

Informe de Laboratorio 06

Tema: Django 1

Nota

Estudiante	Escuela	Asignatura
Milene Pacheco Esquinarila Isasi Vargas Angie	Escuela Profesional de Ingeniería de Sistemas	Pweb2 Semestre: III

Laboratorio	Tema	Duración
06	Django 1	horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A		

1. Ejercicios Propuestos

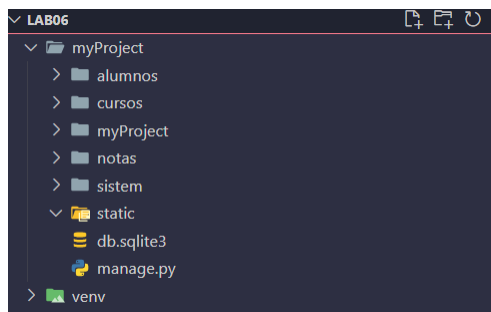
- Implementa un Sistema en Django que maneje una tabla de Alumnos, una de Cursos y una de NotasAlumnosPorCurso y que permita ingresar a nuevos alumnos, nuevos cursos y finalmente permita ingresar las notas por curso. Laboratorio realizado en grupos de 2 estudiantes. Crear un unico proyecto y compartir github con el profesor (CarloCorrales010). No olvidar enviar video a flipgrip de manera personal.

2. URL de Repositorio Github

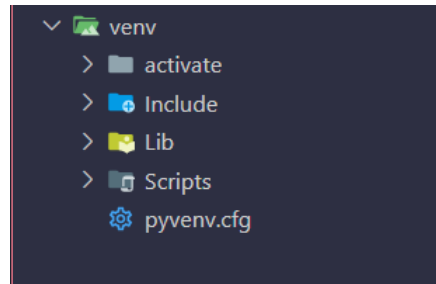
- <https://github.com/Milene-pe/LAB-PWEB2/tree/main/LAB06>

3. Código:

Directorio:



Creamos nuestro entorno virtual:



Esta tarea indica crear 3 aplicaciones, las cuales nombré como: alumnos, cursos y notas, para lo cual las agregue al settings.py de mi proyecto

```

1  # Application definition
2
3  INSTALLED_APPS = [
4      'django.contrib.admin',
5      'django.contrib.auth',
6      'django.contrib.contenttypes',
7      'django.contrib.sessions',
8      'django.contrib.messages',
9      'django.contrib.staticfiles',
10     'sistem',
11     'alumnos',
12     'cursos',
13     'notas'
14 ]
15

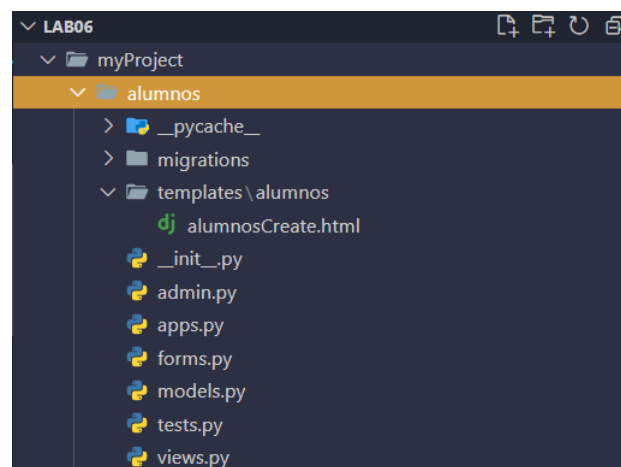
```

Aplicación Alumnos

Se crean por defectos los archivos admin.py, apps.py, models.py, tests.py y view.py.

Pero se creo el forms.py, se mostrará los cambios que se hizo en los archivos

Los datos que se solicitará serán nombres, apellidos y CUI



admin.py:

```
from django.contrib import admin
from .models import Alumno

admin.site.register(Alumno)
```

apps.py:

```
from django.apps import AppConfig

class AlumnoConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'alumnos'
```

models.py:

```
from django.db import models

class Alumno(models.Model):
    nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    cui = models.IntegerField()
```

views.py:

```
from django.shortcuts import render
from .models import Alumno
from .forms import AlumnoForm

# Create your views here.
def alumnoTestView(request):

    obj = Alumno.objects.get(id = 1)
    context = {
        'objecto':obj,
    }
    return render(request, 'alumnos/descripcion.html', context)
def alumnoCreateView(request):
    form = AlumnoForm(request.POST or None)
    if form.is_valid():
        form.save()
        form = AlumnoForm()

    return render(request, 'alumnos/alumnosCreate.html', {'form': form})
```

Se creo el forms.py ya que la tarea solicita que mediante un formulario se ingrese los datos y se guarden en el panel de administracion de Django:

```
from django import forms

from .models import Alumno

class AlumnoForm(forms.ModelForm):
    class Meta:
        model = Alumno
        fields = [
            'nombre',
            'apellido',
            'cui',
        ]
```

Se creo alumnosCreate.html que conecta el formulario para poder ingresar los datos

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Crear Alumno</title>
</head>
<body>
    <header>
        <h1>Crear Alumno</h1>
    </header>
    <main>

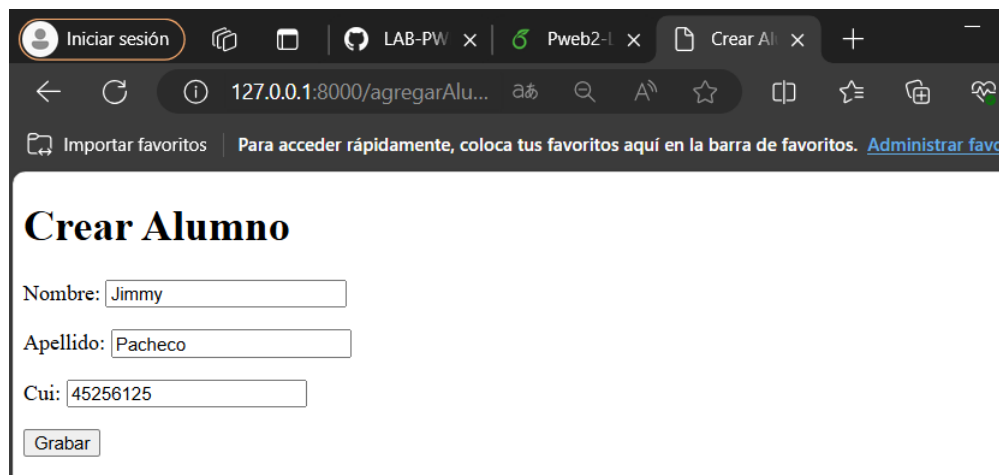
        {% block content %}
        <form method="POST">
            {% csrf_token %}
            {{ form.as_p }}
            <input type="submit" value="Grabar"/>
        </form>
        {% endblock %}
    </main>
</body>
</html>
```

Iniciaremos el servidor

```
(venv) C:\UNIVERSIDAD\GIT\LAB-PWEB2\LAB06\myProject>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 02, 2024 - 18:58:53
Django version 5.0.6, using settings 'myProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Vamos al <http://127.0.0.1:8000/agregarAlumno/> y llenamos el formulario y verificamos si se guardo en el panel de Django



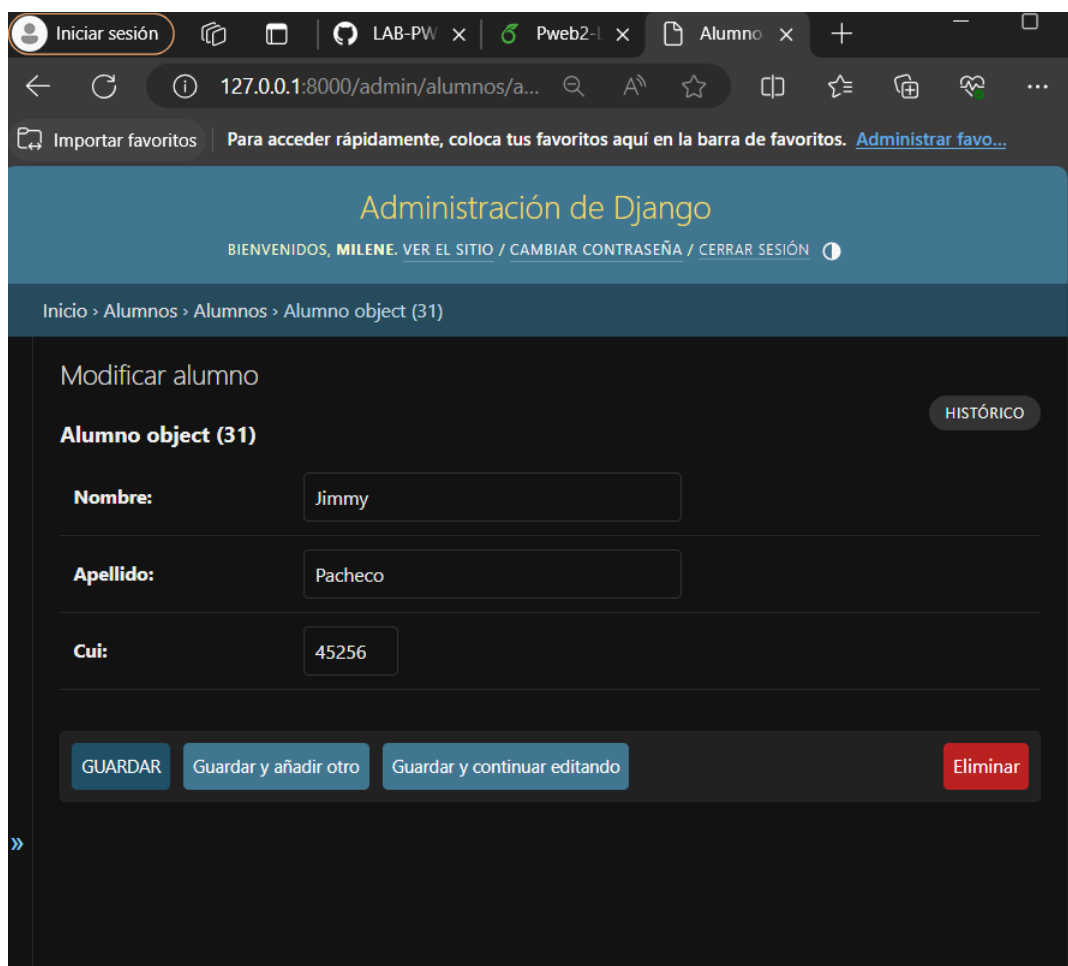
Crear Alumno

Nombre:

Apellido:

Cui:

Verificamos en el panel de Django



Administración de Django

BIENVENIDOS, **MILENE**. [VER EL SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [CERRAR SESIÓN](#)

Inicio > Alumnos > Alumnos > Alumno object (31)

Modificar alumno HISTÓRICO

Alumno object (31)

Nombre:

Apellido:

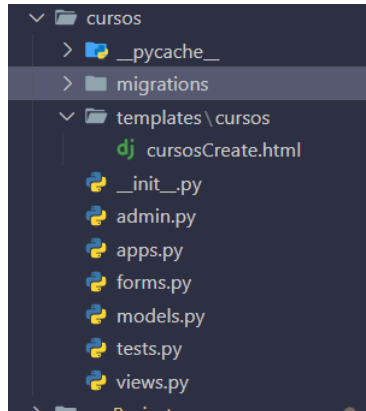
Cui:

Aplicación Cursos

Se crean por defectos los archivos admin.py, apps.py, models.py, tests.py y view.py.

Pero se creo el forms.py, se mostrará los cambios que se hizo en los archivos

Los datos que se solicitará serán nombre, docente y codigo



admin.py:

```
from django.contrib import admin
from .models import Curso

admin.site.register(Curso)
```

apps.py:

```
from django.apps import AppConfig

class CursosConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'cursos'
```

models.py:

```
from django.db import models

class Curso(models.Model):
    nombre = models.CharField(max_length=100)
    docente = models.CharField(max_length=100)
    codigo = models.IntegerField()
```

views.py:

```
from django.shortcuts import render
from .models import Curso
from .forms import CursoForm

# Create your views here.
def cursoTestView(request):
```

```
obj = Curso.objects.get(id = 1)
context = {
    'objecto':obj,
}
return render(request, 'cursos/descripcion.html', context)
def cursoCreateView(request):
    form = CursoForm(request.POST or None)
    if form.is_valid():
        form.save()
        form = CursoForm()

    return render(request, 'cursos/cursosCreate.html', {'form': form})
```

Se creo el forms.py ya que la tarea solicita que mediante un formulario se ingrese los datos y se guarden en el panel de administracion de Django:

```
from django import forms

from .models import Curso

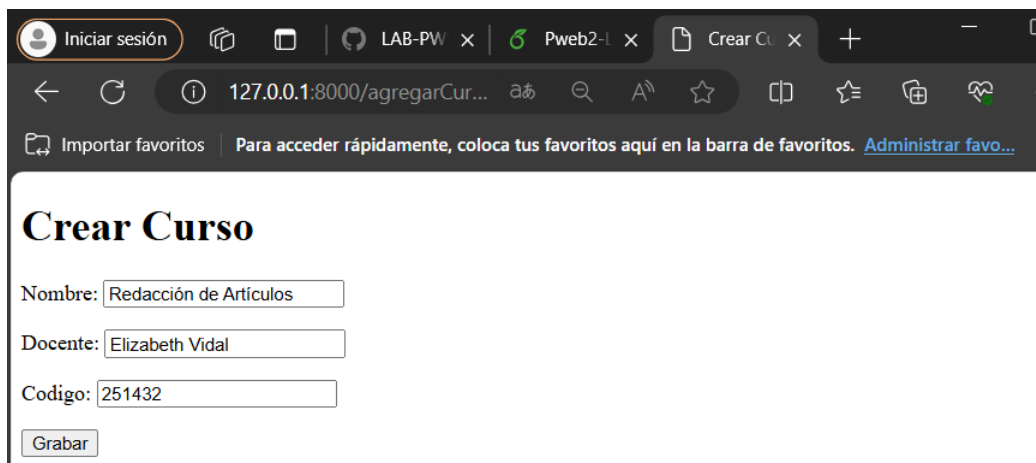
class CursoForm(forms.ModelForm):
    class Meta:
        model = Curso
        fields = [
            'nombre',
            'docente',
            'codigo',
        ]
```

Se creo cursosCreate.html que conecta el formulario para poder ingresar los datos

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Crear Curso</title>
</head>
<body>
    <header>
        <h1>Crear Curso</h1>
    </header>
    <main>

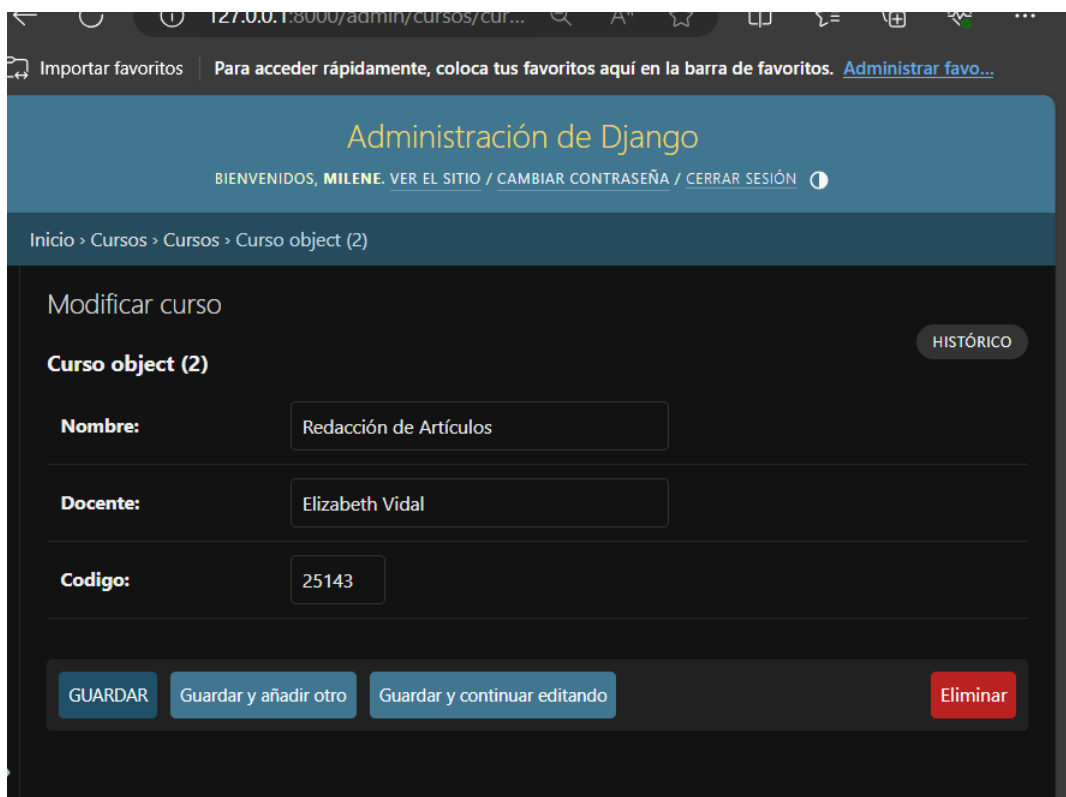
        {% block content %}
        <form method="POST">
            {% csrf_token %}
            {{ form.as_p }}
            <input type="submit" value="Grabar"/>
        </form>
        {% endblock %}
    </main>
</body>
</html>
```

Iniciaremos el servidor, e iremos al `http://127.0.0.1:8000/agregarCurso/` y llenamos el formulario y verificamos si se guardó en el panel de Django



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/agregarCur...`. The page title is "Crear Curso". The form contains three input fields: "Nombre:" with the value "Redacción de Artículos", "Docente:" with the value "Elizabeth Vidal", and "Codigo:" with the value "251432". There is a "Grabar" button at the bottom of the form.

Verificamos en el panel de Django



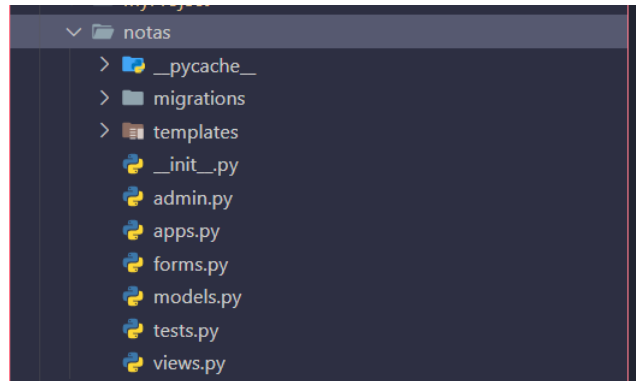
The screenshot shows the Django Admin interface. The header says "Administración de Django" and "BIENVENIDOS, MILENE. VER EL SITIO / CAMBIAR CONTRASEÑA / CERRAR SESIÓN". The breadcrumb trail is "Inicio > Cursos > Cursos > Curso object (2)". The main heading is "Modificar curso". Below it, the form "Curso object (2)" is displayed with the same data as the previous form: "Nombre: Redacción de Artículos", "Docente: Elizabeth Vidal", and "Codigo: 25143". At the bottom, there are four buttons: "GUARDAR", "Guardar y añadir otro", "Guardar y continuar editando", and "Eliminar". A "HISTÓRICO" button is also visible in the top right corner of the form area.

Aplicación Notas

Se crean por defectos los archivos admin.py, apps.py, models.py, tests.py y view.py.

Pero se creo el forms.py, se mostrará los cambios que se hizo en los archivos

Los datos que se solicitará serán alumno, curso y nota



admin.py:

```
from django.contrib import admin
from .models import Nota

admin.site.register(Nota)
```

apps.py:

```
from django.apps import AppConfig

class CursosConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'notas'
```

models.py:

```
from django.db import models

class Nota(models.Model):
    alumno = models.CharField(max_length=100)
    curso = models.CharField(max_length=100)
    nota = models.IntegerField()
```

views.py:

```
from django.shortcuts import render
from .models import Nota
from .forms import NotaForm

# Create your views here.
def notaTestView(request):

    obj = Nota.objects.get(id = 1)
```

```
context = {
    'objecto':obj,
}
return render(request, 'notas/descripcion.html', context)
def notaCreateView(request):
    form = NotaForm(request.POST or None)
    if form.is_valid():
        form.save()
        form = NotaForm()

    return render(request, 'notas/notasCreate.html', {'form': form})
```

Se creo el forms.py ya que la tarea solicita que mediante un formulario se ingrese los datos y se guarden en el panel de administracion de Django:

```
from django import forms

from .models import Nota

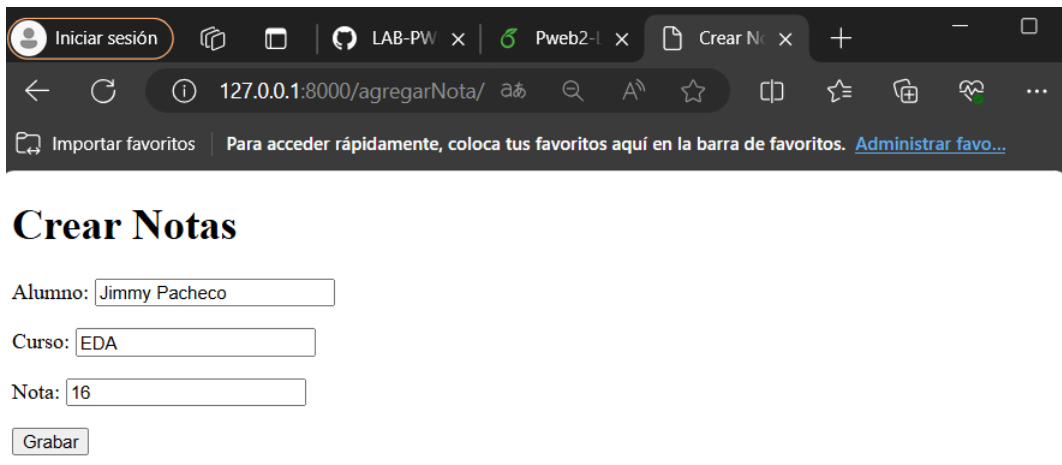
class NotaForm(forms.ModelForm):
    class Meta:
        model = Nota
        fields = [
            'alumno',
            'curso',
            'nota',
        ]
```

Se creo notasCreate.html que conecta el formulario para poder ingresar los datos

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Crear Notas</title>
</head>
<body>
    <header>
        <h1>Crear Notas</h1>
    </header>
    <main>

        {% block content %}
        <form method="POST">
            {% csrf_token %}
            {{ form.as_p }}
            <input type="submit" value="Grabar"/>
        </form>
        {% endblock %}
    </main>
</body>
</html>
```

Iniciaremos el servidor, e iremos al `http://127.0.0.1:8000/agregarNota/` y llenamos el formulario y verificamos si se guardó en el panel de Django



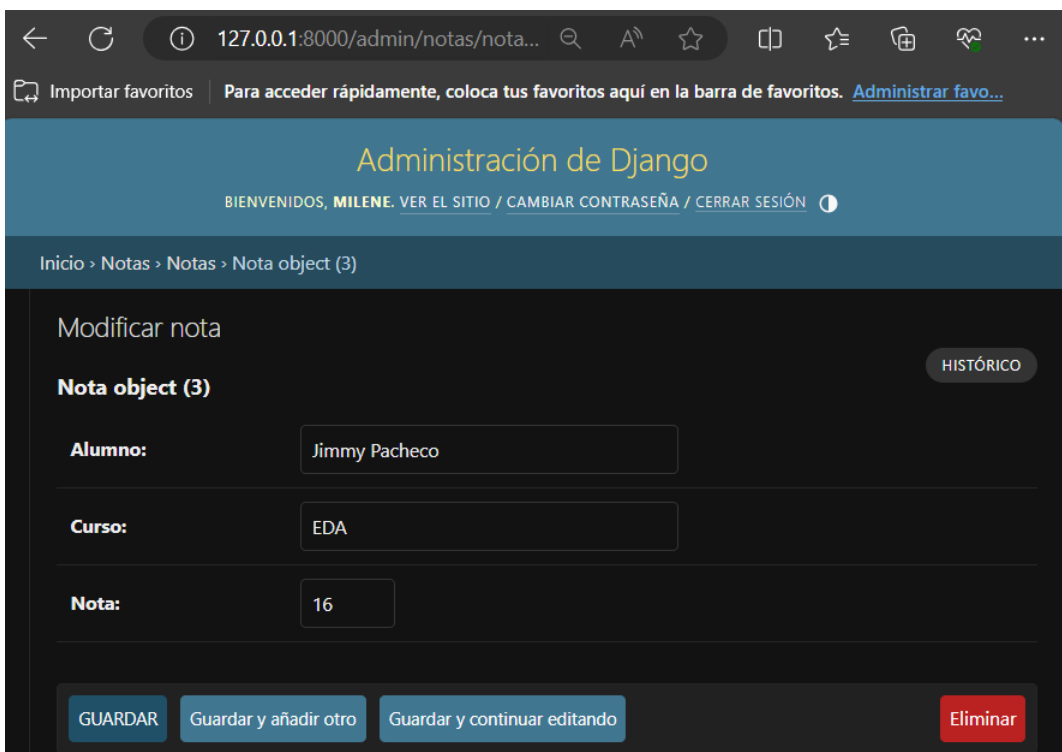
Crear Notas

Alumno:

Curso:

Nota:

Verificamos en el panel de Django



Administración de Django

BIENVENIDOS, MILENE. [VER EL SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [CERRAR SESIÓN](#)

Inicio > Notas > Notas > Nota object (3)

Modificar nota

Nota object (3) HISTÓRICO

Alumno:

Curso:

Nota: