

# Common Room

Universidad Tecnológica Nacional

Facultad Regional Mar del Plata

Tecnicatura Universitaria en Programación

## Gestión de Proyectos de Software

- Comisión 3
- *Adrian Aroca*

## Programación IV

- Comisión 2
- *Lucrecia Bazan*

## Integrantes

Fenoy Rivas, Delfina  
Francano, Ian  
Pérez, Lola

## Introducción

Nuestro proyecto, de nombre **Common Room**, es una aplicación web full stack desarrollada como parte de las materias **Programación IV, Gestión de Desarrollo del Software** de la Tecnicatura Universitaria en Programación de la UTN – Facultad Regional Mar del Plata.

En este documento se abordan aspectos vinculados a la planificación del proyecto, el análisis funcional, la estimación del tamaño mediante el método de **Puntos de Función (PFA)**, y la estimación de esfuerzo y costo utilizando el modelo **COCOMO**.

---

## Contexto del Proyecto

El desarrollo de **Common Room** surgió originalmente en el marco de las materias **Programación III** y **Programación IV**, donde el desafío era construir una aplicación cliente-servidor que combinara tecnologías modernas y fomentará la integración de distintas capas de desarrollo.

Durante su implementación, el equipo trabajó en el diseño del backend con **Spring Boot (Java)**, junto con un frontend basado en **HTML, CSS, JavaScript y Thymeleaf**.

A partir de esta base, en el trabajo actual se busca analizar el proyecto desde la perspectiva de **Gestión de Proyectos**, aplicando técnicas de estimación, planificación y análisis de alcance utilizadas en entornos profesionales.

---

## Opción elegida: Full Stack

Para el desarrollo de este trabajo se optó por la **Opción A – Enfoque Full Stack**, ya que el sistema contempla tanto el desarrollo del **backend (Spring Boot – Java)** como del **frontend (Angular / HTML / CSS / JS)**.

---

## Misión y Alcance del Proyecto

### Misión

La misión de **Common Room** es ofrecer una plataforma web en la que los usuarios puedan compartir sus opiniones, reseñas y valoraciones sobre películas, fomentando el intercambio y la participación dentro de una comunidad cinéfila activa.

El sistema busca brindar una experiencia intuitiva, dinámica y confiable, que facilite la interacción entre usuarios mediante reseñas, calificaciones y perfiles públicos.

## Alcance

El alcance del proyecto incluye el desarrollo de las siguientes funcionalidades principales:

- Registro y autenticación de usuarios (*con distintos roles: visitante, usuario y moderador*).
  - Creación, edición y eliminación de reseñas de películas.
  - Visualización detallada de información sobre películas obtenida desde la API externa TheMovieDB.
  - Búsqueda y filtrado de películas por título o categoría.
  - Visualización de perfiles de usuario y sus reseñas publicadas.
  - Gestión básica de usuarios y mantenimiento de datos desde la interfaz administrativa.
- 

## Descripción General del Sistema

**Common Room** es una aplicación web diseñada bajo una arquitectura **cliente-servidor**, que combina un backend robusto y un frontend dinámico.

El **backend**, desarrollado en **Spring Boot (Java 21)**, se encarga de la lógica de negocio, la autenticación mediante **tokens JWT**, la persistencia de datos en **MySQL** y la comunicación con la **API externa TheMovieDB** para obtener información actualizada de películas.

El **frontend**, implementado en **Angular**, brinda una interfaz visual moderna e intuitiva, permitiendo a los usuarios interactuar de manera sencilla con el sistema, realizar búsquedas, escribir reseñas y explorar el contenido disponible.

El sistema fue pensado para ser escalable, modular y adaptable a futuras mejoras, con el objetivo de ofrecer una experiencia fluida y segura tanto para usuarios registrados como visitantes.

# Requisitos Funcionales

## Listado de Requisitos

- **RF-01** → El Sistema debe permitir al Visitante registrarse.
- **RF-02** → El Sistema debe permitir al Miembro y Moderador iniciar sesión.
- **RF-03** → El Sistema debe permitir al Miembro y Moderador cerrar sesión.
- **RF-04** → El Sistema debe permitir al Miembro y Moderador realizar una alta de reseñas.
- **RF-05** → El Sistema debe permitir al Miembro y Moderador realizar una baja de reseñas propias.
- **RF-06** → El Sistema debe permitir al Miembro y Moderador modificar reseñas propias.
- **RF-07** → El Sistema debe permitir al Visitante, Miembro y Moderador visualizar reseñas de una película.
- **RF-08** → El Sistema debe permitir al Visitante, Miembro y Moderador visualizar reseñas de un usuario.
- **RF-09** → El Sistema debe permitir al Miembro y Moderador visualizar reseñas propias.
- **RF-10** → El Sistema debe permitir al Visitante, Miembro y Moderador visualizar películas.
- **RF-11** → El Sistema debe permitir al Visitante, Miembro y Moderador visualizar una película.
- **RF-12** → El Sistema debe permitir al Visitante, Miembro y Moderador filtrar películas.
- **RF-13** → El Sistema debe permitir al Moderador realizar una baja de reseñas.
- **RF-14** → El Sistema debe permitir al Visitante, Miembro y Moderador visualizar usuarios.
- **RF-15** → El Sistema debe permitir al Visitante, Miembro y Moderador visualizar un perfil de usuario.
- **RF-16** → El Sistema debe permitir al Miembro y Moderador visualizar perfil propio.
- **RF-17** → El Sistema debe permitir al Miembro y Moderador modificar perfil propio.
- **RF-18** → El Sistema debe permitir al Miembro y Moderador eliminar perfil propio.
  
- **RNF-01** → El Sistema debe incluir un Administrador ya creado.
- **RNF-02** → El Sistema debe tener una conexión a la API de “*TheMovieDB*”.
- **RNF-03** → El Sistema debe incluir una interfaz gráfica fácil y simple de entender

## Requisitos funcionales detallados

Campo	Descripción
<i>Código</i>	<b>RF-01</b>
<i>Nombre</i>	<b>Registro de Usuarios</b>
<i>Descripción Detallada</i>	El sistema debe permitir que los usuarios se registren mediante un nombre de usuario, email, contraseña, ícono y descripción opcional.
<i>Contexto</i>	Requisito base para <b>RF-02 (Inicio de Sesión)</b> y <b>RF-03 (Gestión de Reseñas)</b> .
<i>Objetivo</i>	Permitir que nuevos usuarios creen su cuenta de forma segura para acceder a funcionalidades personalizadas como reseñas y listas.
<i>Entradas</i>	- Username ( <i>String, entre 5 y 20 caracteres</i> ) - Email ( <i>String, formato válido, máx. 50 caracteres</i> ) - Password ( <i>String, mínimo 8 caracteres</i> ) - Profile Picture URL ( <i>String, opcional</i> ) - Description ( <i>String, máx. 255 caracteres, opcional</i> )
<i>Salidas</i>	- Token de sesión ( <i>JWT</i> ) - Datos del usuario autenticado - Mensaje de error si el registro falla
<i>Proceso</i>	1. Usuario completa el formulario de registro. 2. Sistema valida los datos ingresados. 3. Verifica que el <i>username</i> y el email no existan. 4. Si todo es válido, guarda el nuevo usuario y genera un token. 5. Si ocurre un error, muestra un mensaje de validación o error general.
<i>Reglas de negocio</i>	- Contraseñas encriptadas ( <i>bcrypt</i> ). - Token de sesión con vigencia de 24 h. - Username y email deben ser únicos.
<i>Prioridad</i>	<b>Alta</b>
<i>Notas adicionales</i>	Este requisito es esencial para el acceso de todos los perfiles.

Campo	Descripción
<i>Código</i>	<b>RF-02</b>
<i>Nombre</i>	<b>Inicio de Sesión</b>
<i>Descripción Detallada</i>	El sistema debe permitir que los usuarios ingresen con su nombre de usuario y contraseña para acceder a su cuenta.
<i>Contexto</i>	Requisito base para <b>RF-03 (Gestión de Reseñas)</b> y <b>RF-05 (Visualizar Perfil)</b> .
<i>Objetivo</i>	Autenticar usuarios registrados y otorgar acceso a las funcionalidades personalizadas del sistema.
<i>Entradas</i>	<ul style="list-style-type: none"> <li>- Username (<i>String, entre 5 y 20 caracteres</i>)</li> <li>- Password (<i>String, mínimo 8 caracteres</i>)</li> </ul>
<i>Salidas</i>	<ul style="list-style-type: none"> <li>- Token de sesión (<i>JWT</i>)</li> <li>- Datos del usuario autenticado</li> <li>- Mensaje de error si las credenciales son inválidas</li> </ul>
<i>Proceso</i>	<ol style="list-style-type: none"> <li>1. Usuario ingresa credenciales en formulario de login.</li> <li>2. Sistema valida el formato de los datos.</li> <li>3. Consulta la BD para verificar la existencia del usuario.</li> <li>4. Si las credenciales son válidas, genera token y redirige al <i>Home</i>.</li> <li>5. Si son inválidas, muestra un mensaje de error.</li> </ol>
<i>Reglas de negocio</i>	<ul style="list-style-type: none"> <li>- Contraseñas encriptadas (<i>bcrypt</i>).</li> <li>- Token con vigencia de 24 horas.</li> </ul>
<i>Prioridad</i>	<b>Alta</b>
<i>Notas adicionales</i>	X

Campo	Descripción
Código	<b>RF-03</b>
Nombre	<b>Gestión de Reseñas (ABM)</b>
Descripción Detallada	El sistema debe permitir a los usuarios autenticados crear, leer, modificar y eliminar sus propias reseñas sobre películas.
Contexto	Este requisito es base para <a href="#"><b>RF-04 (Visualizar ficha de una película)</b></a> y <a href="#"><b>RF-05 (Visualizar perfil de usuarios)</b></a> .
Objetivo	Fomenta la interacción y la generación de contenido por parte de la comunidad, proveyendo valoraciones que ayuden a otros usuarios.
Entradas	<p><b>Alta:</b></p> <ul style="list-style-type: none"> <li>• rating (doublé, entre 0.5 y 5)</li> <li>• description (string, máximo 700 caracteres)</li> <li>• movield (long, puesto por el sistema)</li> </ul> <p><b>Baja:</b></p> <ul style="list-style-type: none"> <li>• reviewId (long, puesto por el sistema)</li> </ul> <p><b>Modificación:</b></p> <ul style="list-style-type: none"> <li>• reviewId (long, puesto por el sistema)</li> <li>• user (User, para saber si tiene el permiso)</li> </ul> <p><b>Consulta:</b></p> <ul style="list-style-type: none"> <li>• movield (long, por la ficha de película)</li> <li>• userId (long, por el perfil del usuario)</li> </ul>
Salidas	<p><b>Alta</b> → Confirmación de alta.</p> <p><b>Baja</b> → Confirmación de eliminación.</p> <p><b>Modificación</b> → Confirmación de cambios, datos actualizados.</p> <p><b>Consulta</b> → Lista de las review según el ID utilizado</p>
Proceso	<p><b>Alta:</b></p> <ol style="list-style-type: none"> <li>1. Usuario ingresa descripción y puntaje en un “model” de nueva reseña.</li> <li>2. Sistema valida que se haya ingresado un rating y la descripción.</li> <li>3. Sistema agrega a la BD la reseña y le incluye el ID de la película.</li> </ol> <p><b>Baja:</b></p> <ol style="list-style-type: none"> <li>1. Usuario selecciona la review propia a eliminar</li> <li>2. Sistema elimina de la base de datos la review.</li> </ol> <p><b>Modificación:</b></p> <ol style="list-style-type: none"> <li>1. Usuario selecciona la review a editar.</li> <li>2. Usuario ingresa descripción o puntaje a modificar.</li> <li>3. Sistema valida los cambios.</li> <li>4. Sistema actualiza en la BD la reseña</li> </ol> <p><b>Consulta:</b></p> <ol style="list-style-type: none"> <li>1. Usuario visualiza la ficha de una película</li> <li>2. Sistema muestra las reseñas relacionadas a esa película</li> <li>3. Usuarios visualiza el perfil de otro usuario</li> <li>4. Sistema muestra las reseñas relacionadas a ese Perfil</li> </ol>
Reglas de negocio	<ul style="list-style-type: none"> <li>- Un usuario solo puede publicar una reseña por película.</li> <li>- El rating debe estar entre 0.5 y 5, en incrementos de 0.5.</li> </ul>
Prioridad	<b>Alta</b>

Campo	Descripción
<i>Código</i>	<b>RF-04</b>
<i>Nombre</i>	<b>Visualizar ficha de una película</b>
<i>Descripción Detallada</i>	Permite a usuarios y visitantes ver la ficha completa de una película, incluyendo su información general y reseñas asociadas.
<i>Contexto</i>	Base para <b>RF-06 (Búsqueda y filtrado de películas)</b> .
<i>Objetivo</i>	Brindar información completa sobre una película, incluyendo reseñas y valoraciones.
<i>Entradas</i>	- movield ( <i>Long, obtenido desde la URL</i> )
<i>Salidas</i>	<ul style="list-style-type: none"> <li>- Datos de la película (<i>id, título, sinopsis, duración, fecha, promedio, poster, etc.</i>)</li> <li>- Listado de reseñas asociadas</li> </ul>
<i>Proceso</i>	<ol style="list-style-type: none"> <li>1. Usuario selecciona película desde el listado o buscador.</li> <li>2. Sistema consulta la BD/API para obtener datos.</li> <li>3. Recupera reseñas relacionadas.</li> <li>4. Muestra la información completa.</li> <li>5. Si el usuario está autenticado, se habilita el botón “Review +”.</li> <li>6. Si hay error, se redirige a “Error404”.</li> </ol>
<i>Reglas de negocio</i>	<ul style="list-style-type: none"> <li>- Solo películas activas o disponibles serán visibles.</li> <li>- Se mostrarán reseñas de usuarios activos.</li> <li>- Visitantes pueden ver fichas pero no agregar reseñas.</li> </ul>
<i>Prioridad</i>	<b>Alta</b>
<i>Notas adicionales</i>	<ul style="list-style-type: none"> <li>- Toda la información de la película es extraída desde la API “TMDB”, únicamente las reseñas salen de la Base de Datos MySQL</li> </ul>

Campo	Descripción
<i>Código</i>	<b>RF-05</b>
<i>Nombre</i>	<b>Visualizar perfil de usuarios</b>
<i>Descripción Detallada</i>	Permite visualizar el perfil público de un usuario, mostrando su información básica, reseñas y listas creadas.
<i>Contexto</i>	Depende de <b>RF-02 (Inicio de Sesión)</b> para determinar permisos y autenticación.
<i>Objetivo</i>	Permitir consultar perfiles públicos y fomentar interacción entre usuarios.
<i>Entradas</i>	<ul style="list-style-type: none"> <li>- username (<i>String, desde la URL</i>)</li> <li>- token JWT (<i>String, opcional</i>)</li> </ul>
<i>Salidas</i>	<ul style="list-style-type: none"> <li>- username (<i>String</i>)</li> <li>- description (<i>String</i>)</li> <li>- profilePictureUrl (<i>String</i>)</li> <li>- createdAt (<i>LocalDateTime</i>)</li> <li>- Listado de reseñas del usuario</li> <li>- Botón “Edit Profile” visible para el propietario</li> </ul>
<i>Proceso</i>	<ol style="list-style-type: none"> <li>1. Usuario accede al perfil desde menú o reseña.</li> <li>2. Sistema obtiene datos públicos del usuario.</li> <li>3. Carga reseñas del perfil.</li> <li>4. Si el usuario autenticado es el dueño, muestra el botón “Edit Profile”.</li> <li>5. Si no existe o falla, muestra modal de error.</li> </ol>
<i>Reglas de negocio</i>	<ul style="list-style-type: none"> <li>- Los perfiles son públicos pero limitados a datos no sensibles.</li> <li>- Solo el propietario ve el botón “Edit Profile”.</li> <li>- Las reseñas mostradas deben pertenecer al usuario visualizado.</li> <li>- Si no hay reseñas, se muestra mensaje informativo.</li> </ul>
<i>Prioridad</i>	<b>Alta</b>
<i>Notas adicionales</i>	X

Campo	Descripción
<i>Código</i>	<b>RF-06</b>
<i>Nombre</i>	<b>Búsqueda y Filtrado de películas</b>
<i>Descripción Detallada</i>	Permite a usuarios y visitantes buscar películas por título y filtrar los resultados provenientes de la API externa.
<i>Contexto</i>	Depende de <u><a href="#">RF-04 (Visualizar Ficha de Película)</a></u> .
<i>Objetivo</i>	Facilitar la exploración del catálogo mediante búsqueda y paginación de resultados.
<i>Entradas</i>	<ul style="list-style-type: none"> <li>- query (<i>String, desde la barra de búsqueda</i>)</li> <li>- page (<i>Integer, opcional para paginación</i>)</li> </ul>
<i>Salidas</i>	<ul style="list-style-type: none"> <li>- Lista de películas (<i>id, título, sinopsis, fecha, póster</i>)</li> <li>- Datos de paginación (<i>totalPages, currentPage</i>)</li> </ul>
<i>Proceso</i>	<ol style="list-style-type: none"> <li>1. Usuario ingresa texto en la barra de búsqueda.</li> <li>2. Sistema consulta la API externa.</li> <li>3. Procesa resultados y los muestra en lista.</li> <li>4. Si no hay resultados, muestra el mensaje “No se encontraron películas...”.</li> <li>5. Si hay varias páginas, genera botones de navegación.</li> <li>6. Al seleccionar una película, redirige a su ficha detallada.</li> </ol>
<i>Reglas de negocio</i>	<ul style="list-style-type: none"> <li>- La búsqueda sólo se ejecuta si el campo no está vacío.</li> <li>- Solo se muestran películas activas o disponibles.</li> <li>- El sistema debe paginar resultados.</li> <li>- Cualquier usuario (visitante o autenticado) puede buscar.</li> </ul>
<i>Prioridad</i>	<b>Alta</b>
Notas adicionales	X

# Medición Funcional (PFA)

## Tabla de conteo

<b>Tipo de Función</b>	<b>Descripción</b>	<b>Complejidad (con DETs y FTRs)</b>	<b>Puntos Asignados</b>	<b>Justificación</b>
EI	<b>Alta de Usuario (Registro)</b> Permite crear un nuevo usuario.	Media (≈ 5 DETs, 1 FTR)	4	Alta de datos validados ( <i>nombre, email, password, ícono</i> ).
EI	<b>Inicio de Sesión (Login)</b> Autenticación de usuario mediante credenciales.	Media (≈ 3 DETs, 1 FTR)	4	Valida credenciales y genera token JWT.
EI	<b>Actualizar Perfil</b> Modificación de datos personales del usuario.	Media (≈ 4 DETs, 1 FTR)	4	Actualiza descripción, imagen, username, etc.
EI	<b>Cambiar contraseña</b> Permite modificar la clave de acceso.	Media (≈ 3 DETs, 1 FTR)	4	Verifica la actual, válida y guarda la nueva contraseña.
EI	<b>Crear Reseña</b> Carga de una nueva reseña por parte del usuario.	Media (≈ 4 DETs, 2 FTRs)	4	Relaciona usuario y película con texto y puntuación.
EI	<b>Modificar Reseña</b> Edición de una reseña existente.	Media (≈ 3 DETs, 2 FTRs)	4	Permite actualizar texto o rating.
EI	<b>Eliminar Reseña</b> Eliminación de una reseña propia.	Baja (≈ 2 DETs, 2 FTRs)	3	Quita reseña sin afectar otras entidades.
EO	<b>Listado de Películas Populares</b> Muestra películas desde la API externa.	Alta (≈ 7 DETs, 2 FTRs)	7	Llama a la API y muestra datos procesados.
EO	<b>Detalle de Película</b> Visualiza toda la información de una película.	Alta (≈ 8 DETs, 2 FTRs)	7	Incluye datos de la película y reseñas asociadas.
EO	<b>Listado de Reseñas por Película</b> Muestra todas las reseñas vinculadas a una película.	Media (≈ 5 DETs, 2 FTRs)	5	Consulta la base de datos y presenta resultados combinados.

<b>EO</b>	<b>Mostrar Perfil de Usuario</b> Visualiza información pública de un usuario.	<b>Media</b> (≈ 6 DETs, 2 FTRs)	<b>5</b>	Presenta datos personales y reseñas del usuario.
<b>EO</b>	<b>Mostrar Reseñas del Usuario</b> Listado de reseñas propias.	<b>Media</b> (≈ 5 DETs, 2 FTRs)	<b>5</b>	Consulta filtrada por userId.
<b>EQ</b>	<b>Búsqueda de Películas</b> Permite buscar películas por nombre o palabra clave.	<b>Alta</b> (≈ 6 DETs, 2 FTRs)	<b>6</b>	Realiza consulta a API externa con parámetros dinámicos.
<b>EQ</b>	<b>Filtrado por Género</b> Filtrar resultados según género o año.	<b>Media</b> (≈ 4 DETs, 2 FTRs)	<b>4</b>	Filtro mediante parámetros enviados a la API.
<b>EQ</b>	<b>Consulta de Usuarios</b> Obtienen información pública de usuarios.	<b>Media</b> (≈ 4 DETs, 2 FTRs)	<b>4</b>	Recupera listado de usuarios registrados visibles públicamente.
<b>ILF</b>	<b>USUARIOS</b> Archivo lógico con datos de los usuarios registrados.	<b>Media</b> (≈ 8 DETs, 1 FTR)	<b>10</b>	Contiene username, email, password, rol, fecha de alta, estado, etc.
<b>ILF</b>	<b>PELÍCULAS</b> Archivo con información de películas gestionadas en el sistema.	<b>Alta</b> (≈ 12 DETs, 2 FTRs)	<b>15</b>	Registra datos extensos: título, año, género, director, duración, rating, sinopsis, poster, etc.
<b>ILF</b>	<b>RESEÑAS</b> Archivo de reseñas de usuarios sobre películas.	<b>Media</b> (≈ 7 DETs, 1 FTR)	<b>10</b>	Registra id, user_id, movie_id, rating, comentario y fechas.
<b>EIF</b>	<b>API TMDB</b> Interfaz externa para obtener información de películas.	<b>Alta</b> (≈ 10 DETs, 1 FTR)	<b>10</b>	Interacción directa con la API de TheMovieDB (servicio externo).

---

### Ayuda memoria

Sigla	Significa	Qué representa	Ejemplo
DET	Data Element Type	Cantidad de campos o datos visibles al usuario	<i>username, email, rating, posterUrl, etc.</i>
FTR	File Type Referenced	Archivos/tablas accedidos o consultados por una función	<i>Usuarios, Reseñas, Películas, API TMDB</i>

---

### Factor de ajuste

Nº	Característica	Valor
1	<i>Comunicación de datos</i>	<b>4</b>
2	<i>Procesamiento distribuido</i>	<b>3</b>
3	<i>Requisitos de rendimiento</i>	<b>3</b>
4	<i>Configuración utilizada</i>	<b>2</b>
5	<i>Frecuencia de transacciones</i>	<b>3</b>
6	<i>Entrada de datos en línea</i>	<b>4</b>
7	<i>Eficiencia del usuario final</i>	<b>4</b>
8	<i>Actualización en línea</i>	<b>4</b>
9	<i>Procesamiento complejo</i>	<b>2</b>
10	<i>Reusabilidad del código</i>	<b>3</b>
11	<i>Facilidad de instalación</i>	<b>2</b>
12	<i>Facilidad de operación</i>	<b>3</b>
13	<i>Múltiples localizaciones</i>	<b>1</b>
14	<i>Facilidad de cambios</i>	<b>3</b>

## Datos finales:

- **Total PFSA = 115**
  - **Total FA = 41**
  - **Total PFA =  $115 * [0.65 + (0.01 * 41)] = 121.9 \approx \underline{122}$**
-

# Modelo COCOMO

## Definición de arquitectura y equipo:

Componente	Tecnología	PF	SLOC por PF	KSLOC
Backend	Java (SpringBoot)	80	50	4,00
Frontend	Angular (TypeScript/HTML/CSS)	35	40	1,40
<b>Total</b>		<b>115</b>		<b>5,40 KSLOC</b>

## Cálculos

Esfuerzo (Persona-Mes)

$$PM = 2,4 \times (5,40) ^ 1,05 = 14,10 \approx 15 \text{ meses-persona}$$

Tiempo de desarrollo

$$TDEV = 2,5 \times (15) ^ 0,35 = 6,45 \approx 7 \text{ meses}$$

Personas promedio necesarias

$$N = 15/7 = 2,14 \approx 3 \text{ personas}$$

# Técnica Tres Puntos

## Escenarios

RF-01 (Registro de Usuarios)

<b>Escenario</b>	<b>Duración</b>	<b>Justificación</b>
<b>Optimista (O)</b>	<u>2 días</u>	- El equipo ya tiene experiencia con Spring Security. - JWT es un proceso conocido. - No surgen problemas con la validación de datos.
<b>Más Probable (M)</b>	<u>3 días</u>	- Experiencia promedio del equipo. - Tiempo estándar para configurar el endpoint, el servicio de usuario y las validaciones. - Pequeños ajustes en la configuración de seguridad.
<b>Pesimista (P)</b>	<u>5 días</u>	- Problemas con la configuración del filtro de JWT. - Dificultades con las dependencias de seguridad. - Errores inesperados en la persistencia de datos con la contraseña encriptada.

RF-02 (Inicio de Sesión)

<b>Escenario</b>	<b>Duración</b>	<b>Justificación</b>
<b>Optimista (O)</b>	<u>1 días</u>	- Se reutiliza el 90% de la configuración de seguridad de RF-01. - La lógica de autenticación es directa. - No hay imprevistos.
<b>Más Probable (M)</b>	<u>2 días</u>	- Implementación estándar del endpoint de login. - Se realizan pruebas de autenticación para casos de éxito y error.
<b>Pesimista (P)</b>	<u>4 días</u>	- Problemas al verificar las contraseñas. - Conflictos en la cadena de filtros de Spring Security. - Dificultad para gestionar correctamente las excepciones de autenticación.

### RF-03 (Gestión de Reseñas)

<b>Escenario</b>	<b>Duración</b>	<b>Justificación</b>
<b>Optimista (O)</b>	<u>3 días</u>	<ul style="list-style-type: none"> <li>- El desarrollador tiene claro el manejo de relaciones en JPA.</li> <li>- Los endpoints CRUD se implementan rápidamente.</li> <li>- La lógica de negocio (<i>una reseña por usuario/película</i>) es simple de implementar.</li> </ul>
<b>Más Probable (M)</b>	<u>5 días</u>	<ul style="list-style-type: none"> <li>- Tiempo estándar para desarrollar los 4 endpoints (<i>Crear, Leer, Actualizar, Borrar</i>).</li> <li>- Implementación de la capa de servicio con validaciones y lógica de autorización para que un usuario solo modifique lo suyo.</li> </ul>
<b>Pesimista (P)</b>	<u>8 días</u>	<ul style="list-style-type: none"> <li>- Complejidad en el manejo de las relaciones entre Usuario, Película y Reseña.</li> <li>- Problemas con las transacciones de la base de datos.</li> <li>- La lógica de autorización para las operaciones de modificar y eliminar resulta más compleja de lo esperado.</li> </ul>

### RF-04 (Visualizar ficha de una película)

<b>Escenario</b>	<b>Duración</b>	<b>Justificación</b>
<b>Optimista (O)</b>	<u>2 días</u>	<ul style="list-style-type: none"> <li>- La consulta para obtener la película y sus reseñas asociadas es sencilla.</li> <li>- No hay problemas de rendimiento.</li> </ul>
<b>Más Probable (M)</b>	<u>3 días</u>	<ul style="list-style-type: none"> <li>- Se requiere optimizar la consulta para evitar problemas (cargar reseñas de forma eficiente).</li> <li>- Desarrollo estándar del endpoint y los DTOs (Data Transfer Objects) para la respuesta.</li> </ul>
<b>Pesimista (P)</b>	<u>6 días</u>	<ul style="list-style-type: none"> <li>- La consulta a la base de datos es lenta y requiere una optimización avanzada.</li> <li>- Problemas con la serialización de datos.</li> <li>- Se necesita refactorizar la estructura de los DTOs.</li> </ul>

#### RF-05 (Visualizar perfil de un usuario)

<b>Escenario</b>	<b>Duración</b>	<b>Justificación</b>
<b>Optimista (O)</b>	<u>2 días</u>	<ul style="list-style-type: none"> <li>- La lógica es muy similar a RF-04 y se reutiliza parte del código.</li> <li>- La consulta para obtener el usuario y sus reseñas es directa.</li> </ul>
<b>Más Probable (M)</b>	<u>3 días</u>	<ul style="list-style-type: none"> <li>- Implementación de la lógica para obtener datos públicos y la lista de reseñas.</li> <li>- Se añade la verificación para mostrar el botón "Edit Profile" si el usuario autenticado es el dueño.</li> </ul>
<b>Pesimista (P)</b>	<u>5 días</u>	<ul style="list-style-type: none"> <li>- Surgen problemas de rendimiento al cargar perfiles con muchas reseñas.</li> <li>- La lógica para diferenciar entre el dueño del perfil y un visitante genera bugs.</li> </ul>

#### RF-06 (Búsqueda y filtrado de Películas)

<b>Escenario</b>	<b>Duración</b>	<b>Justificación</b>
<b>Optimista (O)</b>	<u>2 días</u>	<ul style="list-style-type: none"> <li>- La API externa está bien documentada y es estable.</li> <li>- El mapeo de la respuesta de la API a los objetos Java es directo.</li> </ul>
<b>Más Probable (M)</b>	<u>4 días</u>	<ul style="list-style-type: none"> <li>- Tiempo estándar para configurar el cliente HTTP.</li> <li>- Implementación del manejo de la paginación y la lógica de búsqueda.</li> <li>- Gestión de errores básicos (ej. 404 si no hay resultados).</li> </ul>
<b>Pesimista (P)</b>	<u>7 días</u>	<ul style="list-style-type: none"> <li>- La API externa es inconsistente o tiene una documentación pobre.</li> <li>- Dificultades para mapear una respuesta JSON compleja.</li> <li>- Problemas con el manejo de errores de red, timeouts o límites de peticiones.</li> </ul>

## Cálculo Final

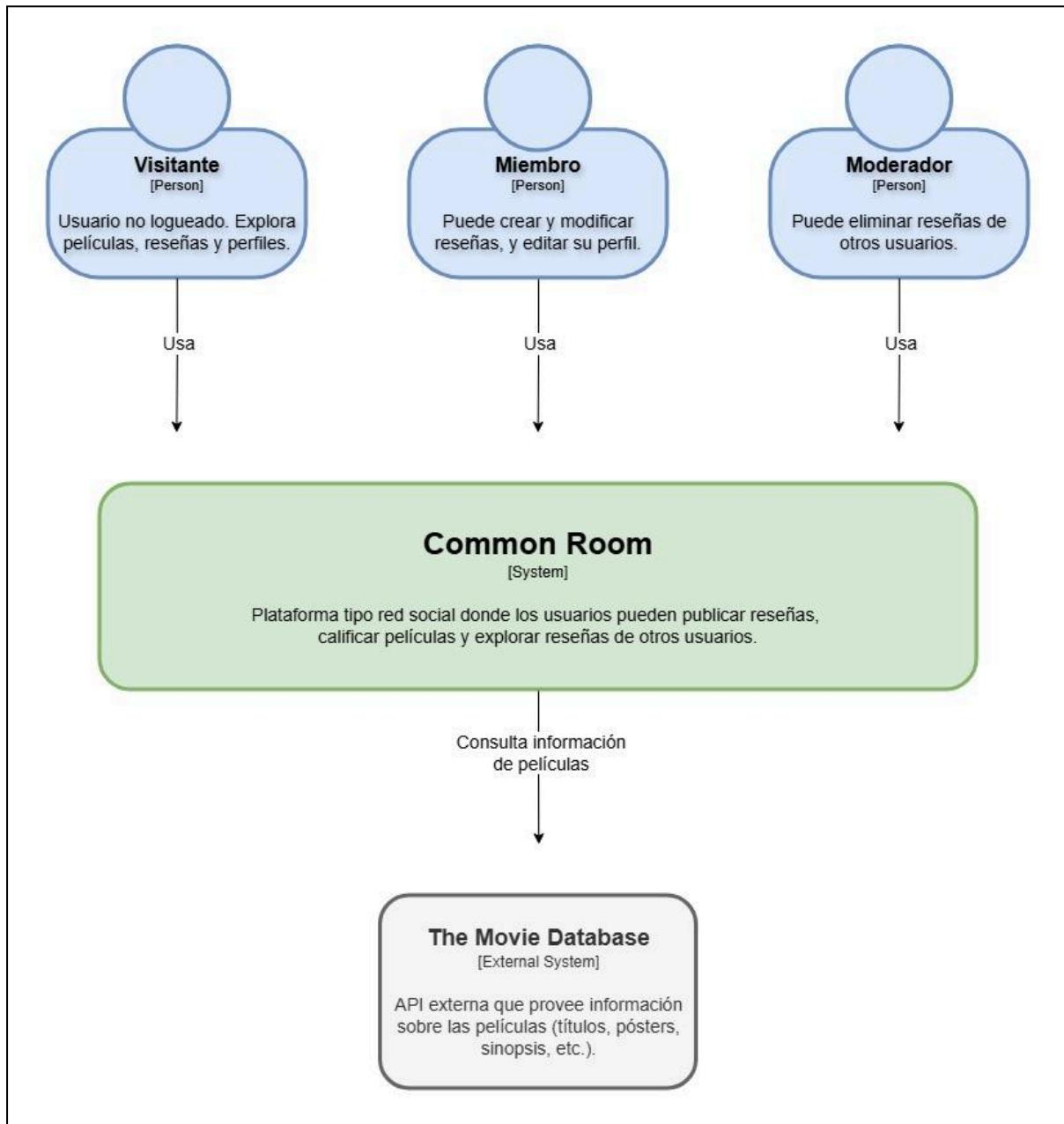
**Duración esperada =  $(O + 4M + P) \div 6$**

- **RF-01: Registró** =  $(2 + 4 \times 3 + 5) \div 6 = 19 \div 6 \approx 3.2 \text{ días}$
- **RF-02: Login** =  $(1 + 4 \times 2 + 4) \div 6 = 13 \div 6 \approx 2.2 \text{ días}$
- **RF-03: Gestión de Reseñas** =  $(3 + 4 \times 5 + 8) \div 6 = 31 \div 6 \approx 5.2 \text{ días}$
- **RF-04: Ficha de Pelicula**=  $(2 + 4 \times 3 + 6) \div 6 = 20 \div 6 \approx 3.3 \text{ días}$
- **RF-05: Perfil de Usuario** =  $(2 + 4 \times 3 + 5) \div 6 = 19 \div 6 \approx 3.2 \text{ días}$
- **RF-06: Busqueda** =  $(2 + 4 \times 4 + 7) \div 6 = 25 \div 6 \approx 4.2 \text{ días}$

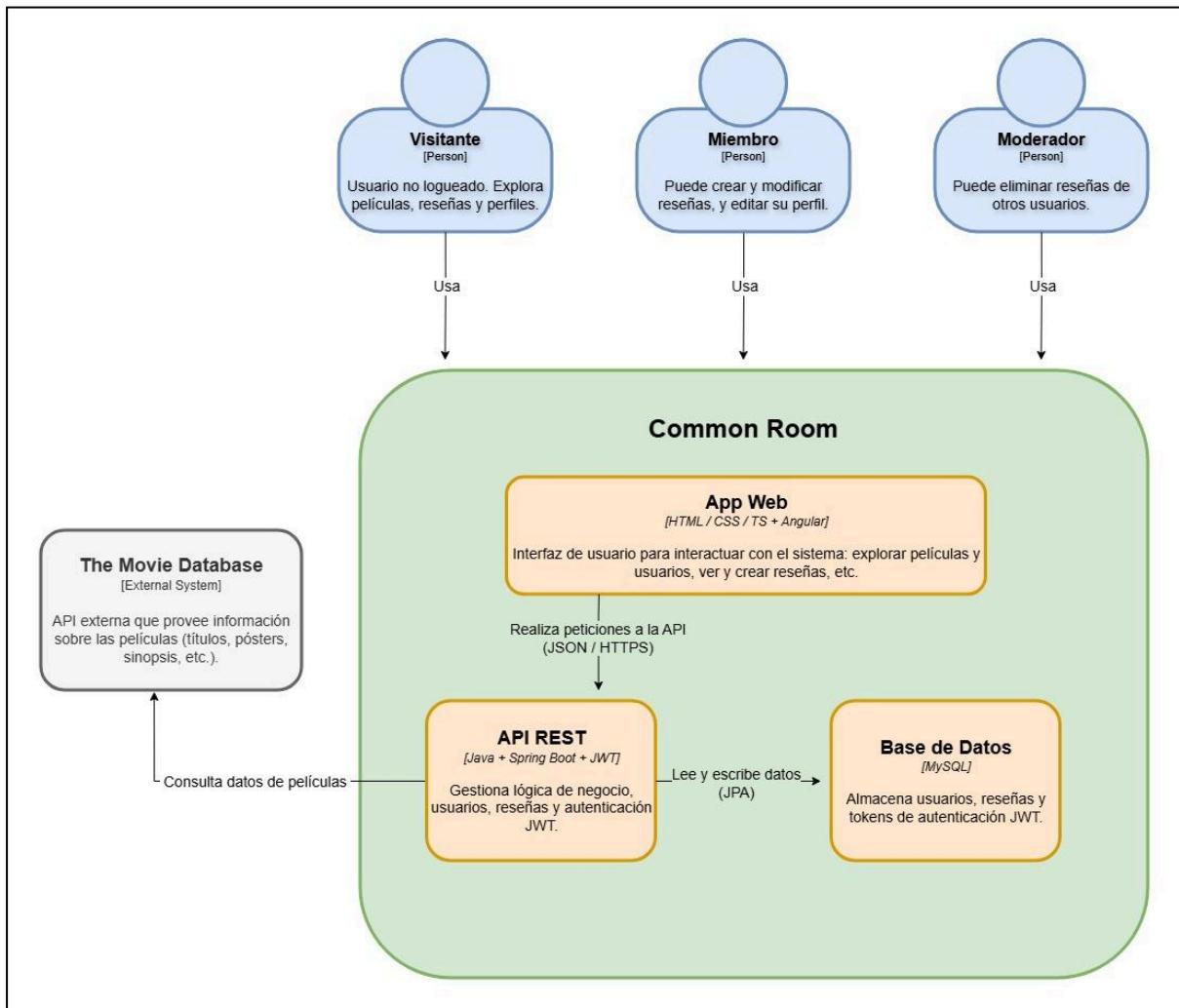
Total estimado de días = 21.3 días

# Arquitectura del Sistema

## Diagrama de contexto



## Diagrama de contenedores



# Gestion de Riesgos

<b>ID</b>	<b>Riesgo</b>	<b>Categoría</b>	<b>Probabilidad</b>	<b>Impacto</b>	<b>Exposición</b>	<b>Estrategia</b>	<b>Responsable</b>
<b>R - 01</b>	Problemas de integración entre el frontend y backend (CORS)	Técnico	<u>Media</u>	<u>Alto</u>	<b>Alta</b>	<b>Mitigar:</b> Realizar pruebas de integración tempranas y continuas.	<i>Ian Francano</i>
<b>R - 02</b>	Fallas o cambios inesperados en la API externa para la búsqueda de películas (RF - 06)	Técnico	<u>Media</u>	<u>Alto</u>	<b>Alta</b>	<b>Mitigar:</b> Desarrollar un manejo de errores robusto para informar al usuario de manera amigable si la API no responde.	<i>Delfina Fenoy</i>
<b>R - 03</b>	Bajo rendimiento en consultas a la base de datos al cargar perfiles y fichas de películas con sus reseñas	Técnico	<u>Alta</u>	<u>Medio</u>	<b>Alta</b>	<b>Mitigar:</b> Utilizar DTOs para devolver solo la información necesaria.	<i>Lola Pérez</i>
<b>R - 04</b>	Vulnerabilidades de seguridad en la implementación de la autenticación con JWT y la gestión de contraseñas	Técnico	<u>Media</u>	<u>Alto</u>	<b>Alta</b>	<b>Mitigar:</b> Almacenar el “secret key” de JWT de forma segura (variables de entorno)	<i>Delfina Fenoy</i>
<b>R - 05</b>	La lógica de negocio para la gestión de reseñas (RF - 03) resulta ser más compleja de lo previsto	Alcance	<u>Media</u>	<u>Alto</u>	<b>Alta</b>	<b>Mitigar:</b> Priorizar funcionalidades	<i>Delfina Fenoy</i>
<b>R - 06</b>	Las estimaciones de tiempo resultan ser demasiado optimistas, generando retrasos en la entrega	Tiempo	<u>Media</u>	<u>Alto</u>	<b>Alta</b>	<b>Mitigar:</b> Realizar seguimiento diario del progreso contra las estimaciones.	<i>Lola Pérez</i>
<b>R - 07</b>	Disponibilidad limitada de un desarrollador clave con conocimiento crítico en Spring Security	Equipo	<u>Baja</u>	<u>Alto</u>	<b>Medio</b>	<b>Mitigar:</b> Fomentar la programación en pareja en las tareas de seguridad.	<i>Ian Francano</i>