

Tema 1: Exemple senzill del procés de desenvolupament del software

Anna Puig

Enginyeria Informàtica

Facultat de Matemàtiques i Informàtica,

Universitat de Barcelona

Curs 2021/2022



UNIVERSITAT DE
BARCELONA

1.3 Exemple senzill

Què veurem?

- Requeriments
- Casos d'ús
- User stories: criteris d'acceptació
- Model de domini
- Model de disseny
 - Diagrames de classes
 - Diagrames de interacció

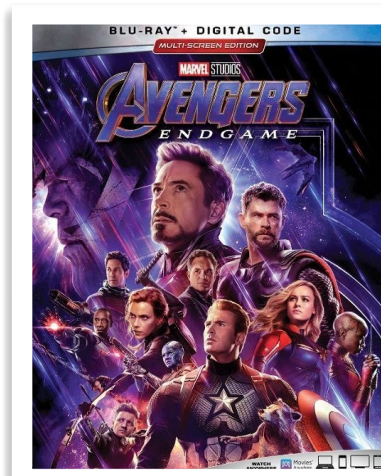
NOTA: En els següents apartats es veu una **breu** introducció a cadascun dels punts.

1.3 Exemple senzill

Escenari: Videostore: es vol una app que estigui activa 24hores del dia tot el món

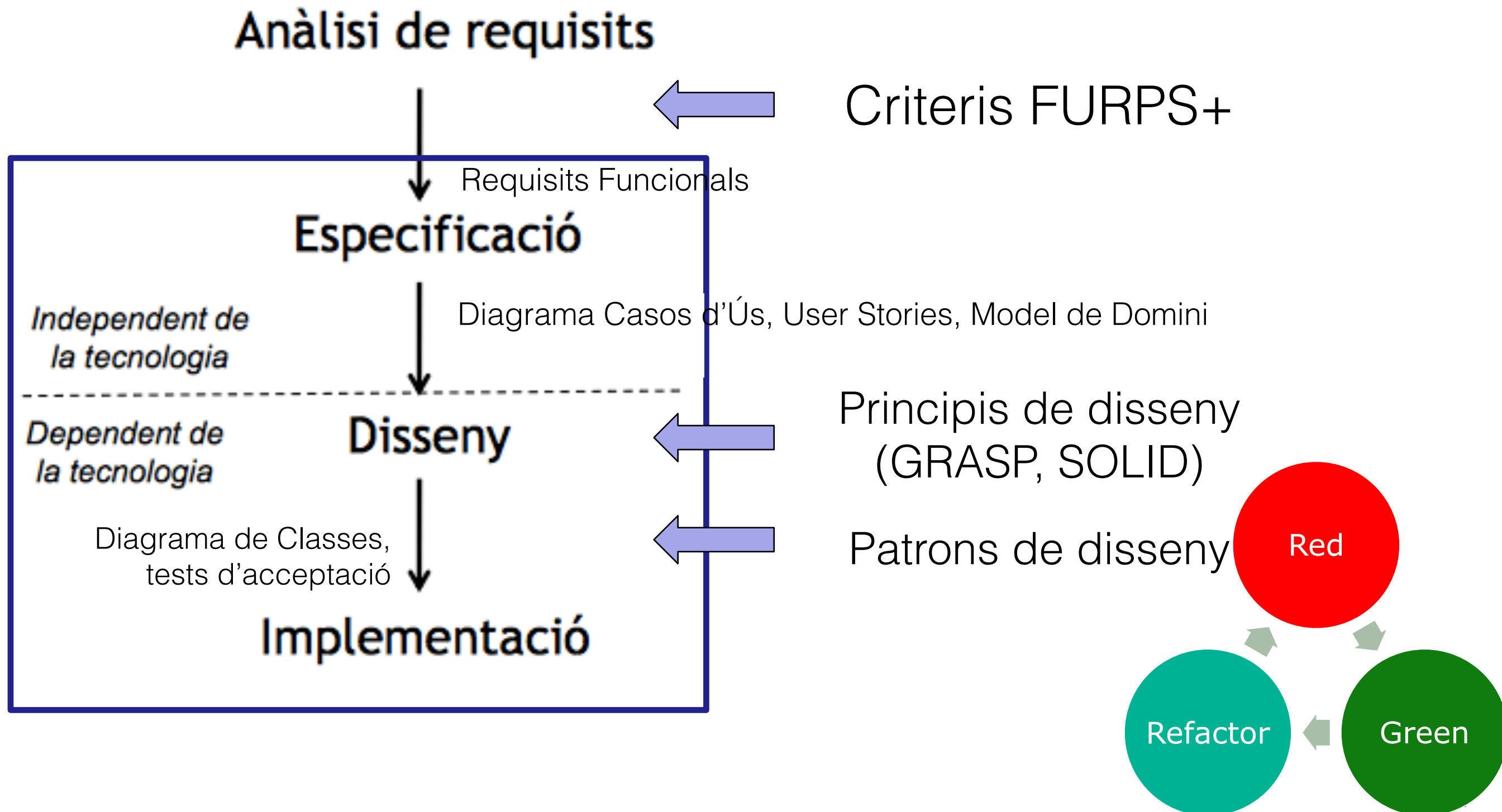


- El preu del lloguer es calcula segons el tipus de pel·lícula, i el temps de lloguer.
- Hi han tres tipus de vídeos:
 - Normals
 - Infantils
 - Novetats
- El clients guanyen punts si lloguen pel·lícules de tipus “Novetats” (o New Releases)



1.1. Procés de desenvolupament de software

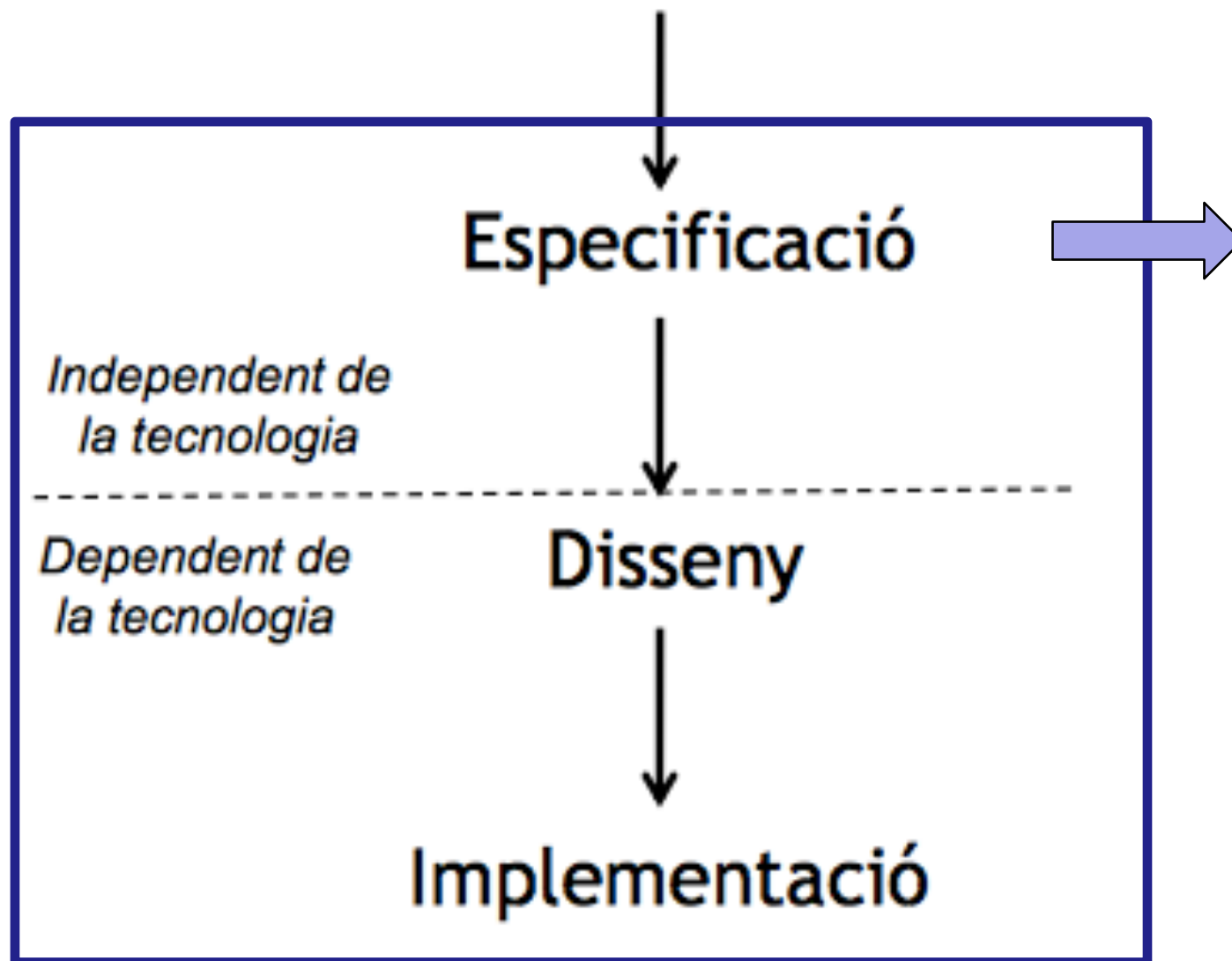
Procés sistemàtic:



1.2 Etapes de développement de software

Processus systématique:

Anàlisi de requisits



Què ha de fer el sistema?

- Especificació de la interacció amb l'usuari (**Casos d'ús, user stories**)
- Especificació de dades (**Esquema conceptual de les dades: Model de Domini**)
- Especificació de processos (Esquema del comportament)

Casos d'ús

- Els casos d'ús són històries sobre com usar un sistema per satisfer uns objectius
- Descriuen el que fa el sistema des del punt de vista d'un observador extern

Emfatitzen el *què* enlloc del *com*

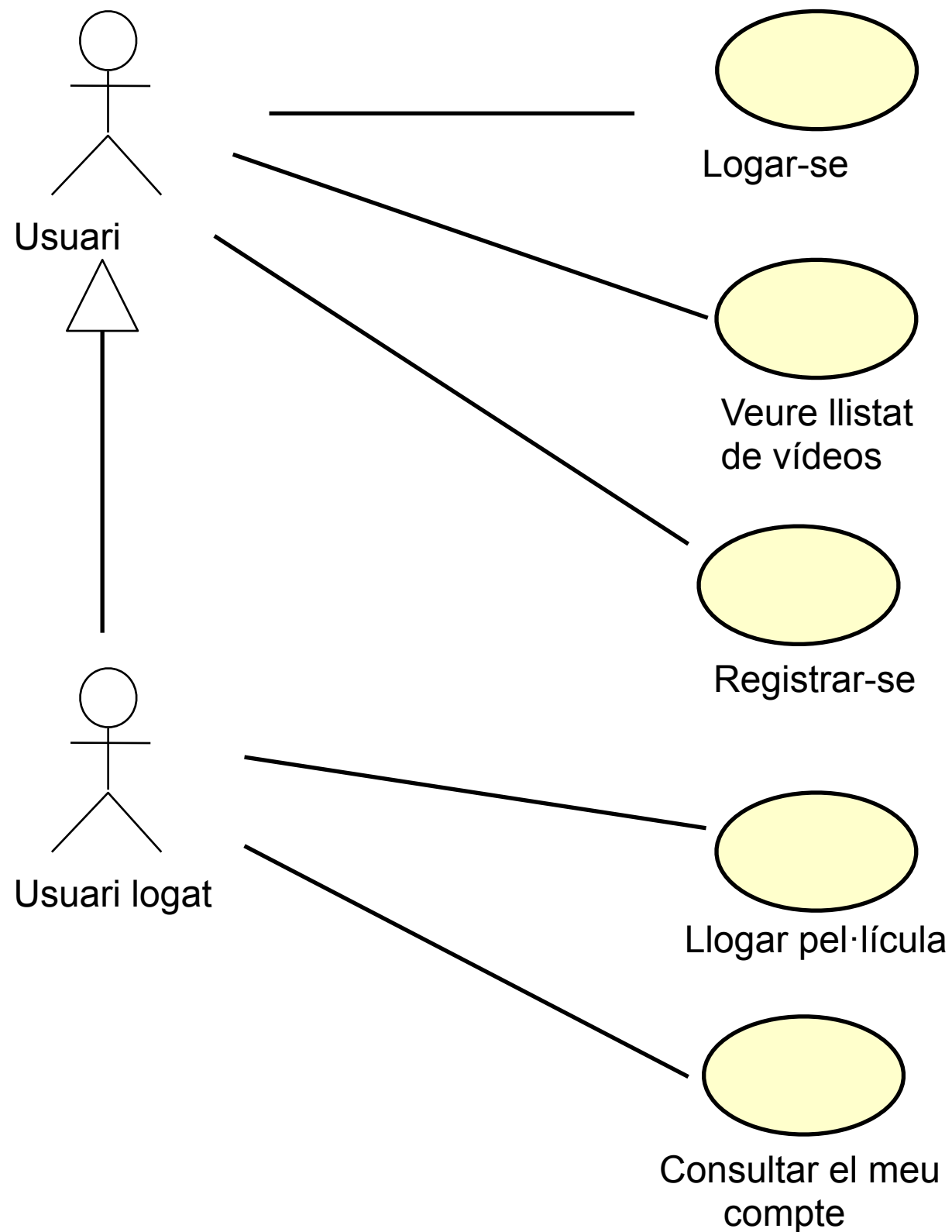
Plantegen **diferents escenaris en els Casos D'Ús**, el que passa quan el *client interactua amb el sistema*.
Proporcionen un resum per a un objectiu

Cas d'ús 1: Login

Cas d'ús 2: Llogar pel·lícula

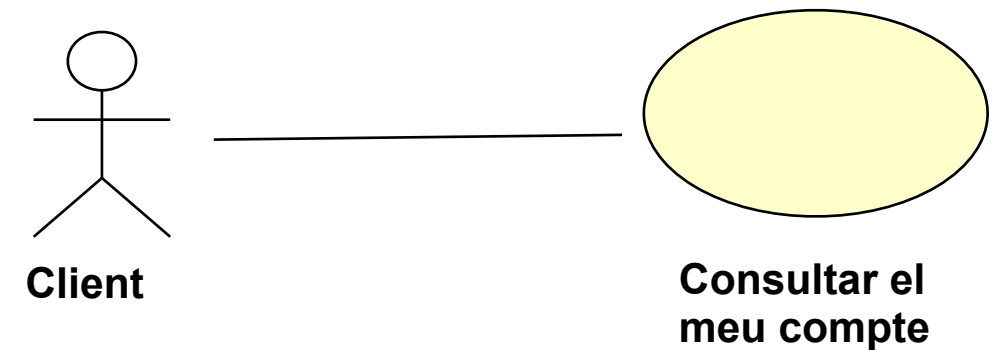
Cas d'ús 3: ConsultarElMeuCompte

Exemple Diagrama de Casos d'Ús



Exemple cas d'ús

Consultar el meu compte



1. Un **client** selecciona consultar el seu compte des del seu Perfil
2. El **sistema** mostra el deute associat al compte i els punts guanyats fins el moment
3. El **client** demana el detall del deute
4. El **sistema** mostra les pel·lícules llogades i els preus de cadascuna d'elles

Extensions:

- 3.a El **client** demana el detall dels punts guanyats
 1. El **sistema** mostra les pel·lícules llogades i els punts de cadascuna
- 4.a. No hi han pel·lícules llogades. El **sistema** mostra un missatge que diu "No hi han pel·lícules llogades"

User stories

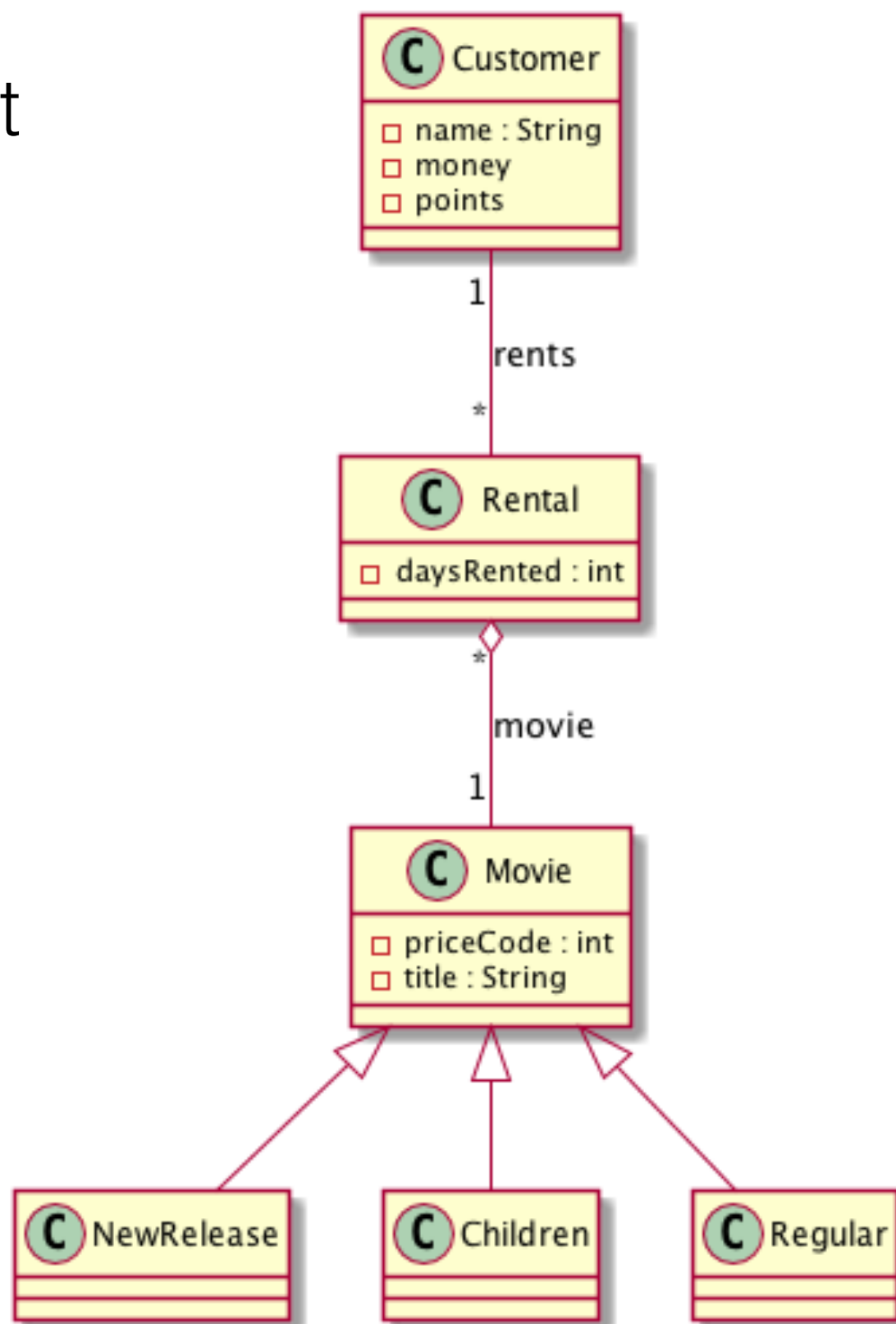
- **Com a** usuari **vull** enregistrar-me **per a** poder llogar pel·lícules
- **Com a** usuari **vull** consultar el preu d'un lloguer d'una pel **per a** decidir si vull/puc llogar
- **Com a** usuari **vull** consultar el meu compte **per a** saber el que dec
- **Com a** usuari **vull** consultar el meu compte **per a** saber els punts que tinc acumulats
- **Com a** usuari **vull** llogar una pel·lícula de Novetats [**per a** guanyar punts]

Test d'acceptació

- US: **Com a** usuari **vull** consultar el meu compte **per a** saber el que dec i els punts que tinc acumulats
 - **En cas que** l'usuari tingui només lloguers de pel·lícules de tipus **Novetat** **quan** l'usuari consulti el seu compte **el sistema** donarà el missatge “Has de pagar” \$deute “i tens “ \$punts “acumulats”.

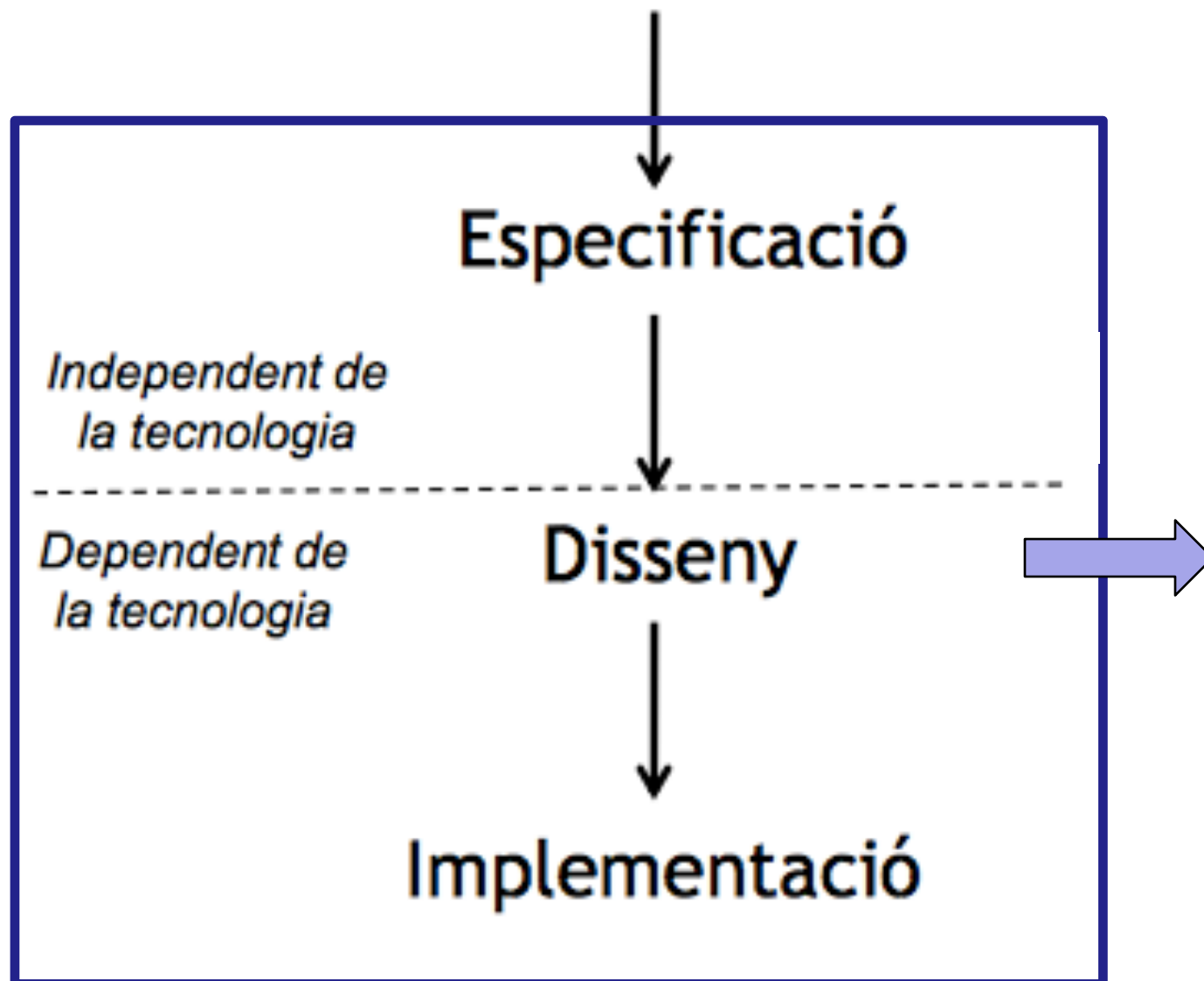
Model de domini

A partir de l'enunciat del problema i del vocabulari que ha sortit en els casos d'ús i les històries d'usuari es defineix el Model de Domini



Procés sistemàtic:

Anàlisi de requisits



Com ho fa el sistema?

- Descripció dels subsistemes i components del sistema software
- Relació entre components
- **Diagrama de classes**
- Diagrames d'activitats

Diagrama de classes de disseny

- Per a cada **test d'acceptació** es defineixen diferents tests amb dades concretes (**exemples**).
- Es dissenya la navegació per les classes implicades del Model de Domini (**Diagrama de Seqüència**)
- Només s'implementen les classes concretes que calen per a complir el test de valors concrets (**Diagrama de classes**)

Diagrama de classes de disseny

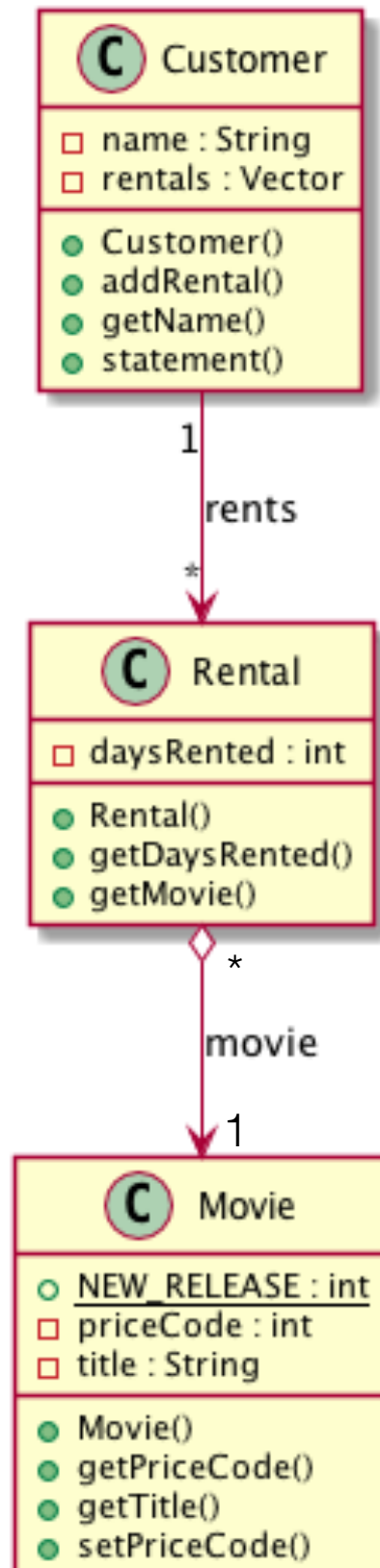
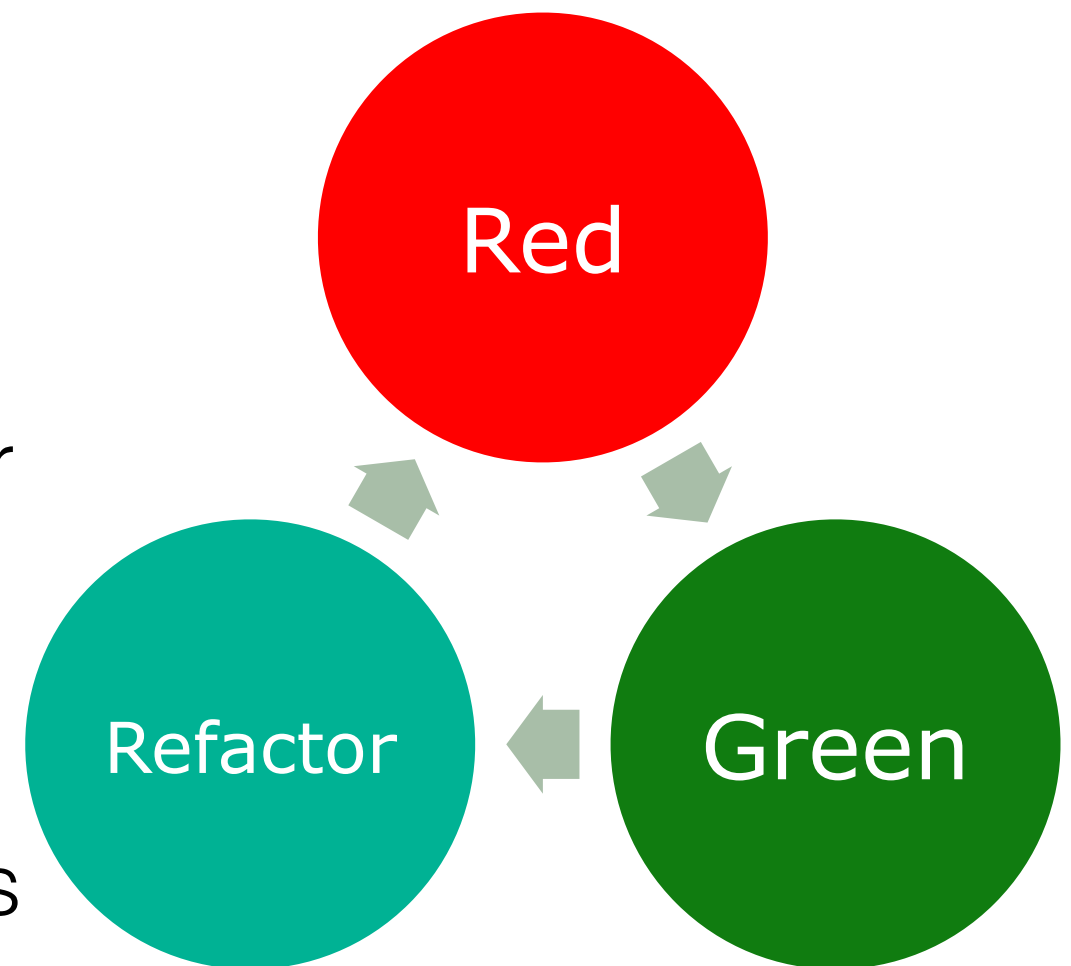


Diagrama de classes de disseny

- Per a cada test d'acceptació es dissenyen diferents tests amb valors concrets (exemples)
- Només s'implementen les classes concretes que calen per a complir el test de valors concrets
- Es testreja el test i quan passa es refactoritza el codi : Ús de **principis de disseny** i **patrons**



1.2. Desenvolupament guiat per tests

Red/Green/Refactor

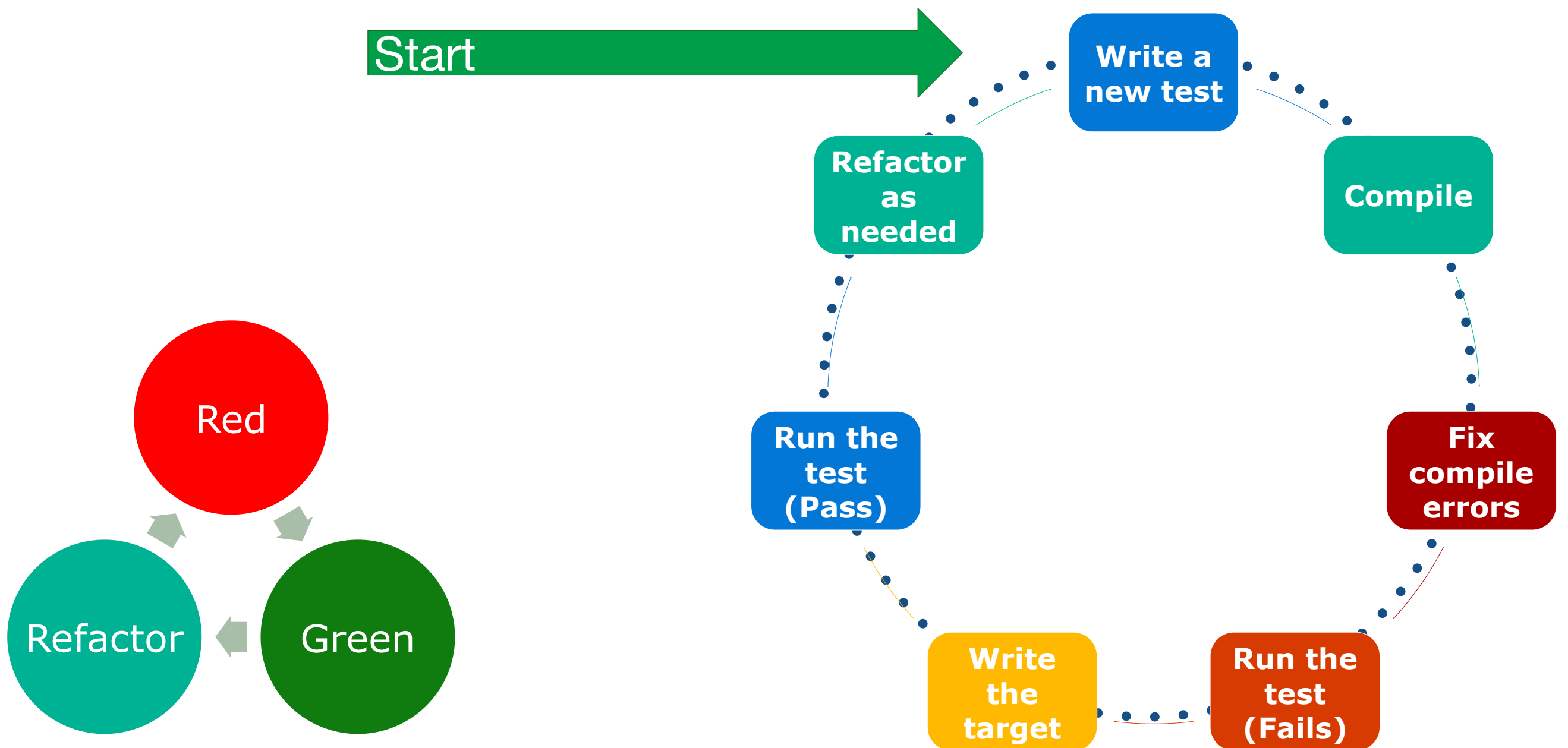
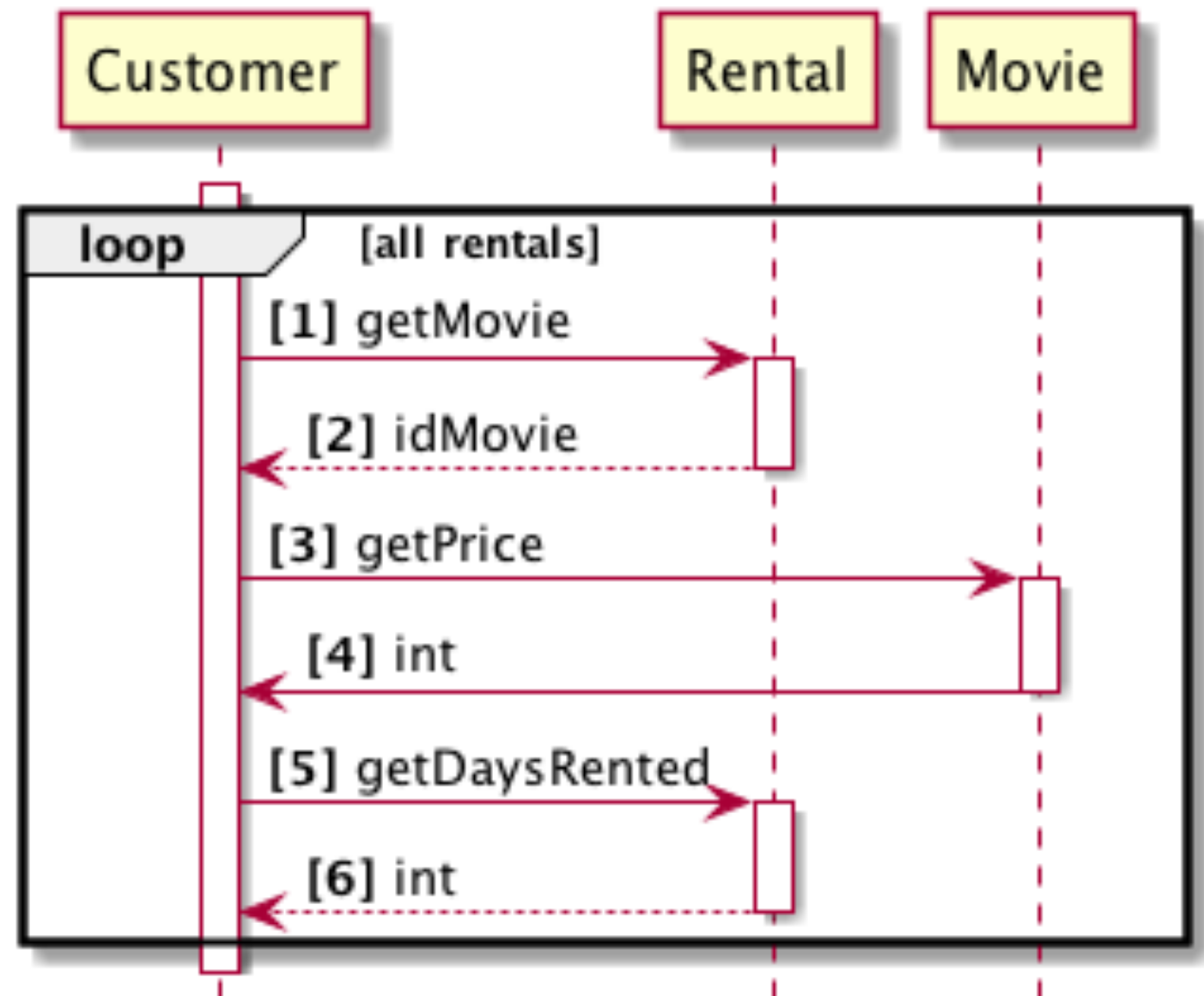


Diagrama de Sistema

Diagrama de Sistema

1. **En cas que** l'usuari llogui una pel·lícula de tipus "**Novetat**" quan l'usuari consulti el seu compte **el sistema** donarà el missatge "Has de pagar" \$deute "i tens " \$punts "acumulats".



Usuari: "Fred"

Lloga: "The Cell" - tipus Novetat - durant 3 dies

el sistema respon: "Rental Record for Fred The Cell You owed 9.0 You earned 2 frequent renter points",

Diagrama de classes de disseny

1. **En cas que** l'usuari tingui només lloguers de pel·lícules de tipus “**Novetat**” quan l'usuari consulti el seu compte **el sistema** donarà el missatge “Has de pagar” \$deute “i tens” \$punts “acumulats”.

Usuari: “Fred”

Lloga: “The Cell” - tipus Novetat - durant 3 dies

el sistema respon: “Rental Record for Fred The Cell You owed 9.0

You earned 2 frequent renter points”.

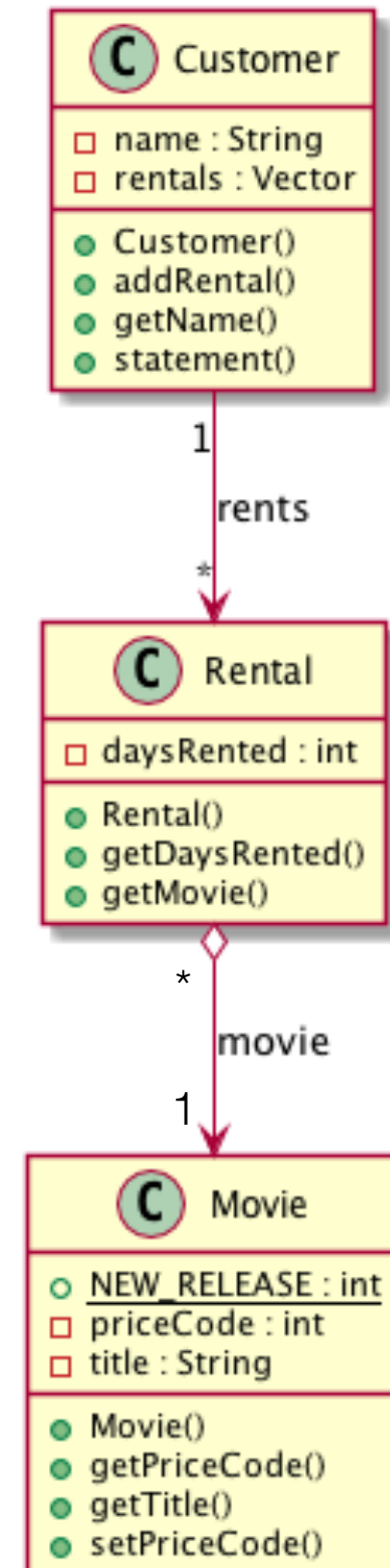


Diagrama de classes de disseny

- Per a cada test d'acceptació es dissenyen diferents tests amb valors concrets
- Només s'implementen les classes concretes que calen per a complir el test de valors concrets
- Es testeja el test i quan passa es refactoritza el codi : Ús de **principis de disseny** i **patrons**

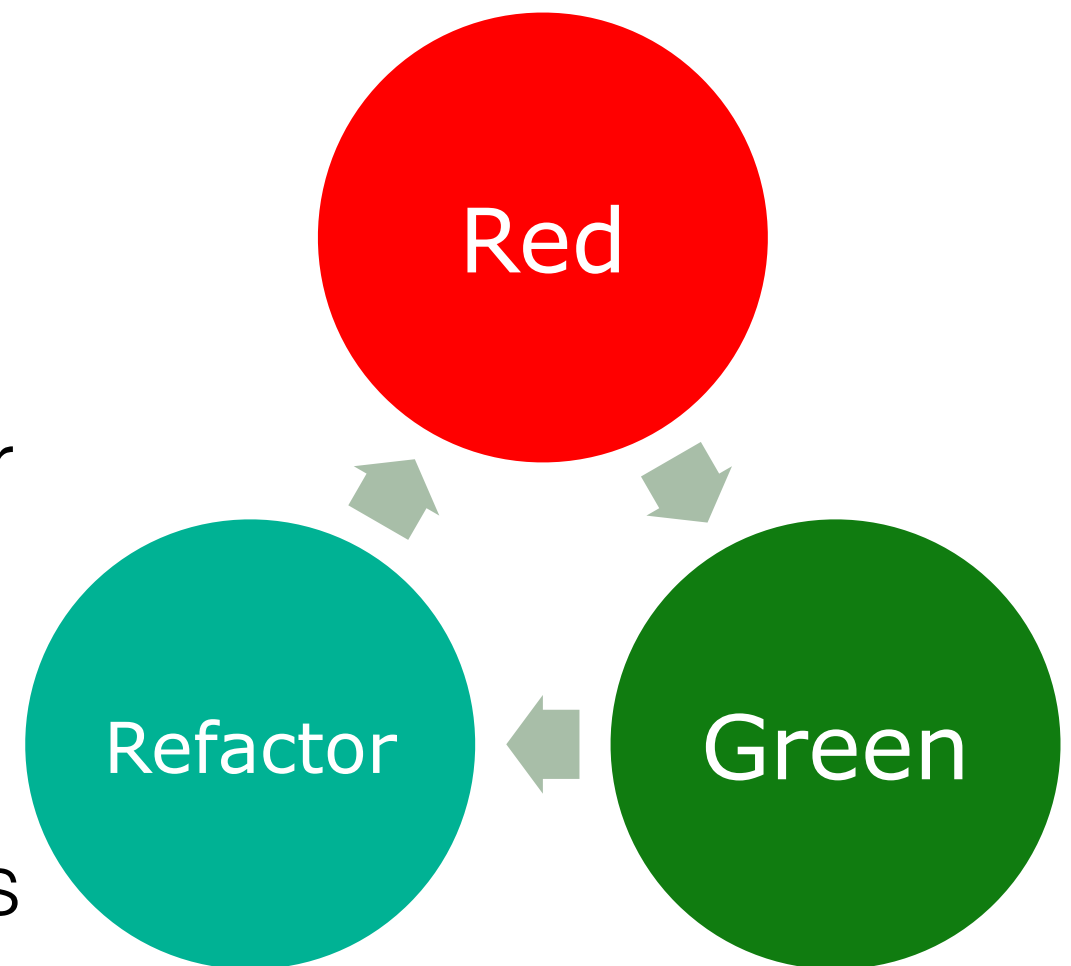


Diagrama de classes de disseny

2. **En cas que** l'usuari tingui només lloguers de pel·lícules de tipus “**Infantil**” quan l'usuari consulti el seu compte **el sistema** donarà el missatge “Has de pagar” \$deute “i tens “ \$punts “acumulats”.

Usuari: “Fred”

Lloga: “The Tigger Movie” - tipus Infantil - durant 3 dies

el sistema respon: “Rental Record for Fred The Tigger Movie You owed 1.5 You earned 1 frequent renter points”,

