
Service Modeling based on SOLID Principles

The Hive

**Preparado por Ana Meza (Scrum Máster), Alejandro
Villarreal, & Isabella Ordosgoitia**

Universidad Tecnológica de Bolívar

11 de abril del 2025

1. User: Servicio de Streaming

1.1. Fase 1: Análisis y Diseño Funcional del Microservicio

Microservicio: *user-service*

Tareas Puntuales

- **Identificar el *bound context*:**
 - ✓ El dominio “*Usuario*” incluye usuario y perfil.
 - ✓ Separamos *user-service* de *profile-service*.
- **Definir la entidad principal:**
 - ✓ Entidad: *User*.
 - ✓ Atributos: *ID, name, last_name, birthdate, email, phone_number, password, status*.
- **Delimitar funcionalidad de microservicio:**
 - ✓ Crear usuario.
 - ✓ Actualizar usuario.
 - ✓ Mostrar usuario.
 - ✓ Eliminar usuario.
- **Diagramar caso de uso:**

Usuario se registra → se validan los datos → se crea perfil o internal user.

Usuario inicia sesión → se validan credenciales → obtiene acceso.

Usuario solicita cambio de contraseña → recibe enlace → actualiza contraseña.

Usuario elimina su cuenta → se marca como inactiva.

Usuario cierra sesión → pierde acceso.
- **Diseñar endpoint inicial (borrador):**
 - ✓ *POST* /users
 - ✓ *GET* /users/{id}
 - ✓ *POST* / users / {id} / update
 - ✓ *POST* / users / {id} / delete
- **Validar con *Stakeholders*:**

Revisión con líder de producto para ajustar estados válidos:

 - ✓ Created.
 - ✓ Eliminated.
 - ✓ Updated.

2. Profile: Servicio de Streaming

2.1. Fase 1: Análisis y Diseño Funcional del Microservicio

Microservicio: *profile-service*

Tareas Puntuales

- **Identificar el *bound context*:**
 - ✓ El dominio “*Usuario*” incluye usuario y perfil.
 - ✓ Separamos *profile-service* de *user-service*.
- **Definir la entidad principal:**
 - ✓ Entidad: *Subscriber*.
 - ✓ Atributos: *photo_url*, *username*, *description*, *lists*, *num_followers*, *num_followed*, *comments*, *subsRole*.
- **Delimitar funcionalidad de microservicio:**
 - ✓ Crear subscriber.
 - ✓ Actualizar subscriber.
 - ✓ Mostrar subscriber.
 - ✓ Eliminar subscriber.
- **Diagramar caso de uso:**

Usuario accede al sitio web → usuario almacena sus datos → usuario se convierte en subscriber.
- **Diseñar endpoint inicial (borrador):**
 - ✓ *POST* /profile
 - ✓ *GET* /profile/{id}
 - ✓ *POST* /profile/{id}/delete
 - ✓ *POST* /profile/{id}/update
- **Validar con *Stakeholders*:**

Revisión con líder de producto para ajustar estados válidos:

 - ✓ Created
 - ✓ Eliminated
 - ✓ Updated

3. Movie List: Servicio de Streaming

3.1. Fase 1: Análisis y Diseño Funcional del Microservicio

Microservicio: *movie-list-service*

Tareas Puntuales

- **Identificar el *bound context*:**
 - ✓ El dominio “*Contenido*” incluye películas, listas.
 - ✓ Separamos *movie-list-service* de *movie-service*.
- **Definir la entidad principal:**
 - ✓ Entidad: *MovieList*.
 - ✓ Atributos: *ID*, *user_id*, *privacy*, *list_name*, *list_description*, *like_by*, *saved_by*, *movies*.
- **Delimitar funcionalidad de microservicio:**
 - ✓ Crear lista de películas.
 - ✓ Agregar película.
 - ✓ Eliminar película.
 - ✓ Visualizar lista.
 - ✓ Editar lista.
 - ✓ Eliminar lista.
- **Diagramar caso de uso:**

Usuario accede al sitio web → usuario crea lista → usuario agrega películas a la lista → se almacena la información.
- **Diseñar endpoint inicial (borrador):**
 - ✓ *POST* /movielist
 - ✓ *POST* /movielist/{id}/movies
 - ✓ *PATCH* /movielist/{id}/movies/remove
 - ✓ *GET* /movielist/{id}/
 - ✓ *GET* /movielist
 - ✓ *PATCH* /movielist/{id}
 - ✓ *DELETE* /movielist
- **Validar con *Stakeholders*:**

Revisión con líder de producto para ajustar estados válidos:

 - ✓ Created

- ✓ Eliminated
- ✓ Updated

4. Article: Servicio de Streaming

4.1. Fase 1: Análisis y Diseño Funcional del Microservicio

Microservicio: *article-service*

Tareas Puntuales

- **Identificar el *bound context*:**
 - ✓ El dominio “*Noticias*” incluye artículos.
- **Definir la entidad principal:**
 - ✓ Entidad: *Article*.
 - ✓ Atributos: *ID, user_id, section_id, content, created_at, has_spoiler*.
- **Delimitar funcionalidad de microservicio:**
 - ✓ Crear artículo.
 - ✓ Eliminar artículo.
 - ✓ Editar artículo.
 - ✓ Visualizar artículo.
- **Diagramar caso de uso:**

Usuario interno accede al sitio web → usuario interno crea artículo → se almacena la información.
- **Diseñar endpoint inicial (borrador):**
 - ✓ *POST* /article
 - ✓ *GET* /article/{id}
 - ✓ *POST* /article/{id}/update
 - ✓ *POST* / article/{id}/delete
- **Validar con *Stakeholders*:**

Revisión con líder de producto para ajustar estados válidos:

 - ✓ Created
 - ✓ Eliminated
 - ✓ Updated

5. Comment: Servicio de Streaming

5.1. Fase 1: Análisis y Diseño Funcional del Microservicio

Microservicio: *comment-service*

Tareas Puntuales

- **Identificar el *bound context*:**
 - ✓ El dominio “*Interacciones*” incluye reacciones, comentarios, calificaciones y seguir.
 - ✓ Separamos *comment-service* de *reaction-service*, *rate-service*, y *follow-service*.
- **Definir la entidad principal:**
 - ✓ Entidad: *Comment*.
 - ✓ Atributos: *ID*, *user_id*, *movie_id*, *description*, *created_at*, *like_by*, *has_spoiler*.
- **Delimitar funcionalidad de microservicio:**
 - ✓ Crear comentario.
 - ✓ Eliminar comentario.
 - ✓ Editar comentario.
 - ✓ Visualizar comentario.
- **Diagramar caso de uso:**

Usuario accede al sitio web → usuario crea comentario → se almacena la información.
- **Diseñar endpoint inicial (borrador):**
 - ✓ *POST* /comment
 - ✓ *GET* /comment/{id}
 - ✓ *POST* /comment/{id}/update
 - ✓ *DELETE* /comment/{id}
- **Validar con *Stakeholders*:**

Revisión con líder de producto para ajustar estados válidos:

 - ✓ Created
 - ✓ Eliminated
 - ✓ Updated

6. Reaction: Servicio de Streaming

6.1. Fase 1: Análisis y Diseño Funcional del Microservicio

Microservicio: *reaction-service*

Tareas Puntuales

- **Identificar el *bound context*:**
 - ✓ El dominio “*Interacciones*” incluye reacciones, comentarios, calificaciones y seguir.
 - ✓ Separamos *reaction-service* de *comment-service*, *rate-service*, y *follow-service*.
- **Definir la entidad principal:**
 - ✓ Entidad: *Reaction*.
 - ✓ Atributos: *ID*, *user_id*, *target_id*, *target_type*, *created_at*.
- **Delimitar funcionalidad de microservicio:**
 - ✓ Reaccionar a una película o comentario.
 - ✓ Eliminar una reacción.
 - ✓ Visualizar reacciones.
- **Diagramar caso de uso:**

Usuario accede a la película → usuario reacciona a película → se guarda la reacción.

Usuario accede a la película → usuario elimina reacción de película → se guarda la reacción.

Usuario accede a comentario → usuario reacciona a comentario → se guarda la reacción.

Usuario accede a comentario → usuario elimina reacción de comentario → se guarda la reacción.
- **Diseñar endpoint inicial (borrador):**
 - ✓ *POST* /reaction
 - ✓ *GET* /reaction/{user_id}/{target_type}
 - ✓ *DELETE* /reaction
- **Validar con *Stakeholders*:**

Revisión con líder de producto para ajustar estados válidos:

 - ✓ Created
 - ✓ Eliminated
 - ✓ Updated

7. Follow: Servicio de Streaming

7.1. Fase 1: Análisis y Diseño Funcional del Microservicio

Microservicio: *follow-service*

Tareas Puntuales

- **Identificar el *bound context*:**
 - ✓ El dominio “*Interacciones*” incluye reacciones, comentarios, calificaciones y seguir.
 - ✓ Separamos *follow-service* de *comment-service*, *rate-service*, y *reaction-service*.
- **Definir la entidad principal:**
 - ✓ Entidad: *Follow*.
 - ✓ Atributos: *ID*, *follower_id*, *followed_id*, *created_at*.
- **Delimitar funcionalidad de microservicio:**
 - ✓ Seguir a un usuario.
 - ✓ Dejar de seguir a un usuario.
 - ✓ Ver a quienes sigue un usuario.
 - ✓ Ver los seguidores de un usuario.
- **Diagramar caso de uso:**

Usuario accede al perfil de otro → usuario sigue el perfil → se guarda → el usuario forma parte de los followers del usuario seguido.

Usuario accede a la lista de following → usuario accede a perfil de otro usuario → usuario deja de seguir el perfil → se guarda la interacción → el usuario se elimina de los followers del otro perfil

Usuario accede al perfil de otro usuario → usuario selecciona lista de followers → se muestra lista de perfiles.
- **Diseñar endpoint inicial (borrador):**
 - ✓ *POST* /follows
 - ✓ *GET* /profile/{*follower_id*}/following
 - ✓ *GET* /profile/{*follower_id*}/followers
 - ✓ *DELETE* /follows/{*follower_id*}/{*followed_id*}
- **Validar con *Stakeholders*:**

Revisión con líder de producto para ajustar estados válidos:

 - ✓ Created

- ✓ Eliminated
- ✓ Updated

8. Rate: Servicio de Streaming

8.1. Fase 1: Análisis y Diseño Funcional del Microservicio

Microservicio: *rate-service*

Tareas Puntuales

- **Identificar el *bound context*:**
 - ✓ El dominio “*Interacciones*” incluye reacciones, comentarios, calificaciones y seguir.
 - ✓ Separamos *rate-service* de *comment-service*, *follow-service*, y *reaction-service*.
- **Definir la entidad principal:**
 - ✓ Entidad: *Rate*.
 - ✓ Atributos: *ID*, *user_id*, *movie_id*, *score*, *created_at*.
- **Delimitar funcionalidad de microservicio:**
 - ✓ Calificar una película.
 - ✓ Ver la calificación de una película.
 - ✓ Actualizar la calificación de un usuario.
 - ✓ Obtener el promedio de calificaciones de una película.
- **Diagramar caso de uso:**

Usuario accede a una película → selecciona una puntuación → se guarda en el sistema → se actualiza el promedio general del contenido.

Usuario accede a película → actualiza una puntuación → se guarda en el sistema → se actualiza el promedio general del contenido..
- **Diseñar endpoint inicial (borrador):**
 - ✓ *POST* */rating/movies/{movie_id}*
 - ✓ *GET* */rating/movies/{movie_id}*
 - ✓ *GET* */rating/movies/{movie_id}/average*
 - ✓ *DELETE* */rating/movies/{movie_id}*
- **Validar con *Stakeholders*:**

Revisión con líder de producto para ajustar estados válidos:

 - ✓ Created
 - ✓ Eliminated
 - ✓ Updated