATICFILES_DIRS = [
126     BASE_DIR / "systemM" / "templates",
127 ]
128
129 # Default primary key field type
130 # https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field
131
132 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Archivo settings.py

7.3 Archivos Estáticos

7.3.1 HTML

Estas plantillas HTML definen la interfaz de usuario para diversas funcionalidades:

- **nav.html:** Plantilla base que define la estructura de navegación común a todas las páginas.

```
1 {% load static %}
2 <!DOCTYPE html>
3 <html lang="en">
4
5 <head>
6     <meta charset="UTF-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Django</title>
9     <link rel="stylesheet" href="{% static 'css/nav.css' %}">
10    <link rel="stylesheet" href="{% static 'css/form-page.css' %}">
11    <link rel="stylesheet" href="{% static 'css/lists.css' %}">
12 </head>
13
14 <body>
15
16     <nav>
17         <header>
18             <div class="logo">
19                 <div class="logo-1">CODER-DOJO</div>
20             </div>
21             <ul class="navlist">
22                 <li><a href="/">Home</a></li>
```

```
23         <li><a href="{% url 'register_student' %}">Registrar Studiante</a></li>
24         <li><a href="{% url 'register_course' %}">Registrar Curso</a></li>
25         <li><a href="{% url 'list_students' %}">Estudiantes</a></li>
26         <li><a href="{% url 'list_courses' %}">Lista de Cursos</a></li>
27         <li><a href="{% url 'send_message' %}">Enviar Mensaje</a></li>
28         <li><a href="{% url 'certified' %}">Generar Certificado</a></li>
29     </ul>
30 </header>
31 </nav>
32 {% block content %}
33 {% endblock %}
34 </body>
35
36 </html>
```

Archivo nav.html

- **index.html**: Plantilla para la página principal del sistema.

```
1 {% extends 'nav.html' %}
2 {% block content %}
3 <main>
4     <h1>Home</h1>
5     <p>Bienvenido al sistema de administración de cursos.</p>
6 </main>
7 {% endblock %}
```

Archivo index.html

- **register_student.html**: Plantilla para registrar un nuevo estudiante.

```
1 {% extends 'nav.html' %}
2 {% block content %}
3 <main>
4     <h1>Registrar un nuevo Estudiante</h1>
5     <form action="{% url 'register_student' %}" method="POST">
6         {% csrf_token %}
7         {{ form.as_p }}
8         <button type="submit">Registrar Estudiante</button>
9     </form>
10 </main>
11 {% endblock %}
```

Archivo register_student.html

- **register_course.html**: Plantilla para registrar un nuevo curso.

```
1 {% extends 'nav.html' %}
2 {% block content %}
3 <main>
4     <h1>Crear un Curso</h1>
5     <form method="POST">
6         {% csrf_token %}
7         {{ form.as_p }}
8         <button type="submit">Crear Curso</button>
9     </form>
10 </main>
11 {% endblock %}
```

Archivo register_course.html

- **list_courses.html**: Plantilla para mostrar la lista de cursos registrados.

```
1  {% extends 'nav.html' %}
2  {% block content %}
3  <main>
4    <h1>Lista de Cursos</h1>
5    <table>
6      <tr>
7        <th>Nombre del Curso</th>
8        <th>Profesor</th>
9        <th>Estudiantes</th>
10     </tr>
11     {% for course in courses %}
12     <tr>
13       <td>{{ course.name }}</td>
14       <td>{{ course.teacher }}</td>
15       <td>
16         <ul>
17           {% for student in course.students.all %}
18           <li>{{ student.name }} - {{ student.email }}</li>
19           {% endfor %}
20         </ul>
21       </td>
22     </tr>
23     {% endfor %}
24   </table>
25 </main>
26 {% endblock %}
```

Archivo list_courses.html

- **list_students.html**: Plantilla para mostrar la lista de estudiantes registrados.

```
1  {% extends 'nav.html' %}
2  {% block content %}
3  <main>
4    <h1>Lista de Estudiantes</h1>
5    <table>
6      <tr>
7        <th>Nombre</th>
8        <th>Email</th>
9      </tr>
10     {% for student in students %}
11     <tr>
12       <td>{{ student.name }}</td>
13       <td>{{ student.email }}</td>
14     </tr>
15     {% endfor %}
16   </table>
17 </main>
18 {% endblock %}
```

Archivo list_students.html

- **send_message.html**: Plantilla para enviar un mensaje a los estudiantes.

```
1  {% extends 'nav.html' %}
2  {% block content %}
3  <main>
4    <h1>Enviar Mensaje a Estudiante</h1>
5    <form method="post">
6      {% csrf_token %}
7      {{ form.as_p }}
8      <button type="submit">Enviar</button>
9    </form>
10 </main>
11 {% endblock %}
```

Archivo send_message.html

- **generate_certified.html:** Plantilla que muestra el formulario para generar certificados.

```
1  {% extends 'nav.html' %}
2  {% block content %}
3  <main>
4    <h1>Generar Certificado para Estudiante</h1>
5    <form method="post">
6      {% csrf_token %}
7      {{ form.as_p }}
8      <button type="submit">Generar Certificado</button>
9    </form>
10 </main>
11 {% endblock %}
```

Archivo generate_certified.html

- **certified.html:** Plantilla que muestra el certificado generado.

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <title>Diploma</title>
7    <style>
8      body {
9        font-family: Arial, sans-serif;
10       margin: 0;
11       padding: 0;
12       background: #f4f4f4;
13       text-align: center;
14     }
15
16     .container {
17       display: flex;
18       justify-content: center;
19       align-items: center;
20       height: 100vh;
21       background: #ffffff;
22       padding: 50px;
23       border: 1px solid #000;
24     }
25
26     .diploma {
```

```
27         width: 90%;
28         height: 90%;
29         border: 1px solid #000;
30         display: flex;
31         flex-direction: row;
32         padding: 20px;
33         background: #ffffff;
34     }
35
36     .diploma-left {
37         background-color: #003366;
38         color: #ffffff;
39         flex: 1;
40         padding: 30px;
41         display: flex;
42         flex-direction: column;
43         justify-content: center;
44         align-items: center;
45     }
46     .diploma-left p{
47         color:#003366;
48         background-color: transparent;
49     }
50
51     .diploma-right {
52         flex: 2;
53         padding: 20px;
54         display: flex;
55         flex-direction: column;
56         justify-content: center;
57         align-items: center;
58     }
59
60     .diploma-right h1,
61     .diploma-right h2,
62     .diploma-right h3,
63     .diploma-right p {
64         text-align: center;
65         margin: 10px 0;
66     }
67
68     .diploma-right .signatures {
69         display: flex;
70         justify-content: space-around;
71         width: 100%;
72         margin-top: 20px;
73     }
74
75     .diploma-right .signatures div {
76         text-align: center;
77         margin-left: 10px;
78     }
79 </style>
80 </head>
81
82 <body>
```

```

83     <div class="container">
84         <div class="diploma">
85             <div class="diploma-left">
86                 <h1>Certificado de Logro</h1>
87                 <p>Porque la educación es el camino al éxito
88                     y el conocimiento es la llave para abrir
89                     las puertas del futuro.
90                 </p>
91             </div>
92             <div class="diploma-right">
93                 <h2>{{ student.name }}</h2>
94                 <p>por haber completado exitosamente el curso</p>
95                 <h3>{{ course.name }}</h3>
96                 <p>"{{ message }}"</p>
97                 <div class="signatures">
98                     <div>
99                         <p>Firma 1</p>
100                        <p>_____</p>
101                    </div>
102                    <div>
103                        <p>Firma 2</p>
104                        <p>_____</p>
105                    </div>
106                </div>
107                <p>Junio, 2024</p>
108            </div>
109        </div>
110    </div>
111 </body>
112
113 </html>

```

Archivo certified.html

7.3.2 CSS

- **nav.css:** Estilo para la barra de navegación.

```

1  * {
2      margin: 0;
3      padding: 0;
4      box-sizing: border-box;
5  }
6
7  nav {
8      background-color: #333;
9      color: white;
10     padding: 10px 20px;
11 }
12
13 nav header {
14     display: flex;
15     justify-content: space-between;
16     align-items: center;
17 }
18
19 .logo {

```

```
20     font-size: 1.5rem;
21     font-weight: bold;
22 }
23
24 .navlist {
25     list-style-type: none;
26     display: flex;
27 }
28
29 .navlist li {
30     margin-right: 20px;
31     font-size: 1.2rem;
32 }
33
34 .navlist li a {
35     color: white;
36     text-decoration: none;
37     padding: 10px;
38 }
39
40 .navlist li a:hover {
41     background-color: #555;
42 }
43
44 .logo-1 {
45     font-size: 1.5rem;
46     font-weight: bold;
47 }
```

Archivo nav.css

- **form-page.css:** Estilo para el formulario de registro de estudiantes y cursos.

```
1  body {
2      font-family: Arial, sans-serif;
3      line-height: 1.6;
4      background-color: #f0f0f0;
5      margin: 0;
6      padding: 0;
7  }
8
9  main {
10     max-width: 800px;
11     margin: 40px auto;
12     background-color: #fff;
13     padding: 40px 20px;
14     border-radius: 10px;
15     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
16 }
17
18 h1 {
19     font-size: 2.5rem;
20     margin-bottom: 20px;
21     text-align: center;
22 }
23
24 form {
```



```
25     max-width: 600px;
26     margin: 0 auto;
27 }
28
29 form p {
30     margin-bottom: 10px;
31 }
32
33 form label {
34     display: block;
35     margin-bottom: 8px;
36     font-weight: bold;
37 }
38
39 form input[type="text"],
40 form input[type="email"],
41 form select {
42     width: calc(100% - 20px);
43     padding: 10px;
44     font-size: 1rem;
45     margin-bottom: 15px;
46     border: 1px solid #ccc;
47     border-radius: 4px;
48 }
49
50 form textarea {
51     width: calc(100% - 20px);
52     padding: 10px;
53     font-size: 1rem;
54     margin-bottom: 15px;
55     border: 1px solid #ccc;
56     border-radius: 4px;
57 }
58
59 form button {
60     background-color: #444;
61     color: #fff;
62     padding: 12px 20px;
63     font-size: 1.2rem;
64     border: none;
65     border-radius: 4px;
66     cursor: pointer;
67 }
68
69 form button:hover {
70     background-color: #777;
71 }
72
73 .errorlist {
74     color: red;
75     margin-bottom: 15px;
76 }
77
78 .errorlist li {
79     list-style-type: none;
80 }
```

```
81  main > p {
82      text-align: center;
83      margin-top: 20px;
84      font-size: 1.2rem;
85  }
86
```

Archivo form-page.css

- **lists.css:** Estilo para las listas de estudiantes y cursos.

```
1  table {
2      width: 100%;
3      border-collapse: collapse;
4      margin-top: 20px;
5  }
6
7  table th, table td {
8      border: 1px solid #ddd;
9      padding: 8px;
10     text-align: left;
11 }
12
13 table th {
14     background-color: #f2f2f2;
15     color: #333;
16 }
17
18 h2 {
19     font-size: 1.8rem;
20     margin-top: 20px;
21 }
22
23 p {
24     margin-bottom: 10px;
25 }
26
27 ul {
28     margin-bottom: 20px;
29 }
30
31 ul li {
32     margin-bottom: 5px;
33 }
34
35 table {
36     width: 100%;
37     border-collapse: collapse;
38     margin-top: 20px;
39 }
40
41 table th, table td {
42     border: 1px solid #ddd;
43     padding: 8px;
44     text-align: left;
45 }
46
```

```

47 table th {
48     background-color: #f2f2f2;
49     color: #333;
50 }
51
52 ul {
53     list-style-type: none;
54     padding: 0;
55     margin: 0;
56 }
57
58 ul li {
59     margin-bottom: 5px;
60 }

```

Archivo lists.css

7.4 Prueba

7.4.1 Registro de Estudiantes

Nombre del Estudiante:

Raul Jara

Correo Electrónico:

jarar727@gmail.com

Registrar Estudiante

Registro de estudiantes.

Table: systemM_student

	id	name	email
	Filter	Filter	Filter
1	1	Alisson Jara	mjarama@unsa.edu.pe
2	2	Chirthian Mestas	cmeztas@unsa.edu.pe
3	3	Yenaro Joel	ynoa@unsa.edu.pe
4	4	Mariel Jara	marieljara656@gmail.com
5	5	Raul Jara	jarar727@gmail.com

Registro de estudiantes en la base de datos.

7.4.2 Registro de Cursos

Crear un Curso

Nombre del Curso:

EDA

Profesor:

Karim Guevara

Estudiantes:

- ☒ Alisson Jara
- ☒ Chirthian Mestas
- ☐ Yenaro Joel
- ☐ Mariel Jara
- ☒ Raul Jara

Crear Curso

Registro de cursos.

Table: systemM_course

	id	name	teacher
	Filter	Filter	Filter
1	1	PW2	Lino Pinto
2	2	EDA	Karim Guevara

Registro de cursos en la base de datos.

Table: systemM_course_students

	id	course_id	student_id
	Filter	Filter	Filter
	1	1	1
	2	1	2
	3	2	1
	4	2	2
	5	2	5

Relación entre estudiantes y cursos.

7.4.3 Lista de Estudiantes

Lista de Estudiantes

Nombre	Email
Alisson Jara	mjarama@unsa.edu.pe
Chirthian Mestas	cmeztas@unsa.edu.pe
Yenaro Joel	ynoa@unsa.edu.pe
Mariel Jara	mariejara656@gmail.com
Raul Jara	jarar727@gmail.com

Lista de estudiantes.

7.4.4 Lista de Cursos

Lista de Cursos

Nombre del Curso	Profesor	Estudiantes
PW2	Lino Pinto	Alisson Jara - mjarama@unsa.edu.pe Chirthian Mestas - cmeztas@unsa.edu.pe
EDA	Karim Guevara	Alisson Jara - mjarama@unsa.edu.pe Chirthian Mestas - cmeztas@unsa.edu.pe Raul Jara - jarar727@gmail.com

Lista de cursos.

7.4.5 Envío de Mensajes

Mensaje para el curso PW2 Recibidos x



mjarama@unsa.edu.pe

Esta semana avanzaremos el framework Angular



mjarama@unsa.edu.pe

Recuerda mandar el informe de PW2 de Lab08

Envío de mensajes.

7.4.6 Generación de Certificados



mjararama@unsa.edu.pe

para mí ▼

Felicidades por finalizar el curso de PW2:)

1 archivo adjunto • Analizado por Gmail ⓘ



Generación de certificados.

8 URL del repositorio en GitHub

- <https://github.com/Alsnj20/pw2-24a/tree/main/lab08>

9 Estructura de laboratorio 08

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab08/
|--Course_Management_System/
|  |--systemM/
|    |--migrations/
|    |--templates/
|      |--css/
|        |--form-page.css
|        |--lists.css
|        |--nav.css
|      |--pdf/
|        |--certified.html
|        |--generate_certified.html
|        |--index.html
|        |--invoice.html
|        |--list_courses.html
|        |--list_students.html
|        |--nav.html
|        |--register_course.html
|        |--register_student.html
|        |--send_message.html
|  |--__init__.py
|  |--admin.py
|  |--apps.py
|  |--forms.py
|  |--models.py
|  |--renderers.py
|  |--tests.py
|  |--urls.py
|  |--views.py
|  |--system/
|--relations_examples/
|--email_example/
|--print_pdfs/
|--Latex/
  |--linopinto_pw2_24a_lab08.tex
  |--linopinto_pw2_24a_lab08.pdf
  |--img/
    |--commits.png
    |--r1operaciones.png
    |--r2operaciones.png
    |--r1tabla1.png
    |--r1tabla2.png
    |--r2tabla1.png
    |--r2tabla2.png
    |--r2tabla3.png
    |--send_email.png
    |--print_pdf.png
    |--registerStudent.png
    |--registerSBD.png
    |--registerCourse.png
    |--registerCBD.png
    |--relation.png
```

```
--listStudents.png
--listCourses.png
--sendMessage.png
--certificado.png
|.gitignore
```

10 Rúbrica

Tabla: Rúbrica para contenido del Informe y evidencias

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o observación)	4	×	4	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación)	4	×	4	
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4	×	4	
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación)	2	×	2	
7.Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2	×	1	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación)	4	×	4	
Total		20	Completo	19	

11 Referencias

- <https://github.com/>
- <https://git-scm.com/>
- <https://www.w3schools.com/python/>