

Evaluación Sumativa I:
Taller: "Aplicación web con Django Admin" (25%)

ÁREA ACADÉMICA	Informática y Telecomunicaciones	CARRERA	Ingeniería en Informática
ASIGNATURA	Programación Back End	CÓDIGO	TI2041
SEDE	La Serena	DOCENTE	Mariavictoria Freites
Unidad de Aprendizaje	N° II	Criterios a Evaluar	(2.1.1, 2.1.2)
DURACIÓN		FECHA	

NOMBRE ESTUDIANTE:

Apellido Paterno

Apellido Materno

Nombres

RUT: -

PUNTAJE MÁXIMO

110 pts.

NOTA:

PUNTAJE OBTENIDO

Firma conforme

Solicita re-corrección

Sí

No

Motivo:

INSTRUCCIONES GENERALES:

1. La nota 4.0 se obtiene logrando un 60% del puntaje total.
2. Preocúpese de la redacción, ortografía y legibilidad de sus respuestas.
3. Está prohibido el préstamo (o solicitud) de materiales durante la evaluación.
4. Si se encuentra una similitud superior o igual al 20% con respecto a otra entrega. Ambas serán calificadas con nota 1.
5. Si se sospecha el uso de herramientas de inteligencia artificial (IA) para el desarrollo de la evaluación, esta será calificada con nota 1.
6. Se prohíbe el uso de celulares, mp3, mp4, iphone, ipod o similares durante la evaluación. (Según corresponda indicar: Se prohíbe el uso de calculadoras).
7. La fecha de entrega será **el martes 07 de octubre de 2025, hasta las 18:30 horas a través de la plataforma**. No se recibirán entregas en computadores personales, ni a través de correo electrónico. Tampoco se recibirán entregas posteriores a la fecha límite.

Aprendizaje esperado

2.1 Codifica aplicaciones web para dar soluciones tentativas a una problemática utilizando un framework del lado del servidor.

Criterios de evaluación

- 2.1.1 Configura la conexión a una base de datos, según requerimiento.
- 2.1.2 Utiliza el administrador de Django, de acuerdo a los requerimientos.

Requerimientos:

Cada integrante del equipo debe desarrollar un mantenedor en Django, contenido en una aplicación propia dentro del proyecto.

- Si el grupo tiene 3 integrantes, deberán existir 3 aplicaciones distintas, cada una con su propio mantenedor.
- Cada aplicación debe contener al menos 2 modelos relacionados mediante ForeignKey.
- Los modelos deben reflejar la realidad de la empresa asignada al grupo.
- Se deben realizar migraciones y crear la base de datos real (MySQL o PostgreSQL, entre otras).
- Los modelos deben estar registrados en el administrador de Django.
- Se deben ingresar registros de prueba (mínimo 10) en cada modelo desde el administrador.
- Personalizar el administrador para mostrar columnas relevantes y facilitar la gestión.

Ejemplo docente:

Modelo principal: Doctor

Modelo secundario: Especialidad (relación con Doctor)

Modelo secundario: Área (relación con Doctor).

Instrucciones:

Por lo anterior, debe:

- Configurar la conexión a la base de datos en el archivo settings.py.
- Definir los modelos en cada aplicación con sus respectivas relaciones.
- Ejecutar los comandos makemigrations y migrate.
- Registrar los modelos en el administrador de Django.
- Ingresar datos de prueba en cada modelo.
- El proyecto debe entregarse comprimido (.zip o .rar).

Instrumento de Evaluación:

Criterios	En 100%	Sobre 50%	Bajo 50%	No entregado
Configura correctamente la base de datos en Django	20	12	6	0
Define modelos en coherencia con la problemática	20	12	6	0
Implementa relaciones ForeignKey entre modelos	20	12	6	0
Realiza migraciones y crea las tablas en la BD	10	6	2	0
Registra los modelos en el administrador	10	6	2	0
Personaliza el administrador (columnas, filtros, búsqueda)	10	6	2	0
Ingresar registros de prueba en el administrador	10	6	2	0
Crea enlace entre la web y el administrador	10	6	2	0