

Informe Testing

Presentado por:

Pablo Andrés Nino Barreto- pninob@unal.edu.co

Jeisson Duván Bareño Ruiz- jbareno@unal.edu.co

Víctor Camilo Cañón Castellanos - vcanonc@unal.edu.co

Jacobo Álzate Corredor- jalzateco@unal.edu.co

Profesor:

Oscar Eduardo Alvarez Rodriguez- oalvarezr@unal.edu.co



Universidad Nacional de Colombia

Facultad de Ingeniería

**Departamento de Ingeniería de Sistemas e Industrial
2025**

Informe de Pruebas

Introducción breve

La aplicación es un **Sistema de pruebas en línea** (SISEMAX) desarrollado con Django en el backend. Se encarga de gestionar asignaturas, grados y grupos académicos, permitiendo la administración de cursos y usuarios en un entorno escolar.

Resumen de los tests

Nombre del integrante	Tipo de prueba	Componente probado	Herramienta usada
Pablo Niño	Prueba unitaria	Modelo Asignatura	Django TestCase
Victor Cañon	Prueba unitaria	Modelo Grado	Django TestCase
Jacobo Alzate	Prueba unitaria	Modelo Grupo	Django TestCase
Jeisson Bareño	Prueba unitaria	Método __str__ de Asignatura	Django TestCase

```

tests.py
academico > tests.py > ...
1  from django.test import TestCase
2  from academico.models import Asignatura, Grado, Grupo
3
4
5  #Test de Pablo Niño
6  class AsignaturaModelTest(TestCase):
7      def test_crear_asignatura(self):
8          asignatura = Asignatura.objects.create(nombre="Matemáticas")
9          asignatura_recuperada = Asignatura.objects.get(id_asignatura=asignatura.id_asignatura)
10         self.assertEqual(asignatura_recuperada.nombre, "Matemáticas")
11
12
13  #test de Victor Cañon
14  class GradoModelTest(TestCase):
15      def test_crear_grado(self):
16          grado = Grado.objects.create(nombre="Primero")
17          grado_recuperado = Grado.objects.get(id_grado=grado.id_grado)
18          self.assertEqual(grado_recuperado.nombre, "Primero")
19
20
21  #Test de Jacobo Alzate
22  class GrupoModelTest(TestCase):
23      def test_crear_grupo(self):
24          grado = Grado.objects.create(nombre="Primero")
25          grupo = Grupo.objects.create(nombre="Grupo A", id_grado=grado)
26          grupo_recuperado = Grupo.objects.get(id_grupo=grupo.id_grupo)
27          self.assertEqual(grupo_recuperado.nombre, "Grupo A")
28          self.assertEqual(grupo_recuperado.id_grado, grado)
29
30
31  #Test de Jeisson Bareño
32  class AsignaturaStrTest(TestCase):
33      def test_str_asignatura(self):
34          asignatura = Asignatura.objects.create(nombre="Física")
35          self.assertEqual(str(asignatura), "Física")
36

```

Resultado de los test

Imagen de la ejecución en la terminal

```
35 self.assertEqual(str(asignatura), "Física")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
● PS C:\Users\barre\Documents\ing proyecto> python manage.py test academico
Found 4 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
....
-----
Ran 4 tests in 0.004s

OK
Destroying test database for alias 'default'...
```

Lecciones aprendidas y Dificultades encontradas

- Inicialmente, el test del modelo Grupo falló debido a la ausencia de un Grado requerido en la clave foránea. Fue necesario corregir el test creando primero un Grado antes de asignarlo a un Grupo.
- Las pruebas unitarias ayudan a identificar errores en la lógica de negocio desde etapas tempranas.