- Análisis del código fuente y patrones de diseño:

Dentro del código base provisto, deberán identificar y analizar el uso de un patrón de diseño que ya se encuentra implementado.
Expliquen:
¿Qué patrón identificaron?
El patrón que identificamos es uno de la familia de los creacionales ya que se nos pide “crear” un docente dentro del código, en este caso el patrón es Singleton.
¿Qué problema resuelve este patrón en ese contexto?
El problema que resuelve el Singleton es evitar que la creación del profesor falle o que se introduzcan datos inconsistentes debido a que diferentes partes del código están accediendo a diferentes instancias de un recurso compartido (como la conexión a la base de datos o la configuración de validación).
El patrón garantiza la unicidad y el acceso centralizado, haciendo que el proceso de alta sea robusto y predecible.

- Implementación de Historias de Usuario (HU): Cada equipo deberá implementar la siguiente Historia de Usuario en el proyecto.

ID de HU	001
Título	Alta de profesor al sistema
Declaración	Como administrador del sistema, quiero registrar un nuevo profesor ingresando su información personal, para poder asignarlo a las asignaturas correspondientes dentro de una carrera.
Descripción Detallada	El Administrador accederá a un formulario donde deberá ingresar la información personal del nuevo profesor. El sistema debe asegurar que los campos Nombre, Apellido, Correo y DNI sean completados y que el formato del correo sea válido. Antes de la persistencia, el sistema debe consultar la base de datos para garantizar que el DNI y el Correo sean únicos. Si todas las validaciones son exitosas, el profesor se guarda y se muestra un mensaje de confirmación; de lo contrario, se muestra un mensaje de error claro al usuario. El formulario debe permitir al usuario cancelar la operación en cualquier momento.
Criterios de Validación (Criterios de Aceptación)	• Flujo exitoso: Al completar todos los campos obligatorios (nombre, apellido, correo, DNI) con datos válidos y guardar, el sistema muestra un mensaje de éxito..

	• Validaciones de Datos: El sistema debe impedir el registro si:
	  ☐ Faltan campos obligatorios.
	  ☐ Si el formato del correo electrónico no es válido.
	  ☐ El correo electrónico o el DNI ya existen en la base de datos.
	• Manejo de Errores: Si alguna validación falla, el sistema debe mostrar un mensaje de error claro, sin permitir que se guarde el formulario.
	• Acción de Cancelar: El formulario debe incluir un botón "Cancelar" que elimine todos los datos ingresados y devuelva al usuario a la pantalla anterior.

Tareas Asociadas a la Implementación	1. Implementación de la Entidad/Modelo: Crear o modificar la clase `Profesor.java` con los atributos necesarios. 2. Implementación del Formulario: Diseñar la interfaz de usuario con todos los campos de entrada y los botones Guardar y Cancelar. 3. Implementación del Controlador: Crear el controlador que gestiona la solicitud, recibe los datos y aplica la lógica de negocio que incluye todas las validaciones. 4. Manejo de Errores y Mensajes: Implementar la lógica para devolver y mostrar mensajes de éxito o error claros según el resultado de las validaciones y la persistencia.