

## UTR-E4 Proyecto Universo Cinema (UniCine)

Se recuerda que la práctica propuesta es la base para la calificación de varios criterios de evaluación. Por lo que un desarrollo deficiente afectará a futuras calificaciones.

Deberás construir una aplicación sencilla para la gestión de una sala de cine. El alumno debe proponer una solución individual que se ajuste al modelo y reglas de negocio indicadas. Al ser la base para otras prácticas además deberá cumplir con los siguientes criterios.

- Proyecto de Winforms
- Define una **clase única encargada de gestionar el acceso a la bd y las reglas de negocio.**
  - Esta clase es la base para el apartado de pruebas unitarias por lo que deberás definir dos constructores. Uno sin parámetros y otro que reciba el contexto de datos. Este último constructor ÚNICAMENTE SE EMPLEARÁ EN LAS PRUEBAS UNITARIAS.
  - EXISTE UNA PROPIEDAD O MÉTODO PRIVADO que devuelve una instancia nueva del contexto o el contexto para pruebas unitarias si este no es nulo.
  - CADA MÉTODO TENDRÁ UNA INSTANCIA NUEVA DEL CONTEXTO.
  - LOS MÉTODOS NO SE LLAMARÁN ENTRE SI, todas las validaciones y acceso a datos se realizan con el mismo objeto.
- Define una excepción propia de la aplicación, esta excepción será gestionada por la aplicación de manera que no se rompa mostrando la información asociada al usuario.
- Las excepciones de negocio tendrán un mensaje descriptivo para el usuario que le permita identificar el error.
- Define una clase única encargada del código de apoyo para los buscadores que necesites.
- El namespace del proyecto será UniCine\_TUNOMBRE.

### Criterios de calificación.

De la solución aportada por el alumno se valorará que sea funcional, completa, acorde con los principios vistos durante el curso y usable.

Se van a considerar errores graves de programación que impidan la superación de la práctica los siguientes:

- No lanzar excepciones de negocio.
- No gestionar excepciones de negocio adecuadamente.
- Acceso a datos desde múltiples contextos.
- No utilizar los métodos linq apropiados.

### Modelo

#### Peliculas

- PeliculaId, pk
- Nombre
- Categoría [enum] -> Todos, >12, >16, >18
- Genero [enum] -> Acción, Aventura, Terror, Comedia, Romance

- Duracion (minutos)
- Anno -> año de estreno
- Sinopsis

### **Sesiones**

- SesionId, pk
- Sala [enum] -> Sala1, Sala2, Sala3, Sala4, Sala5, Sala6, Sala7
- DiaSemana [enum] -> Lunes, Martes, Miércoles, Jueves, Viernes, Sábado, Domingo
- Comienzo -> hora de comienzo de la sesión
- FinMax -> hora límite de finalización de la sesión
- Precio [real]
- Aforo [entero]

### **Proyecciones**

- Sesionid, pk
- PeliculaId, pk
- Inicio, pk -> fecha de inicio proyección
- Fin (null)-> fecha de fin proyección

### **Reglas de negocio**

#### **Películas**

- Al borrar. Si una película tiene proyecciones asociadas lanza excepción.
- Al modificar validar que la duración no exceda la hora de finalización de la sesión. -> lanza excepción.
- Duración > 0 y 300 -> validar en el formulario

#### **Sesiones**

- Al modificar validar que el tiempo de la sesión sea mayor o igual a la duración de las películas asociadas -> lanzar excepción.
- Aforo > 0 -> Validar en el formulario
- Precio >= 0 -> Validar en el formulario

#### **Proyecciones**

- Al crear/modificar validar que duración de la película sea inferior o igual al tiempo de la sesión. -> Lanzar excepción
- Al crear modificar validar que no exista otra proyección con la misma sesión que se solape con en fechas. -> Lanzar excepción.
- La fecha de inicio debe ser menor o igual que la fecha de fin. -> Lanzar excepción.