Taller de diseño OO

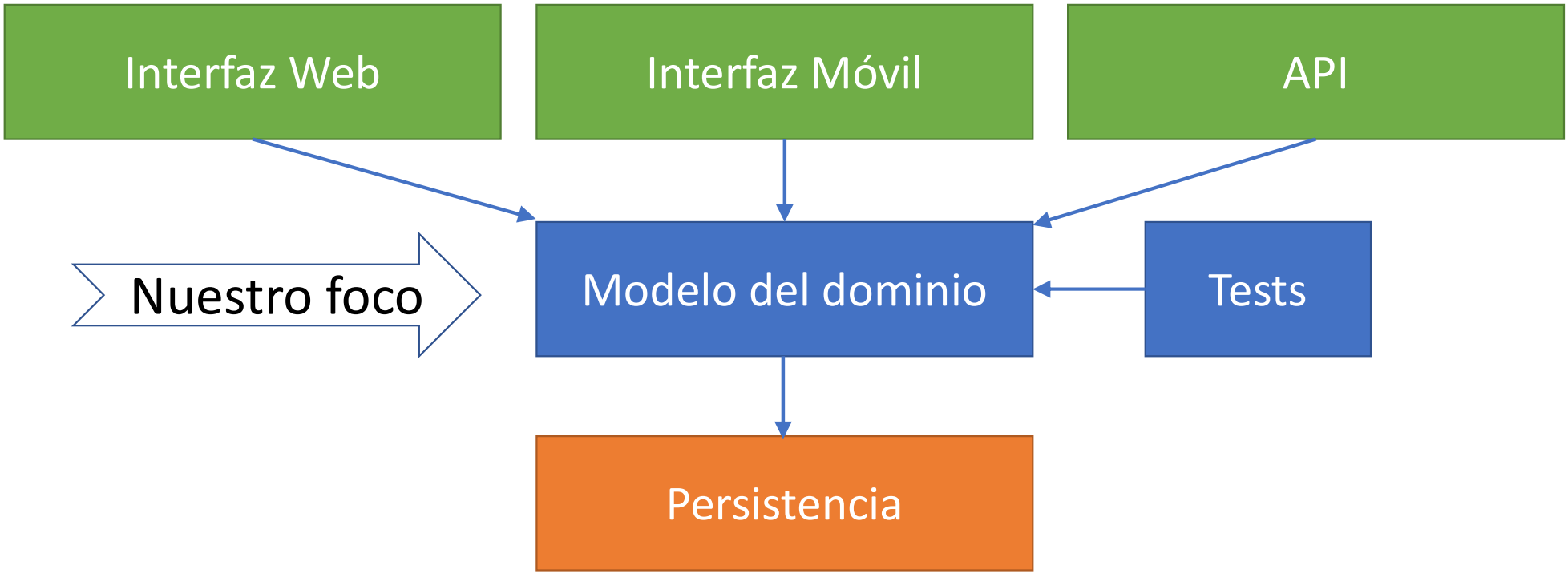
Dinámica del taller

- Vamos a partir de casos de uso breves, bien organizados (no siempre pasa)
  - Si están conectados desde la computadora, pueden tener a mano la especificación
- Alternaremos entre diagramas de clase, diagramas de secuencia y Pharo para tomar y plasmar nuestras decisiones de diseño
- Vamos a trabajar en dos iteraciones; una hoy con lo mas simple y otra la semana que viene agregando complejidad
- Vamos a simular ser un pequeño grupo de diseñogramadores discutiendo las decisiones que tomamos

Herramientas para tener a mano

- La especificación del sistema (los casos de uso)
- Documento de criterios y heurísticas de diseño
- Heurísticas de asignación de responsabilidades (HAR)
- Editor de UML (yo voy a usar StarUML, evaluación)
- Recomendaciones de sintaxis UML (D. de Clase y Secuencia)

Contexto…


Interfaz Web
Interfaz Móvil
API
Nuestro foco
Modelo del dominio
Tests
Persistencia

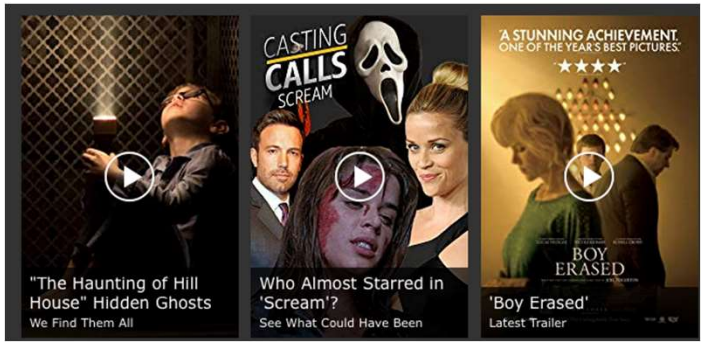
De la especificación al diseño y al código

Nuestra tarea como diseñadores/programadores orientados a objetos, por lo general, implicará:

- Entender qué debe hacer el sistema (los casos de uso)
- Identificar potenciales objetos, propiedades, relaciones
- Por cada caso de uso, determinar cuales son los objetos involucrados y que debe hacer cada uno (asignar responsabilidades).
- Implementar las responsabilidades individuales de los objetos y escribir los tests que aseguran que el objeto hace lo que se supone que haga

Con rueditas al principio

- Producir buenos diseños es una capacidad que se adquiere y mejora con la práctica
- A medida que ganamos práctica, nuestros procesos de diseño se vuelven mas “intuitivos” y basados en experiencias previas
- Al iniciar, es útil tener algunas ayudas mecánicas, que nos sirvan de andamio o rueditas de bici, y nos ayuden a avanzar – hoy vamos a seguir algunas de esas
- Cada uno decide cuando está listo para abandonarlas



Cinéfiloos

Un sitio para encontrar películas que nos puedan gustar, y gente con gustos como los nuestros

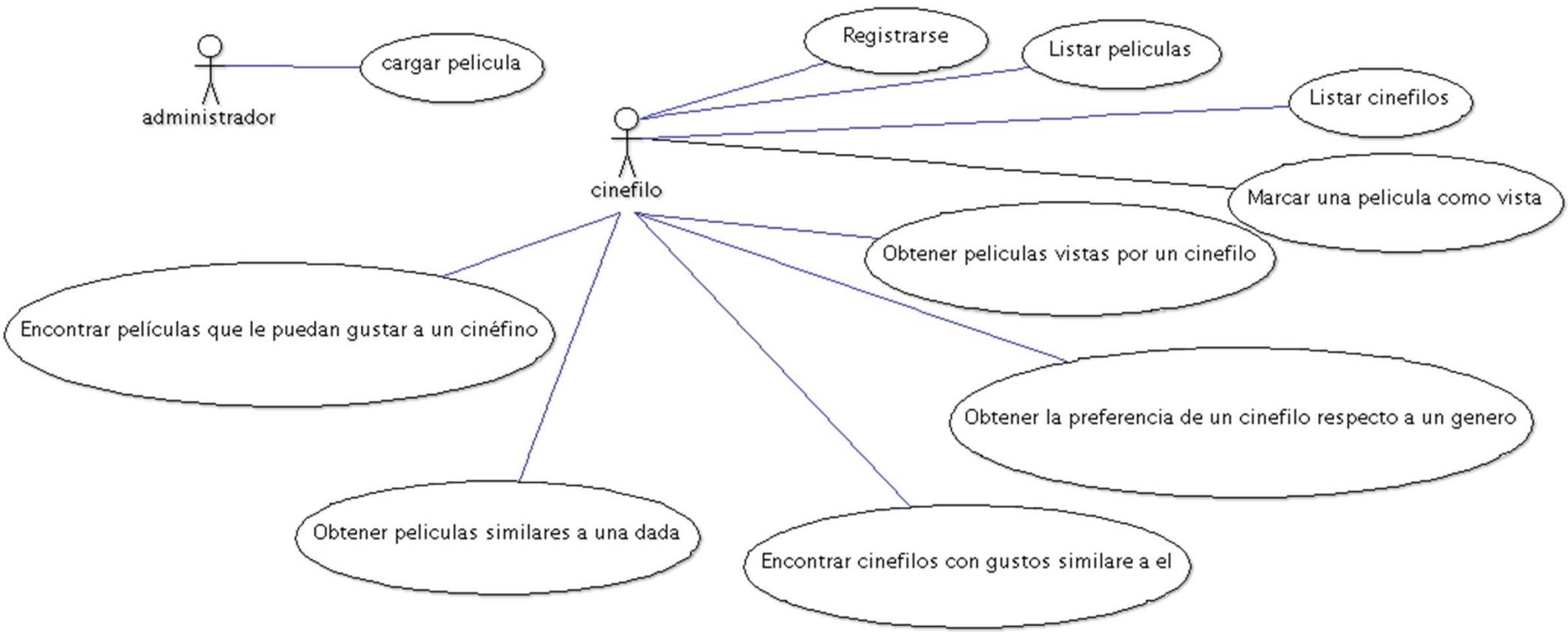
Cinéfilo: El **cinéfilo**, en términos generales, es una persona que tiene un gusto especial por el cine.

De la especificación al diseño y el código

Nuestra tarea como diseñadores/programadores orientados a objetos, por lo general, implicará:

- **Entender qué debe hacer el sistema (los casos de uso)**
- Identificar potenciales objetos, propiedades, relaciones
- Por cada caso de uso, determinar cuales son los objetos involucrados y que debe hacer cada uno (asignar responsabilidades).
- Implementar las responsabilidades individuales de los objetos y escribir los tests que aseguran que el objeto hace lo que se supone que haga

Cinefiloos es un sitio para llevar registro de las películas que uno ha visto, encontrar a cinéfilos con gustos parecidos, y obtener sugerencias. El sistema se apoya en IMDB, un sitio en el que ya están registradas la mayoría de las películas. Eso hace innecesario cargar toda la información. A continuación se describen los casos de uso en formato breve.


administrador
cargar pelicula
cinefilo
Registrarse
Listar peliculas
Listar cinefilos
Marcar una pelicula como vista
Obtener peliculas vistas por un cinefilo
Encontrar películas que le puedan gustar a un cinéfino
Obtener la preferencia de un cinefilo respecto a un genero
Obtener peliculas similares a una dada
Encontrar cinefilos con gustos similare a el

Cargar película: Se ingresa título, la URL de la película en IMDB, y la URL de la imagen de portada de la película (también tomada de IMDB). Adicionalmente indica el "perfil de género" de la película. Cinefiloos (el sistema) registra la película.

El perfil de género indica en una escala de 0 a 9, cuánto de cada género tiene una película. Los géneros que se consideran son:

horror, action, romance, suspense, comedy, y sci-fi.

Si no se indica perfil de género, tomará 0 para todos los géneros (sin género definido).

Registrar un cinéfilos: el cinéfilo ingresa su nombre completo, y su email. El sistema registra al cinéfilo y lo retorna.

Obtener películas: El sistema retorna la lista de películas.

Obtener cinéfilos: El sistema retorna la lista de cinéfilos.

Marcar una película como vista: Se indica un cinéfilo y una película. El sistema registra la película, como vista por el cinéfilo.

Obtener películas vistas por un cinéfilo: Dado un cinéfilo, el sistema retorna la lista de películas vistas por el cinéfilo.

Encontrar películas similares: Retornar todas las películas que son similares a una que se indique. Retorna todas las películas cuyo "índice de similitud" a la película indicada es menor a 6 (menor índice implica, más parecidas). La lista no está ordenada. La película indicada también está en la lista.

	horror	action	romance	suspense	comedy	sci-fi
Película 1	9	0	0	5	0	0
Película 2	0	0	0	9	0	7

Índice de similitud (mejor significa más parecidos):

- por cada género calculo la diferencia entre las dos
- sumo los valores absolutos

índice: abs(9-0)+0+0+abs(5-9)+0+abs(0-7) = 20

Obtener la preferencia de genero de cine de un cinéfilo: Dado un cinéfilo retorna su preferencia de genero de cine (que tiene un número de 0 a 9 por cada género, al igual que las películas).

La preferencia de género de cine de un cinéfilo se calcula a partir de las películas que vio (asumimos que le gustaron).

Si no vió ninguna, se toma 4.5 para todos los géneros (ni muy muy, ni tan tan). Luego, toma el promedio entre esa base y todas las películas que vea.

	horror	action	romance	suspense	comedy	sci-fi
Base	4.5	4.5	4.5	4.5	4.5	4.5
Película 1	9	0	0	5	0	0
Película 2	6	0	0	9	0	2
Promedio	6.16	1.5	1.5	6.1	1.5	2.1

Encontrar cinéfilos con gustos similares: Dado un cinéfilo, encontrar todos aquellos que tienen preferencias de género de cine parecidas. Para calcular la similitud entre dos cinéfilos, se toman las preferencias de género de cine de cada uno.

Retorna todos los cinéfilos cuyo "índice de similitud" al indicado es menor a 5 (menor índice significa más parecido). La lista no está ordenada. El cinéfilo está en la lista también.

El cálculo es idéntico al de similitud entre películas

Encontrar películas que pueden gustar, no vistas: Dado un cinéfilo, retorna todas las películas cuyo "índice de similitud" a la preferencia del cinéfilo es menor a 7, y que el cinéfilo no vio. La lista no está ordenada.

El cálculo es idéntico al de similitud entre películas.

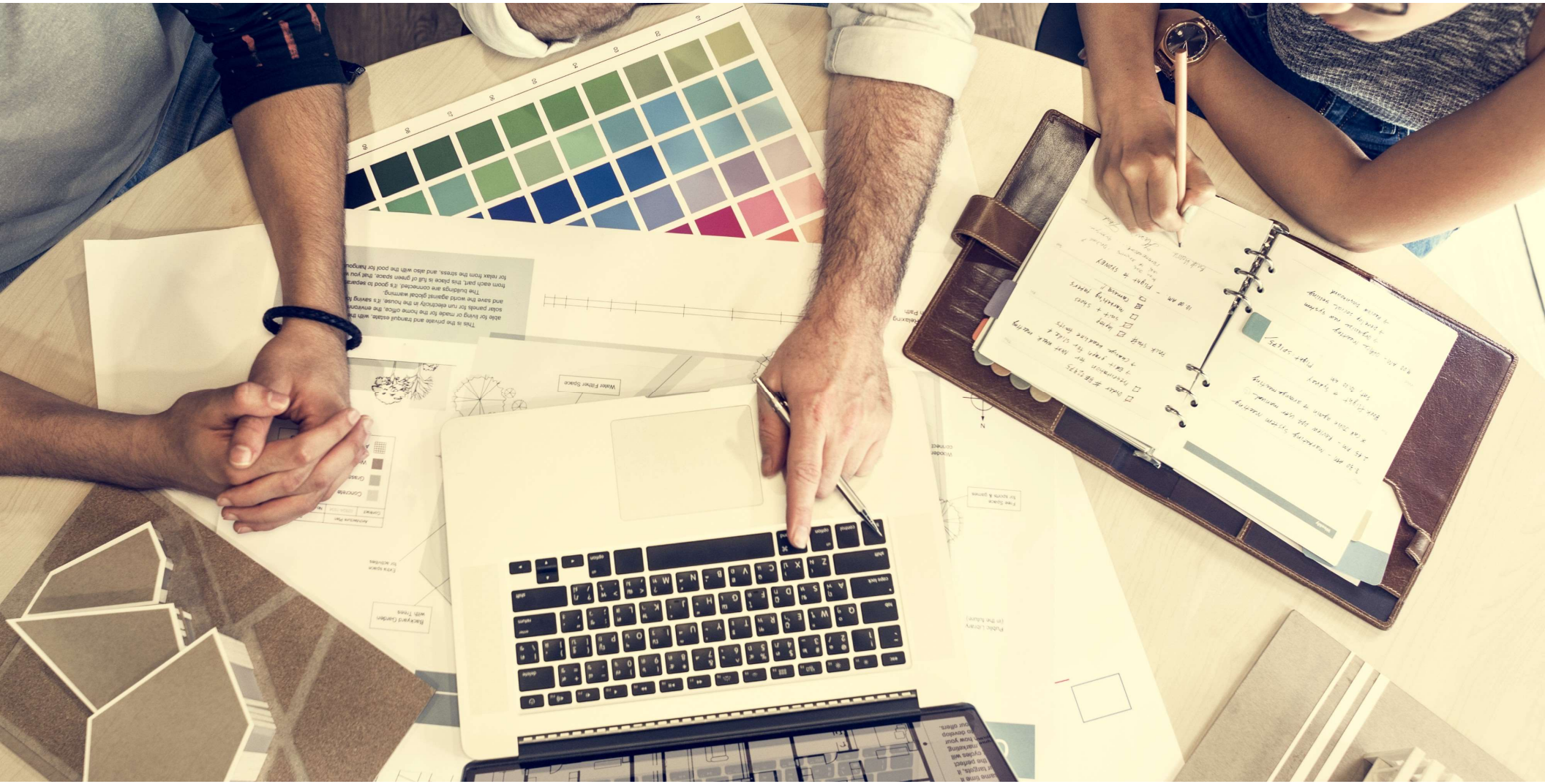
De la especificación al diseño y el código (iteración 1 – sin Genero)

Nuestra tarea como diseñadores/programadores orientados a objetos, por lo general, implicará:

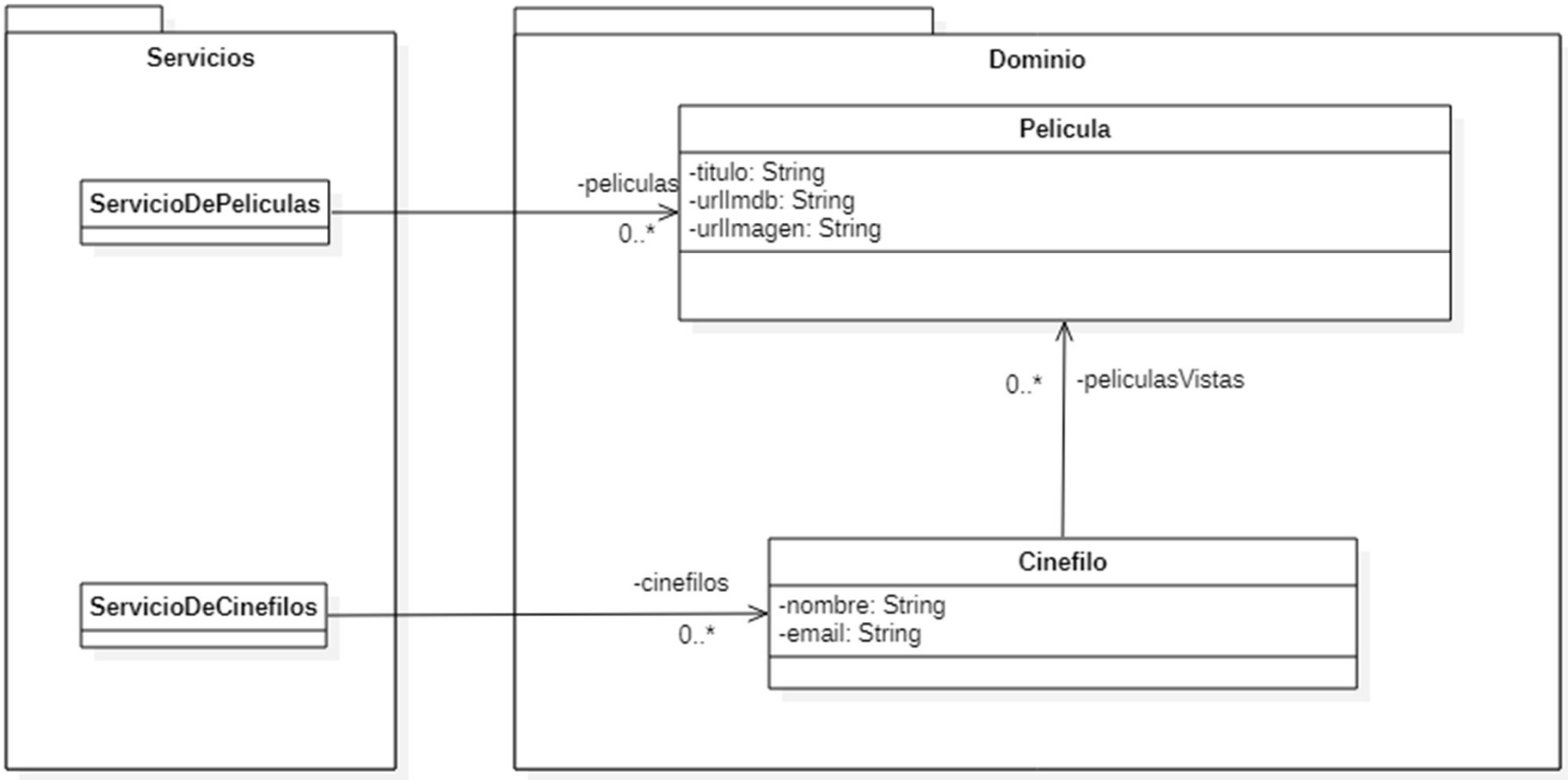
- Entender qué debe hacer el sistema (los casos de uso)
- **Identificar potenciales objetos, propiedades, relaciones**
- Por cada caso de uso, determinar cuales son los objetos involucrados y que debe hacer cada uno (asignar responsabilidades).
- Implementar las responsabilidades individuales de los objetos y escribir los tests que aseguran que el objeto hace lo que se supone que haga

Identificar potenciales objetos, propiedades, relaciones

- En la especificación (los casos de uso) intentamos identificar los conceptos que hacen al dominio (buscamos sustantivos)
  - De esos, algunos serán potenciales clases (comenzamos con las más obvias)
  - Otros serán atributos de esas clases (nuevamente, comenzamos con los mas obvios)
  - Otros serán relaciones
- Presto atención a posibles sinónimos
- Describo el dominio en voz alta y me escucho (para intentar identificar conceptos que no estén escritos, o encontrar términos más descriptivos).
- Itero hasta que ya no me queden en la lista de cosas, elementos que esté seguro como agregar
- Obtengo un modelo conceptual (clases candidatas)



Cópienlo así lo tenemos a mano



De la especificación al diseño y el código (iteración 1 – sin Genero)

Nuestra tarea como diseñadores/programadores orientados a objetos, por lo general, implicará:

- Entender qué debe hacer el sistema (los casos de uso)
- Identificar potenciales objetos, propiedades, relaciones
- **Por cada caso de uso, determinar cuales son los objetos involucrados y que debe hacer cada uno (asignar responsabilidades).**
- Implementar las responsabilidades individuales de los objetos y escribir los tests que aseguran que el objeto hace lo que se supone que haga

Asignar responsabilidades

- ¿Pre-condiciones?
  - ¿qué objetos deben existir?
  - ¿en que estado tienen que estar, qué objetos?
- ¿Post-condiciones?
  - ¿qué objetos nuevos aparecen?
  - ¿a que objetos les cambia su estado / atributos?
  - ¿qué relaciones cambian?
- Diferencia entre pre y post condiciones → ¿qué objeto hace que parte?
  - Para esto me ayuda construir diagramas de secuencia, comenzando con los objetos que existen (pre-condiciones)
  - Para asignar bien responsabilidades me sirven las heurísticas (vamos a recordarlas) y la práctica
  - No hay respuestas absolutas… lo importante en este punto es conocer los criterios

Heurística: Identificar creadores

- Asignar a la clase B la responsabilidad de crear una instancia de la clase A si:
  - B contiene objetos A (agregación, composición).
  - B registra instancias de A.
  - B tiene los datos para inicializar objetos A.
  - B usa a objetos A en forma exclusiva.

Heurística: Identificar expertos en Información

- Asignar una responsabilidad al experto en información (la clase que tiene la información necesaria para realizar la responsabilidad).
- Expresa la intuición de que los objetos hacen cosas relacionadas con la información que tienen.
- Para cumplir con su responsabilidad, un objeto puede requerir de información que se encuentra dispersa en diferentes clases expertas “parciales” información

Heurística: Aprovecha el polimorfismo

- cuando el comportamiento varía según el tipo, modele los distintos tipos como clases (si es que no existen) asigne la responsabilidad a la clases para las que varía el comportamiento. Nos permite sustituir objetos que tienen idéntica interfaz.

Heurística: No hables con extraños

- Evite diseñar objetos que recorren largos caminos de estructura y envían mensajes (hablan) a objetos distantes o indirectos (extraños). Dentro de un método sólo pueden enviarse mensajes a objetos conocidos:
  - this
  - un parámetro del método
  - un objeto que esté directamente asociado a this
  - un miembro de una colección que sea atributo de this
  - un objeto creado dentro del método
  - Los demás objetos son extraños.

Heurística: Bajo acoplamiento

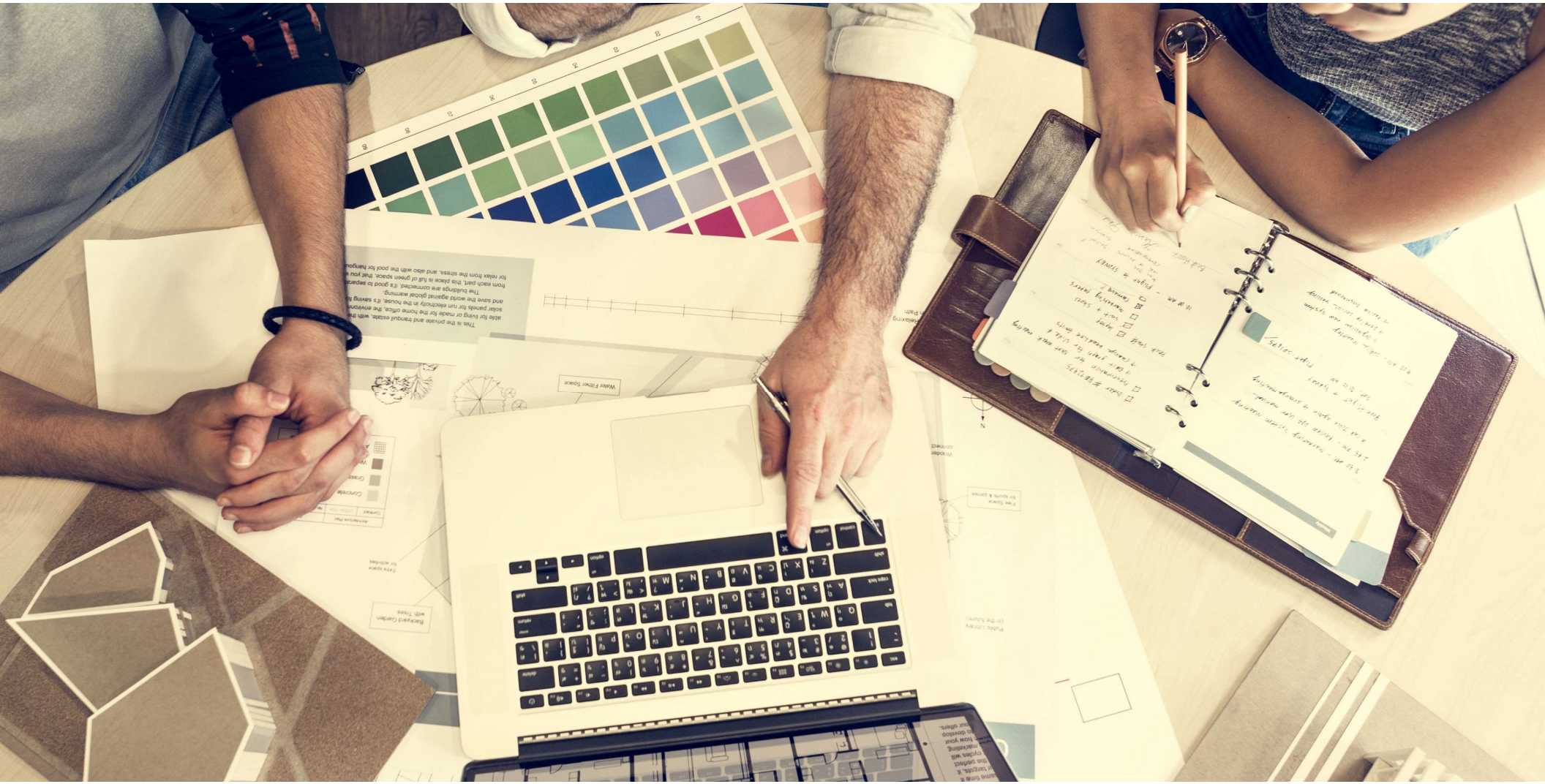
- Asignar responsabilidades de manera que el acoplamiento permanezca lo más bajo posible. El acoplamiento es una medida de dependencia de un objeto con otros. Es bajo si mantiene pocas relaciones con otros objetos. El alto acoplamiento dificulta el entendimiento y complica la propagación de cambios en el diseño.
- No se puede considerar de manera aislada a otras heurísticas, sino que debe incluirse como principio de diseño que influye en la elección de la asignación de responsabilidad.

Heurística: Alta cohesión

- Asignar responsabilidades de manera que la cohesión permanezca lo más fuerte posible. La cohesión es una medida de la **fuerza con la que se relacionan las responsabilidades de un objeto**, y la **cantidad** de ellas. Resulta en clases más fáciles de mantener, entender y reutilizar.
- El nivel de cohesión no se puede considerar de manera aislada a otras responsabilidades y otras heurísticas, como Experto y Bajo Acoplamiento.

Heurística: Delegar, delegar, delegar

- Un objeto no hace nada que pueda delegar a otro que conoce
- Siempre que vea que un objeto pide cosas a otro para hacer algo me pregunto: no podré delegárselo a el (no será él el “experto”)



Registrar un cinéfilos: el cinéfilo ingresa su nombre completo, y su email. El sistema registra al cinéfilo y lo retorna.

¿Pre-condiciones?

¿Post-condiciones?

¿Qué hay que hacer?

Registrar un cinéfilos: el cinéfilo ingresa su nombre completo, y su email. El sistema registra al cinéfilo y lo retorna.

Pre-condiciones

- La colección de cinéfilos existe.

Post-condiciones

- Hay una nueva instancia de Cinefilo
- Se establece nombre e email para el cinéfilo
- El cinéfilo no se relaciona con ninguna película.
- El cinéfilo está en la colección de cinéfilos
- Se devolvió el cinéfilo

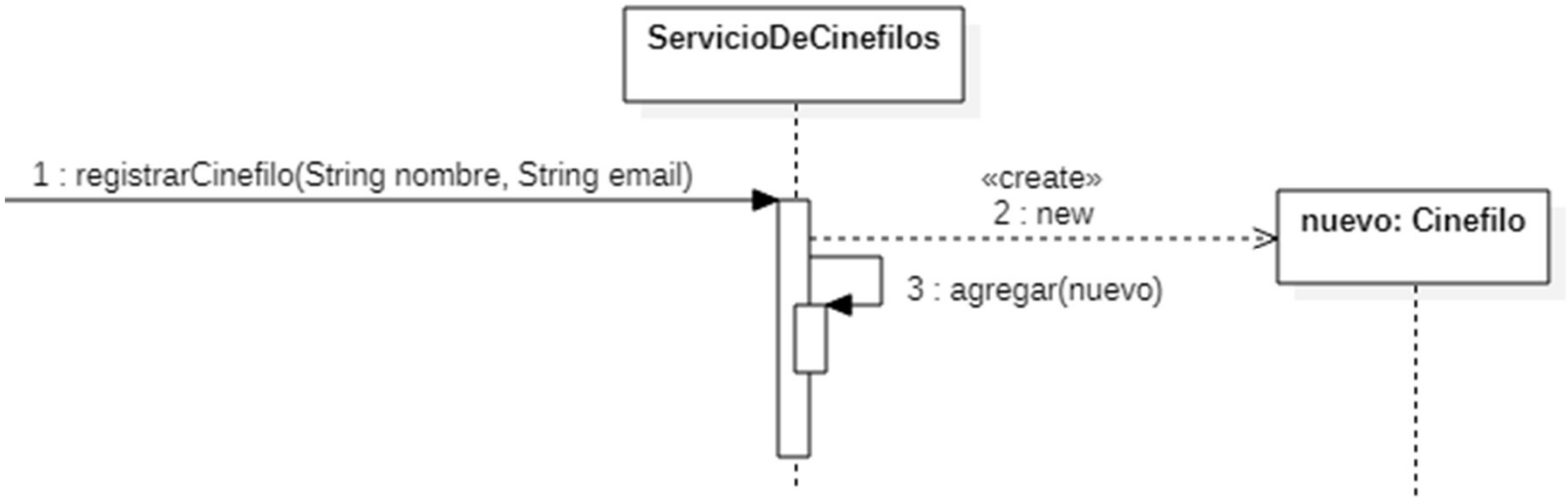
Qué hay que hacer

- Crear e inicializar el cinéfilo
- Agregar el cinéfilo a la colección
- Devolver el cinéfilo

Registrar un cinéfilos: el cinéfilo ingresa su nombre completo, y su email. El sistema registra al cinéfilo y lo retorna.

Qué hay que hacer

- Crear e inicializar el cinéfilo
- Agregar el cinéfilo a la colección
- Devolver el cinéfilo



Cargar película: Se ingresa título, la URL de la película en IMDB, y la URL de la imagen de portada de la película (también tomada de IMDB). Cinefiloos (el sistema) registra la película.

¿Pre-condiciones?

¿Post-condiciones?

¿Qué hay que hacer?

Cargar película: Se ingresa título, la URL de la película en IMDB, y la URL de la imagen de portada de la película (también tomada de IMDB). Cinefiloos (el sistema) registra la película.

Pre-condiciones

- La colección de películas existe.

Post-condiciones

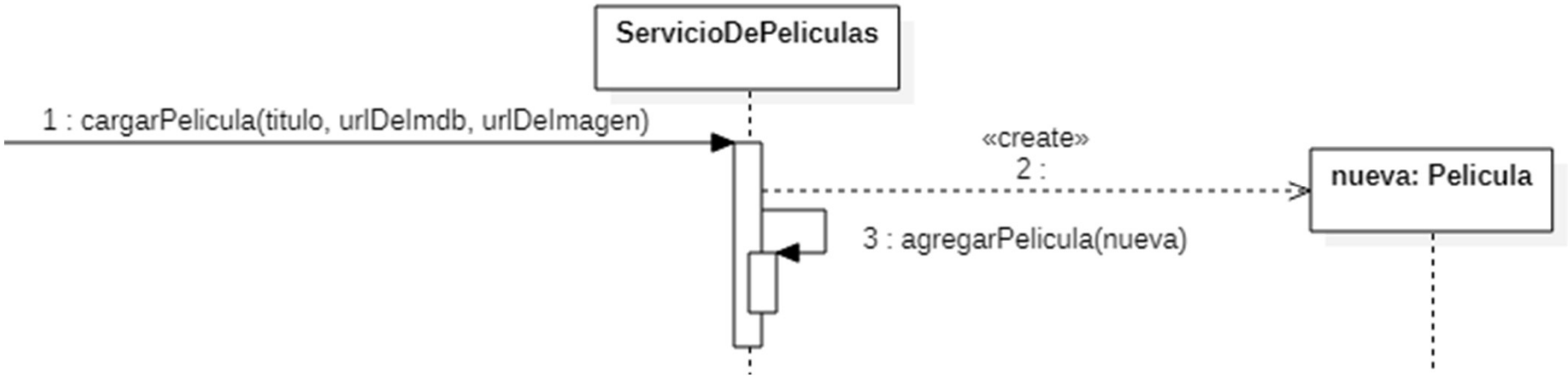
- Hay una nueva instancia de Pelicula
- Se establece el título, imdbUrl y imageUrl para la película
- La película está en la colección de películas

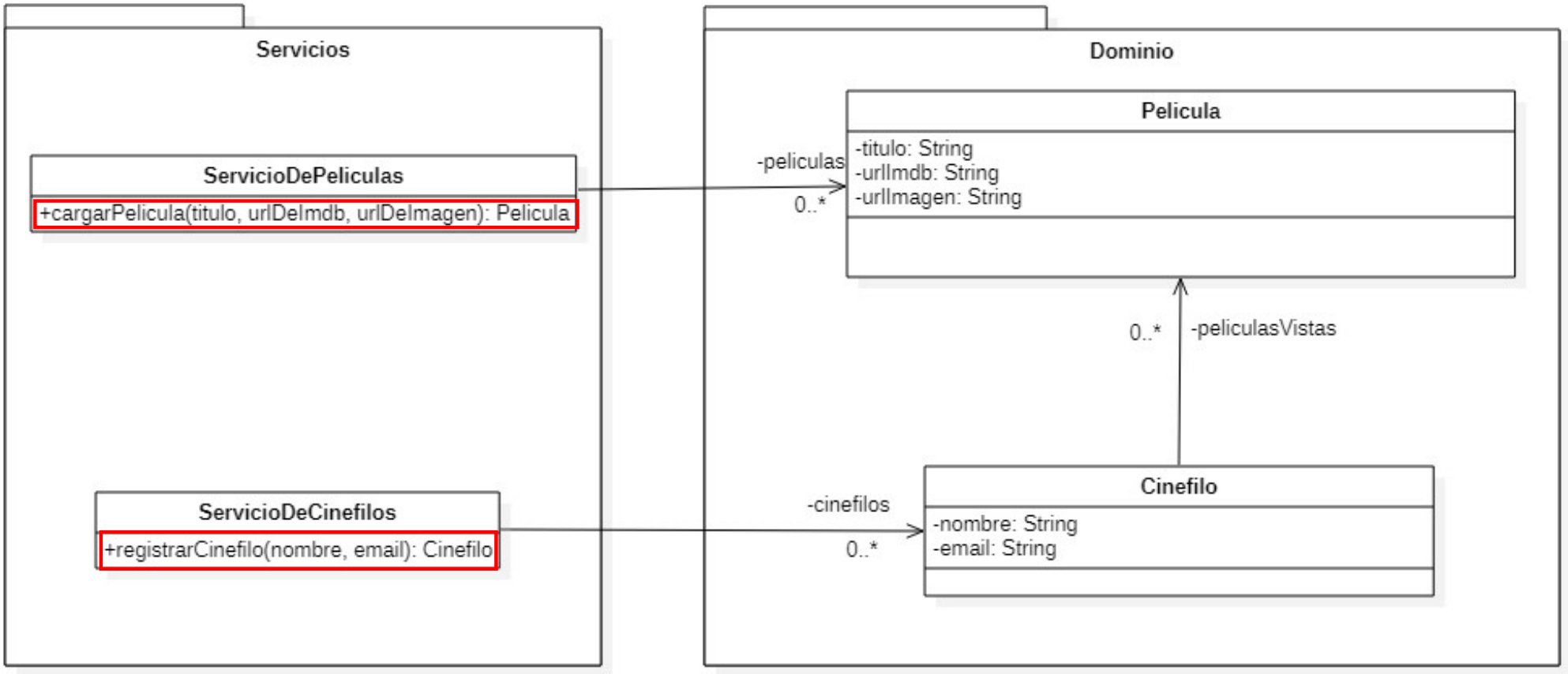
Qué hay que hacer

- Crear e inicializar la película
- Agregar la película a la colección
- Devolver la película

Cargar película: Se ingresa título, la URL de la película en IMDB, y la URL de la imagen de portada de la película (también tomada de IMDB). Cinefiloos (el sistema) registra la película.

- Crear e inicializar la película
- Agregar la película a la colección
- Devolver la película




Servicios
ServicioDePeliculas
+cargarPelicula(titulo, urlDeImdb, urlDeImagen): Pelicula
ServicioDeCinefilos
+registrarCinefilo(nombre, email): Cinefilo
Dominio
Pelicula
-titulo: String
-urlImdb: String
-urlImagen: String
-peliculas
0..*
0..*
-peliculasVistas
Cinefilo
-nombre: String
-email: String
-cinefilos
0..*

Obtener películas: El sistema retorna la lista de películas.

¿Pre-condiciones?

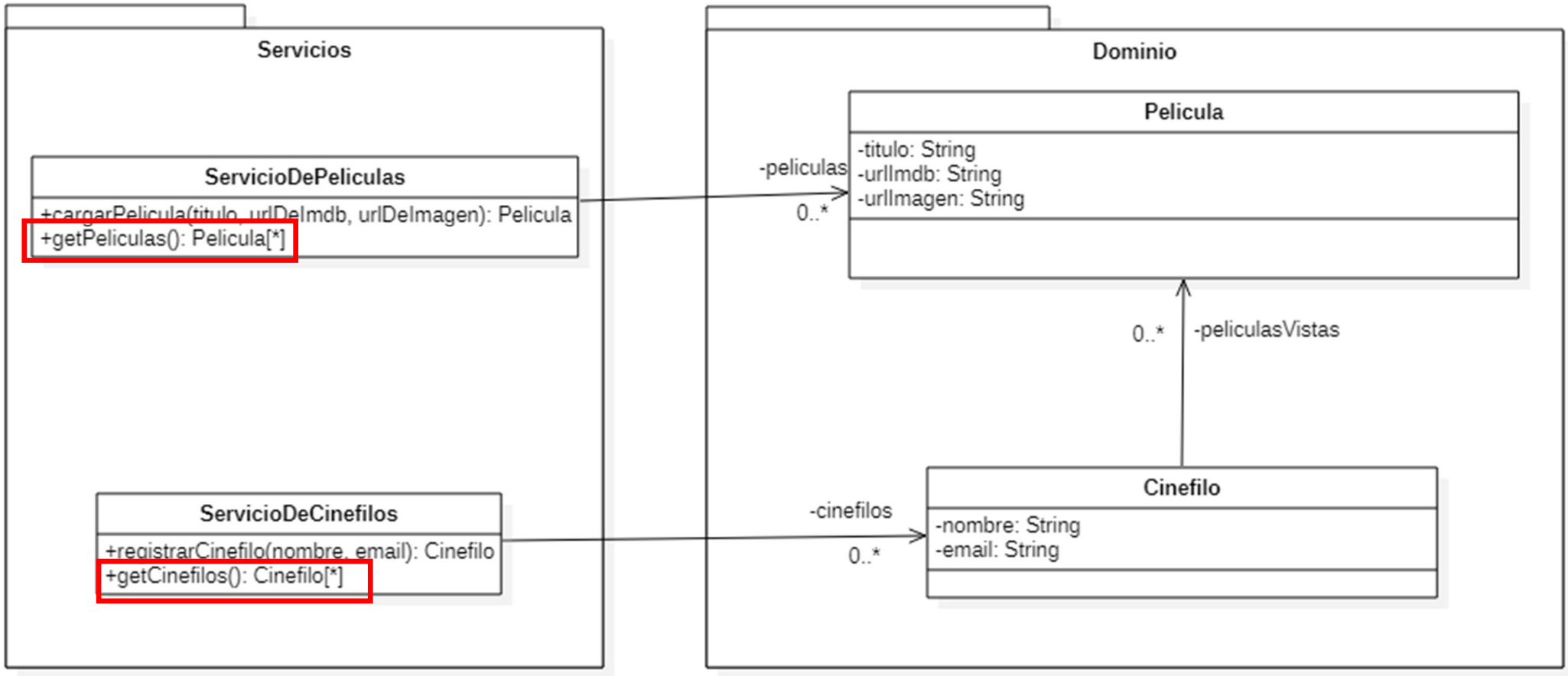
- La colección de películas existe (puede estar vacía o tener películas)

¿Post-condiciones?

- Nada cambia, solo se devuelve la colección.

¿Qué hay que hacer?

Obtener cinéfilos: El sistema retorna la lista de cinéfilos (sería similar)


Servicios
ServicioDePeliculas
+cargarPelicula(titulo, urlDeImdb, urlDeImagen): Pelicula
+getPeliculas(): Pelicula[*]
ServicioDeCinefilos
+registrarCinefilo(nombre, email): Cinefilo
+getCinefilos(): Cinefilo[*]
Dominio
Pelicula
-titulo: String
-urlImdb: String
-urlImagen: String
-peliculas
0..*
0..*
-peliculasVistas
Cinefilo
-nombre: String
-email: String
-cinefilos
0..*

Marcar una película como vista: Se indica un cinéfilo y una película. El sistema registra la película, como vista por el cinéfilo.

¿Pre-condiciones?

- El cinéfilo existe y está en la colección de cinéfilos
- La película existe y está en la colección de películas
- El cinéfilo no tiene a la película entre la lista de las que vió

¿Post-condiciones?

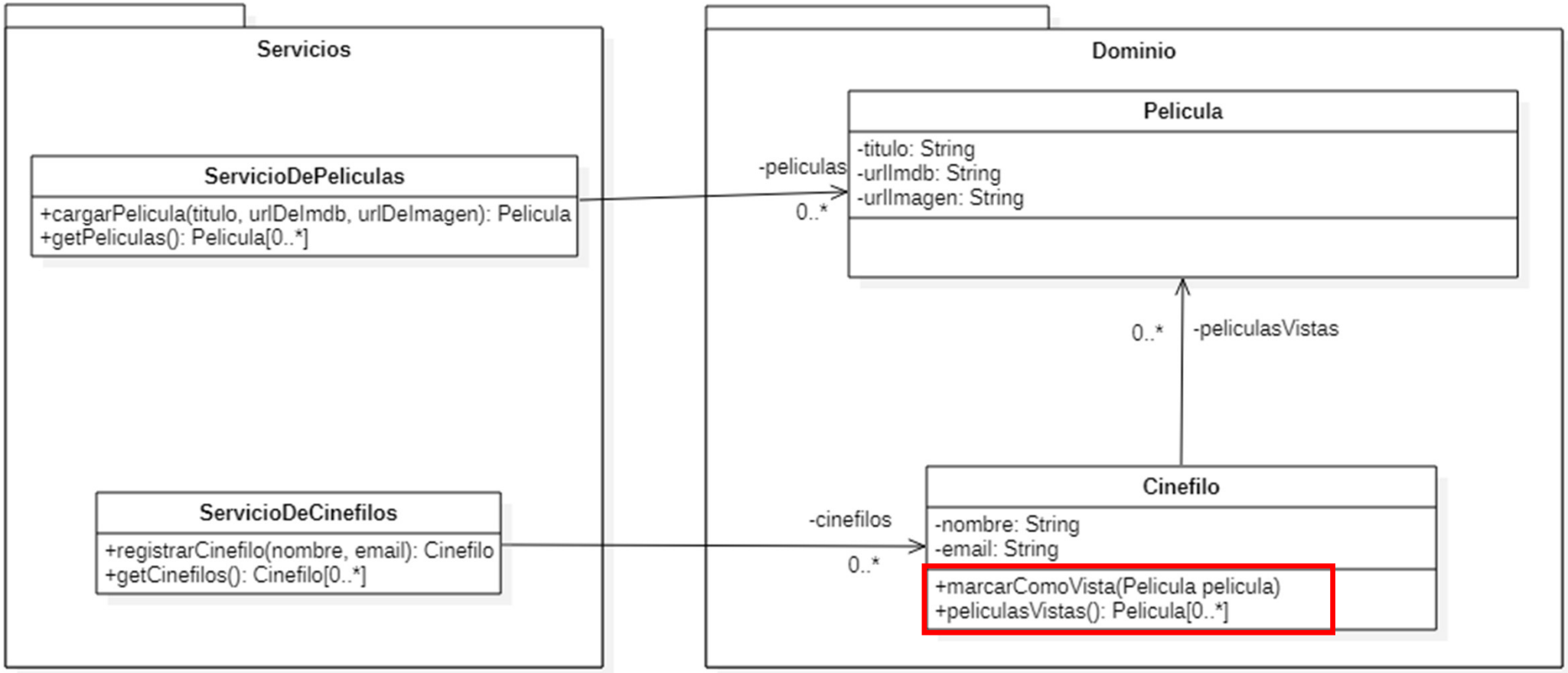
- El cinéfilo tiene a la película entre la lista de las que vió

¿Qué hay que hacer?

- Agregar la película a la lista de las que vió el cinéfilo

¿Obtener películas vistas por un cinéfilo?: Dado un cinéfilo, el sistema retorna la lista de películas vistas por el cinéfilo.

Tenemos el cinéfilo y la película (los objetos), ¿necesitamos involucrar a los servicios?


Servicios
ServicioDePeliculas
+cargarPelicula(titulo, urlDeImdb, urlDeImagen): Pelicula
+getPeliculas(): Pelicula[0..*]
ServicioDeCinefilos
+registrarCinefilo(nombre, email): Cinefilo
+getCinefilos(): Cinefilo[0..*]
Dominio
Pelicula
-titulo: String
-urlImdb: String
-urlImagen: String
-peliculas
0..*
0..*
-peliculasVistas
Cinefilo
-nombre: String
-email: String
+marcarComoVista(Pelicula pelicula)
+peliculasVistas(): Pelicula[0..*]
-cinefilos
0..*

Obtener cinéfilos que vieron una película: Dada una película, el sistema retorna la lista de cinéfilos que la vieron**.**

¿Pre-condiciones?

- La película existe y está en la colección de películas
- Dos cinéfilos existen y están en la colección de cinéfilos
- Los dos cinéfilos tienen a la película entre la lista de películas que vieron

¿Post-condiciones?

¿Qué hay que hacer?

- ¿Como nos afectan las decisiones de diseño que ya tomamos? (p.e., el cinéfilo conoce las películas que vió y no al revés)
- ¿Qué podríamos hacer diferente? ¿Se justifica?

Obtener cinéfilos que vieron una película: Dada una película, el sistema retorna la lista de cinéfilos que la vieron**.**