CODIGO: EIF509
NOMBRE: Desarrollo de Aplicaciones Web
VALOR: 15%
ENTREGA: 10 de Septiembre 2016

Escuela de Informática
CICLO: II ciclo 2017
PROFESOR: Maikol Guzmán Alán, MPM

- **Técnicas**:
    - **Leer** cuidadosamente el proyecto
    - Seguir los estándares para presentar proyectos (orden del código, documentación)

Contenido a evaluar

- Diseño en capas
    - Persistencia
        - Models
        - DAO
    - Servicios
- Frameworks
    - ORM / Hibernate
    - Spring
        - Testing
        - AOP
        - Error Handling
        - Logging (LogBack)
        - Git
        - Maven

Descripción del examen

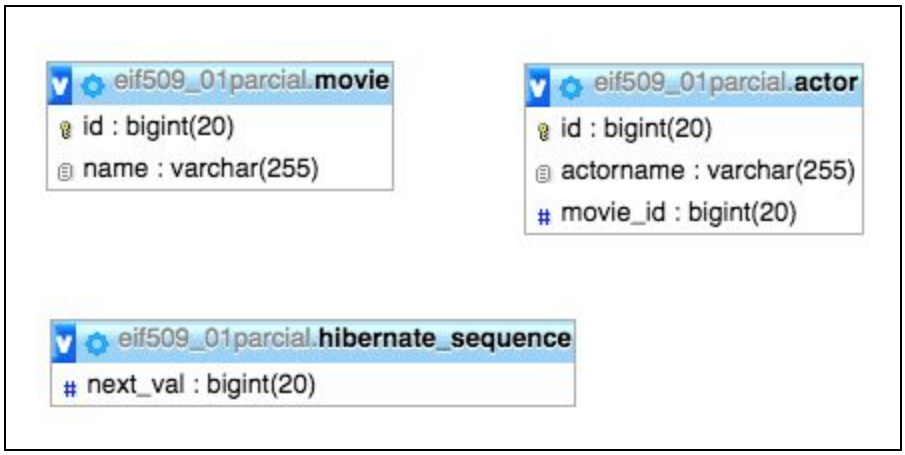
Desarrollar el examen a partir de la **prueba de integración** `VideoServiceTest.java`

El estudiante tiene que desarrollar un análisis inverso a partir del código fuente de la prueba de integración, para lograr desarrollar todas las capas y configuraciones necesarias para desarrollar el examen.

Persistencia

Base de Datos

El siguiente es un diagrama de Entidad Relación



BD: **eif509_01parcial**
User: **eif509db**
Pass: **1234$una**

Detalles:

- La base de datos se crea directamente desde el código (NO ES NECESARIO CREAR LA BD A PARTIR DE UN SCRIPT SQL)
- El diagrama contribuye para el desarrollo de los modelos

[Arquitectura en Capas]

Desarrollar los modelos necesarios con la configuración necesaria de JPA con Hibernate 4

Se solicita lo siguiente:

1. **Modelos**
    a. Configuración de hibernate basada en lo siguiente:
        i. **Actor**
            1. Debe de existir una relación **@ManyToOne** a **Movie**
            2. **CascadeType.ALL**
        ii. **Movie**
            1. Debe existir una relación **@OneToMany** a una lista de **Actor**
            2. **fetch = FetchType.EAGER**
            3. **orphanRemoval = true**
            4. **mappedBy = "movie"**
2. **DAO's**
    a. Definición e implementación del patrón necesario para desarrollar los DAO's
    b. Definición de los métodos necesarios definidos en la prueba de integración
    c. Configuraciones necesarias de Spring y Hibernate
3. **Services**
    a. Definición e implementación del patrón necesario para desarrollar los services
    b. Definición de los métodos necesarios definidos en la prueba de integración para el VideoService

[AOP]

Implementar AOP para logging de información

Se solicita lo siguiente:

1. Definir un **pointcut** **únicamente** para el método `deleteAllMovies`
2. Loguear en un archivo el nombre y el método antes y después de llamar al método seleccionado en el pointcut
    a. El mensaje debe de empezar con el texto **"@Before [EXAMEN-AOP]**

[Unit Testing]
Crear el Unit Testing para Service **Se solicita lo siguiente:** **1.** Desarrollar la **prueba de unidad** para **TODOS** los métodos definidos en el service 2. Se debe usar mockito para mockear los objetos necesarios

Evaluación:

Detalle	Puntaje
[BÁSICO] Configuración básica y funcional de todos los frameworks Hibernate, Spring, LogBack, AOP, Test	**2**
[BÁSICO] Uso de GitHub, al menos 5 commits reportados en el sistema	**1**
[MODELOS] Definición correcta del modelo y sus relaciones	**6**
[DAO] Definición correcta (interfaces e implementadores) de los métodos necesarios	**6**
[SERVICE] Definición correcta (interfaces e implementadores) de los métodos necesarios	**6**
[AOP] Implementación correcta del patrón	**2**
[UNIT TEST] Unit Test de todos los métodos necesarios en el service	**2**
[EXECUTE] Integration Test debe de correr	**4**

Reglas:

1. El estudiante debe de clonar la base del examen desde el GitHub y trabajar desde la base presentada.
    a. GitHub Link: https://classroom.github.com/a/y9fybqm6
2. El estudiante debe de hacer el commit en el repositorio indicado y tener claro la fecha final