

Metodologías de Desarrollo de Software

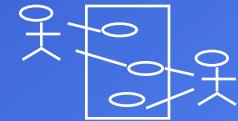
Unified Modeling Language (UML)

Ivar Jacobson

Grady Booch

James Rumbaugh

Use Cases - Elements



- **Use case:** A use case involves the interaction of actors and the system
- **Actor:** An actor represents a coherent set of roles that use cases play when interacting with these use cases
- **Activation:** Interaction between a use case and an Actor
- **Relationships:** Inclusion, extention, generalization



Actor



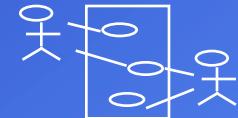
Use case



Activation



Relationship



Use Cases

Nombre del caso de uso: Nombre representativo de dicha funcionalidad

Descripción breve: Descripción en un párrafo de lo que hace ese caso de uso

Actor primario/secundario(s): Actores involucrados en el CU

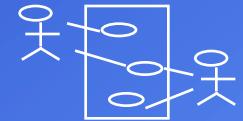
Trigger: Situación por la que el caso de uso se activa

Curso básico: Flujo de ejecución ideal

Cursos alternativos: Desviaciones del flujo de ejecución ideal

Precondición: Algo que debe ser verdadero (chequeado por el caso de uso) antes de desarrollar su funcionalidad

Use Cases - Relationships

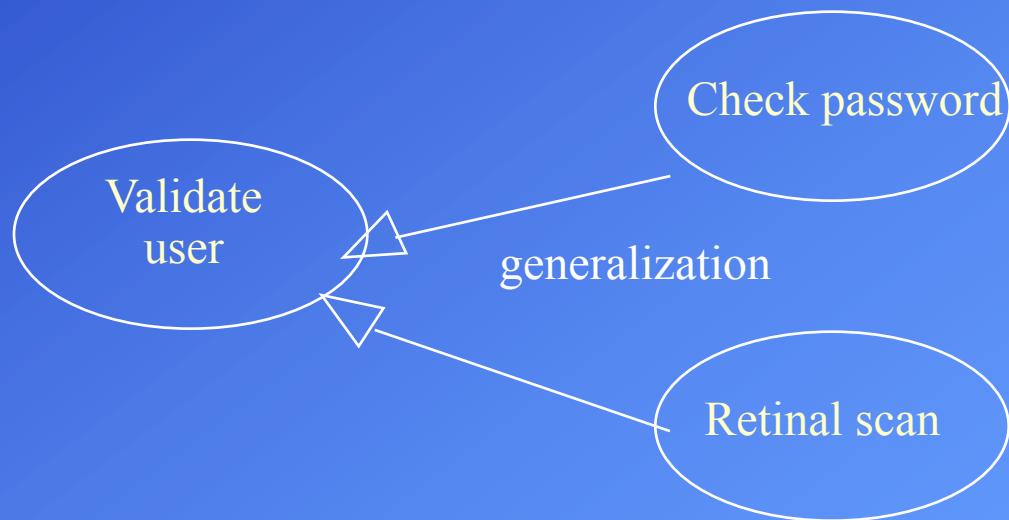


- *Generalization* among use cases is just like generalization among classes
- An *include* relationship between use cases means that the use case explicitly incorporates the behavior of another use case at a location specified in the base. The included use case never stands alone
- The *extend* relationship between use cases means that the base use case implicitly incorporates the behavior of another use case at a location specified indirectly by the extending use case

Use Cases - Generalization



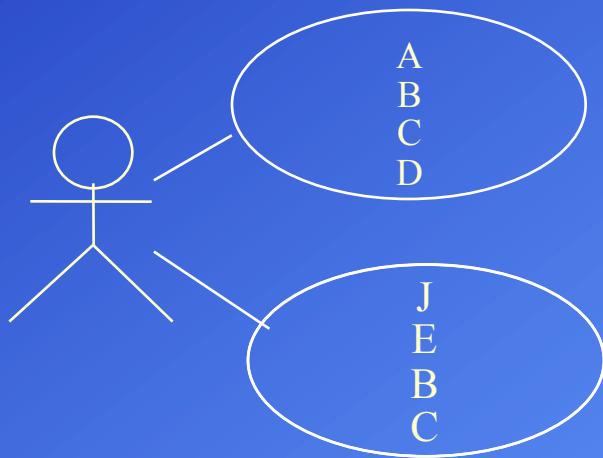
- As in a class diagram. If a use case can be developed in different ways so a generalization relationship is needed
- There are as use cases as different development use case alternatives are, and the father





Use Cases - Include

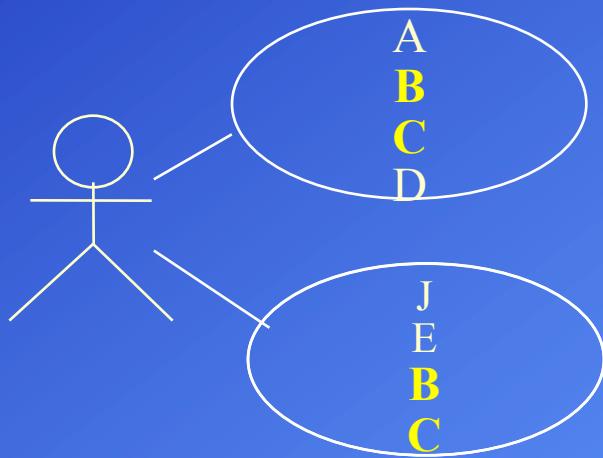
- When two or more use cases share a part of behavior so the shared behavior is factorized in a new use case
- The new use case is not a normal use case, that is it will not be activated by an actor (90%)
- When the original use case is activated the included use cases are activated too (always)
- The include relationship is “explicit”





Use Cases - Include

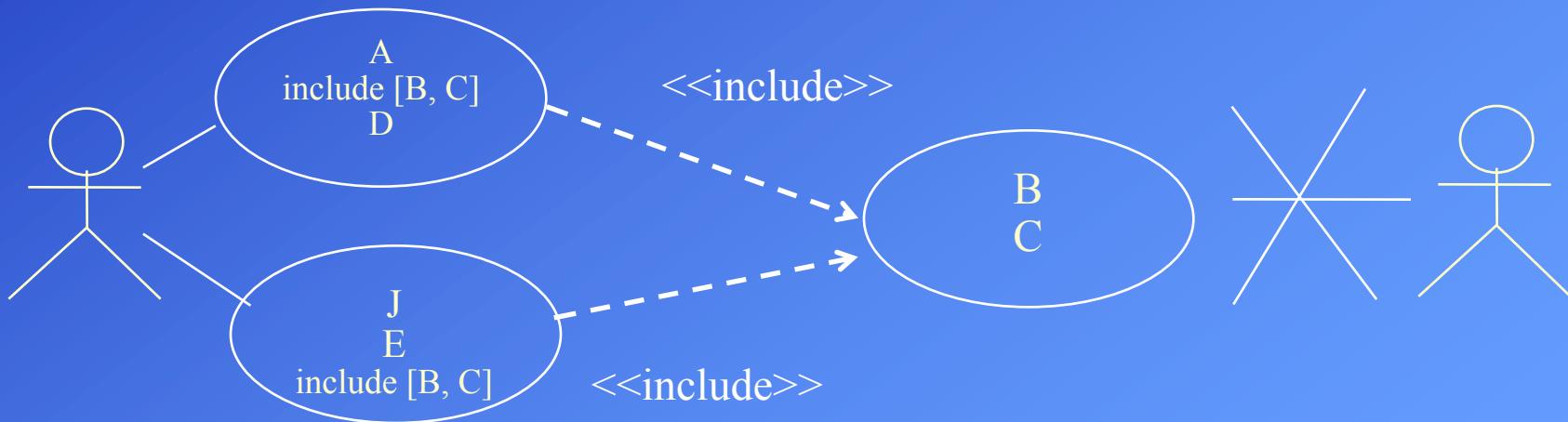
- When two or more use cases share a part of behavior so the shared behavior is factorized in a new use case
- The new use case is not a normal use case, that is it will not be activated by an actor (90%)
- When the original use case is activated the included use cases are activated too (always)
- The include relationship is “explicit”



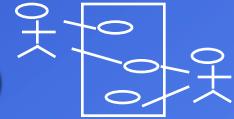


Use Cases - Include

- When two or more use cases share a part of behavior so the shared behavior is factorized in a new use case
- The new use case is not a normal use case, that is it will not be activated by an actor (90%)
- When the original use case is activated the included use cases are activated too (always)
- The include relationship is “explicit”



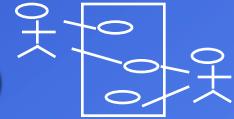
Ejemplo – El Renacuajo



Cuando un socio se inscribe a un curso, Josefa le muestra los cursos disponibles con su horario, profesor y fecha de inicio. Además, controla que el socio haya optado previamente por la modalidad curso de natación. Además, controla que no haya más de 15 inscriptos en el curso. Si esta situación ocurre, Josefa abre un nuevo curso e inscribe a la persona interesada.

Para concurrir al complejo un socio de piletas libres debe solicitar un turno indicando un día y horario. Cuando esto ocurre, Josefa controla que el socio haya optado previamente por la modalidad. Además, verifica que exista un andarivel disponible. Se considera que un andarivel se encuentra disponible cuando en esa fecha y horario no se dicta ningún curso ni existen más de 4 socios con reserva.

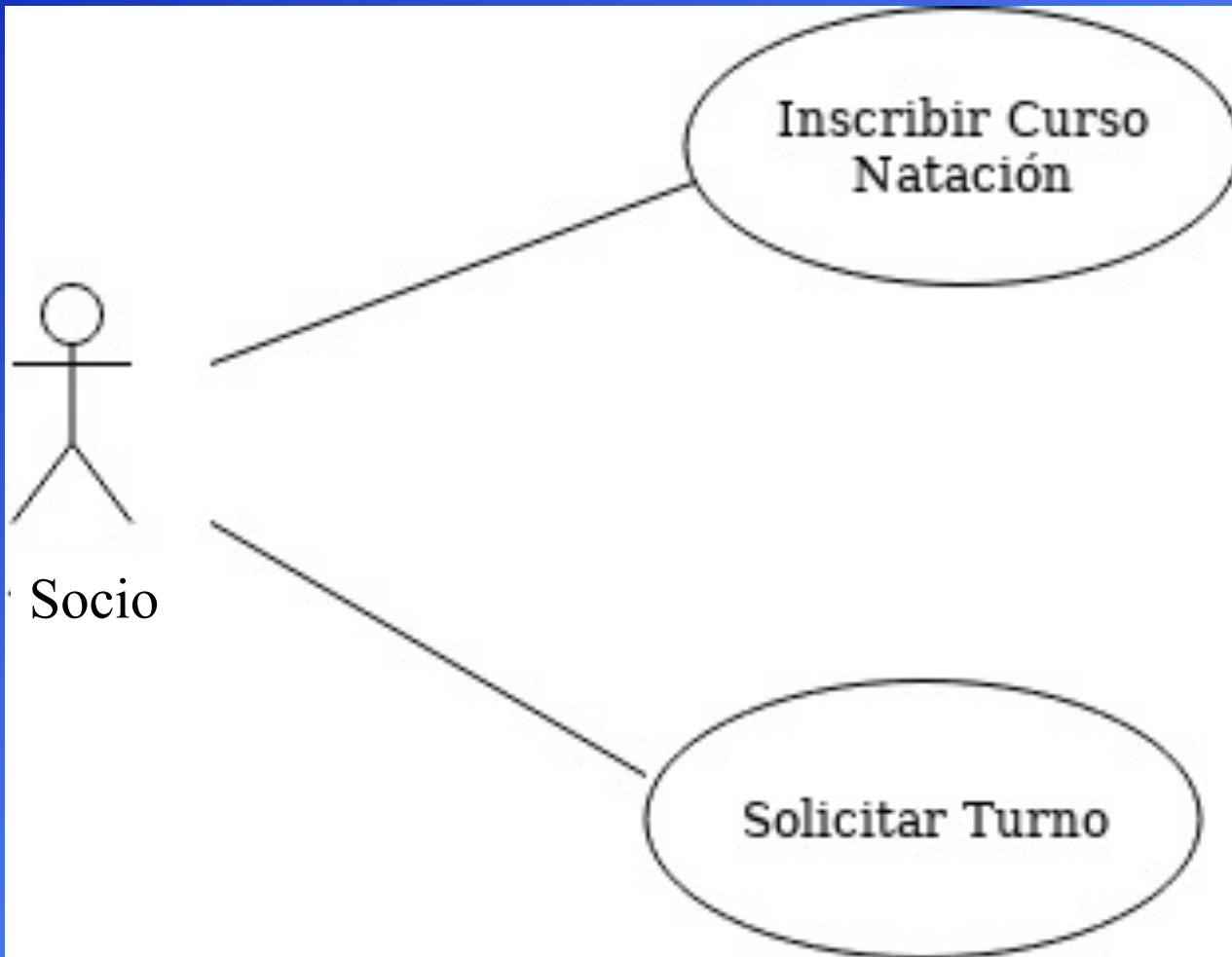
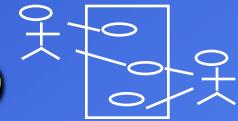
Ejemplo – El Renacuajo



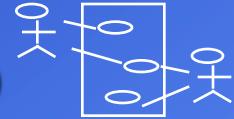
Cuando un **socio se inscribe a un curso**, Josefa le muestra los cursos disponibles con su horario, profesor y fecha de inicio. Además, controla que el socio haya optado previamente por la modalidad **curso de natación**. Además, controla que no haya más de 15 inscriptos en el curso. Si esta situación ocurre, Josefa abre un nuevo curso e inscribe a la persona interesada.

Para concurrir al complejo un socio de piletta libre debe **solicitar un turno** indicando un día y horario. Cuando esto ocurre, Josefa controla que el socio haya optado previamente por la modalidad. Además, verifica que exista un andarivel disponible. Se considera que un andarivel se encuentra disponible cuando en esa fecha y horario no se dicta ningún curso ni existen más de 4 socios con reserva.

Ejemplo – El Renacuajo



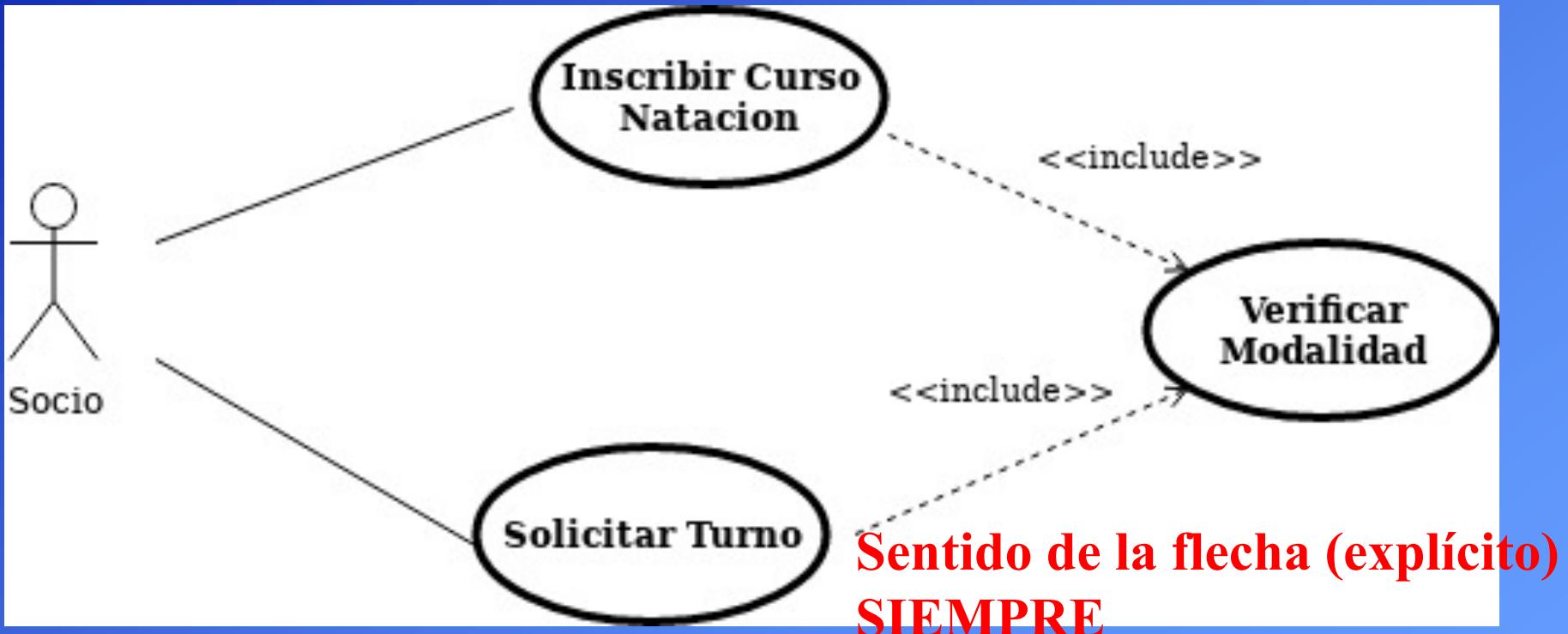
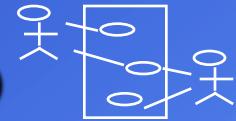
Ejemplo – El Renacuajo



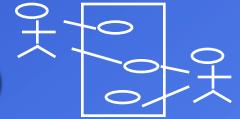
Cuando un socio se inscribe a un curso, Josefa le muestra los cursos disponibles con su horario, profesor y fecha de inicio. Además, *controla que el socio haya optado previamente por la modalidad* curso de natación. Además, controla que no haya comenzado y que no tenga más de 15 inscriptos en el curso. Si esta situación ocurre, Josefa abre un nuevo curso e inscribe a la persona interesada.

Para concurrir al complejo un socio de piletas libre debe solicitar un turno indicando un día y horario. Cuando esto ocurre, Josefa *controla que el socio haya optado previamente por la modalidad*. Además, verifica que exista un andarivel disponible. Se considera que un andarivel se encuentra disponible cuando en esa fecha y horario no se dicta ningún curso ni existen más de 4 socios con reserva.

Ejemplo – El Renacuajo



Ejemplo – El Renacuajo

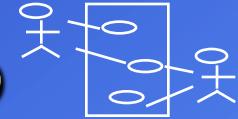


Nombre Caso de Uso: Inscribir curso de natación

Descripción: Este caso de uso permite los socios se inscriban a cursos de natación.

Actor primario: Socio

Ejemplo – El Renacuajo

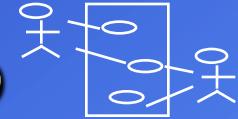


Curso básico:

1. **El Caso de Uso comienza cuando un socio quiere inscribirse en un curso de natación.**
2. **[INCLUDE] Verificar modalidad.**
3. **El Sistema lista los cursos disponibles. Por cada curso se detallan los horarios de asistencia, el profesor que lo dicta y la fecha de comienzo del curso.**
4. **El Socio selecciona uno de los cursos.**
5. **El Sistema verifica que el curso todavía no haya comenzado.**
6. **El Sistema verifica la cantidad de inscriptos en el curso.**
7. **El Sistema registra la inscripción al curso.**
8. **El Caso de Uso finaliza.**

Caso de uso incluido/que incluye: Verificar modalidad

Ejemplo – El Renacuajo



Nombre: Verificar modalidad

Descripción: Este caso de uso se utiliza para verificar que la modalidad del socio se corresponda con las actividades que realiza. Un socio de modalidad “curso de natación” tiene permitido inscribirse a cursos, mientras que un socio de modalidad “pileta libre” tiene permitido solicitar turnos.

Curso básico:

- 1. El Sistema obtiene la modalidad del socio**
- 2. El Sistema verifica la modalidad del socio con la actividad que quiere realizar.**
- 3. El caso de uso finaliza.**

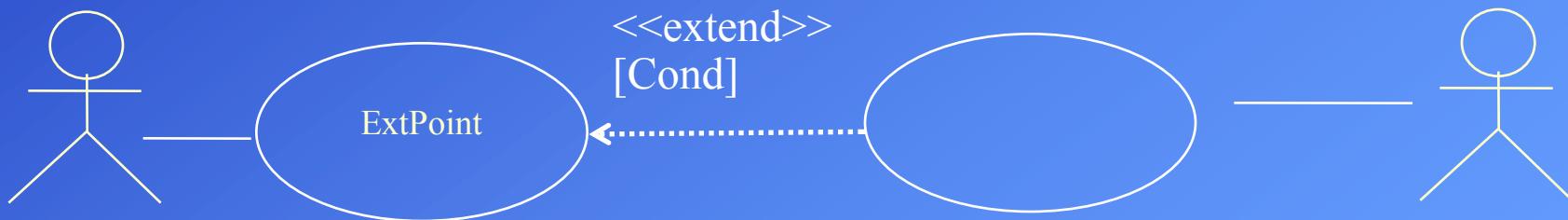
Curso alternativo: La modalidad es incompatible

- 2. El Sistema verifica la modalidad del socio con la actividad que quiere realizar.**
 - 2.1. El sistema muestra un error.**
 - 2.2. El sistema informa acerca de la incompatibilidad.**
 - 2.3. El caso de uso finaliza.**

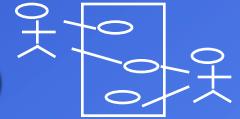


Use Cases - Extend

- If a condition is true a use case needs another to complete its behavior
- The new use case is a normal use case, that is it is activated by an actor (90%)
- When the original use case is activated according to the condition the extend use case is activated
- The extend relationship is “implicit”



Ejemplo – El Renacuajo

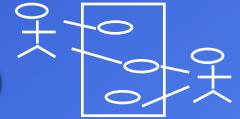


Cuando un nuevo socio se registra, la secretaria Josefa le pide su nombre y apellido, email, dirección y edad.

Adicionalmente, la secretaria solicita la modalidad en la que ese socio va a asistir.

En caso de no haberlo decidido aún, puede informarlo en cualquier momento. Actualmente, las modalidades en las que un socio puede asistir al complejo son: curso de natación o pileta libre.

Ejemplo – El Renacuajo

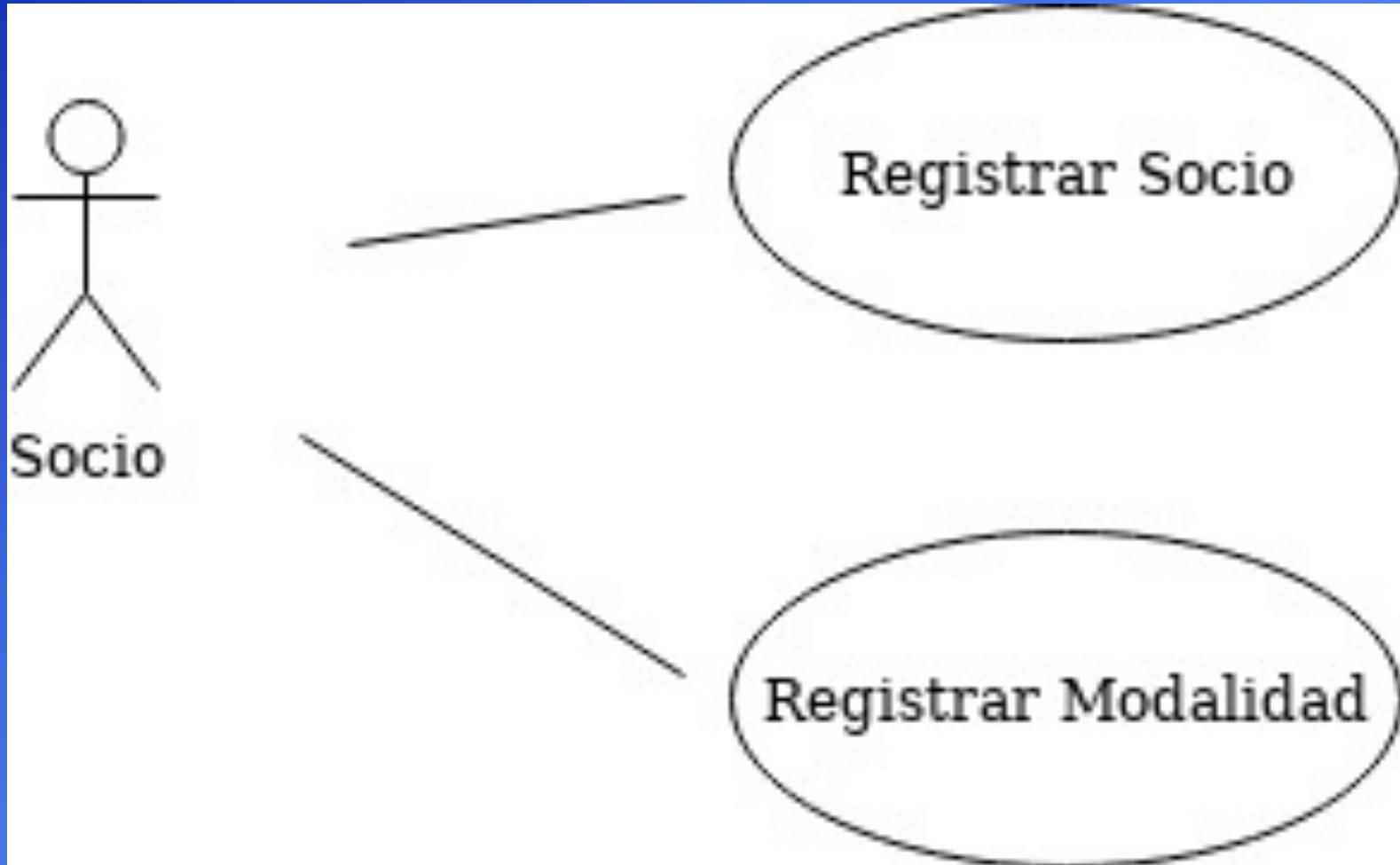
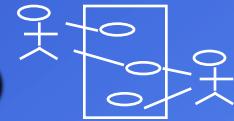


Cuando un nuevo socio se registra, la secretaria Josefa le pide su nombre y apellido, email, dirección y edad.

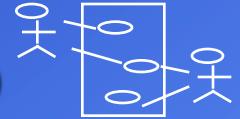
Adicionalmente, la secretaria solicita la modalidad en la que ese socio va a asistir.

En caso de no haberlo decidido aún, puede informarlo en cualquier momento. Actualmente, las modalidades en las que un socio puede asistir al complejo son: curso de natación o pileta libre.

Ejemplo – El Renacuajo



Ejemplo – El Renacuajo

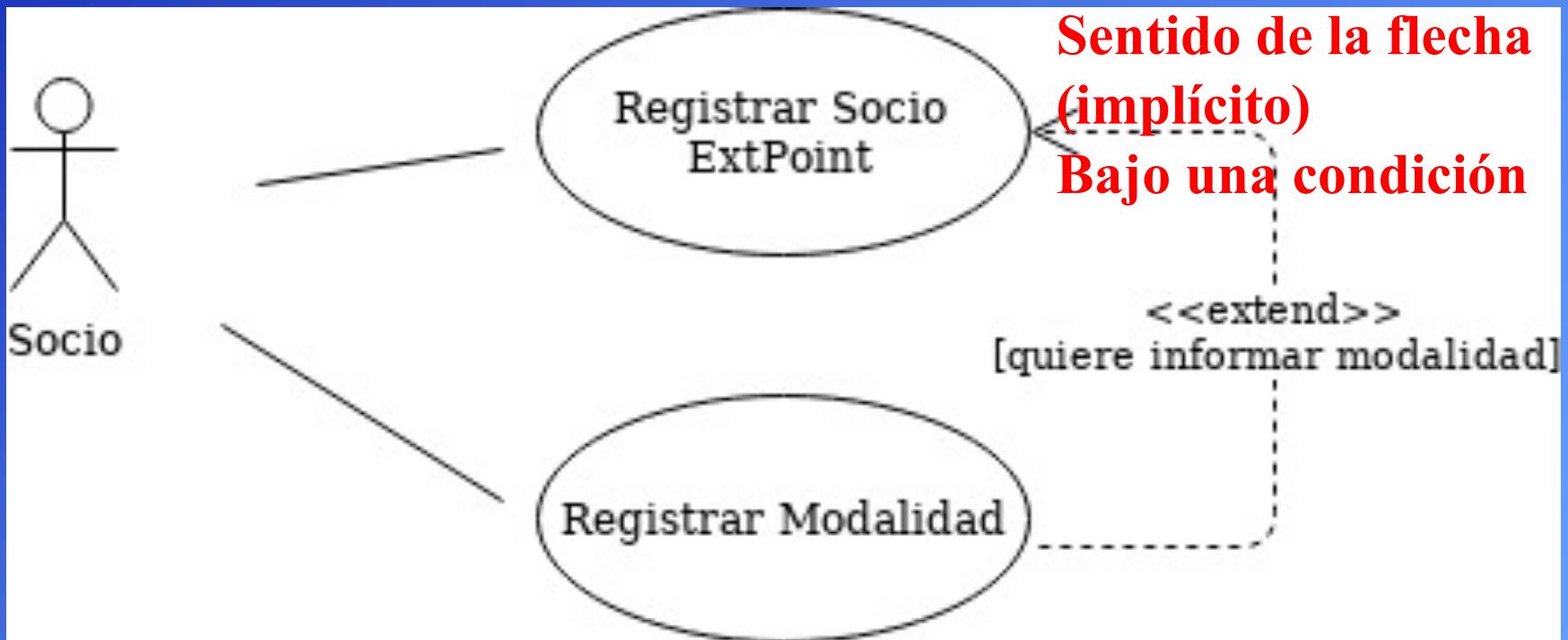
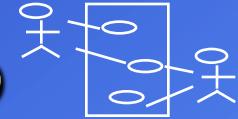


Cuando un nuevo socio se registra, la secretaria Josefa le pide su nombre y apellido, email, dirección y edad.

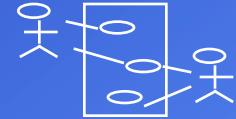
Adicionalmente, la secretaria solicita la modalidad en la que ese socio va a asistir.

En caso de no haberlo decidido aún, puede informarlo en cualquier momento. Actualmente, las modalidades en las que un socio puede asistir al complejo son: curso de natación o pileta libre.

Ejemplo – El Renacuajo



Use Case - Basic Flow

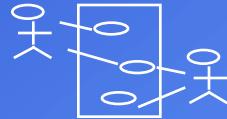


Nombre Caso de uso: Registrar Socio

Descripción breve: Un socio quiere registrarse en el complejo de la pileta cubierta "El Renacuajo"

Actor primario: Socio

Trigger: El caso de uso comienza cuando un socio quiere registrarse en el sistema

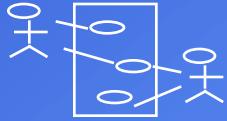


Basic Flow - Registrar Socio

Curso básico:

1. El Caso de Uso comienza cuando un socio quiere registrarse en el sistema.
2. El Sistema solicita los datos personales.
3. El Socio ingresa su nombre, apellido, email, dirección y edad.
4. El Sistema solicita un identificador único.
5. El Socio ingresa su número de documento.
6. El Sistema verifica el identificador ingresado.
7. [PtoExt]
8. El Sistema guarda la información del nuevo socio
9. El Caso de Uso finaliza

Basic Flow - Registrar Modalidad



Nombre Caso de Uso: Registrar modalidad

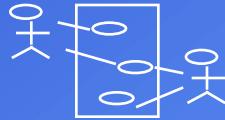
Descripción: Este caso de uso permite que un socio pueda seleccionar la modalidad en la que asiste a la pileta denatación.

Actor primario: Socio

Trigger: El caso de uso comienza cuando un socio desea informar la modalidad de asistencia a la pileta.

Basic Flow - Registrar Modalidad

Curso básico:



1. El caso de uso comienza cuando un socio desea informar la modalidad de asistencia a la pileta.
2. El sistema presenta las modalidades de asistencia a la pileta disponibles:

*Curso de natación

*Pileta libre

3. El socio selecciona una modalidad de asistencia.
4. El sistema verifica la existencia de datos previos.
5. El sistema actualiza la nueva modalidad.
6. El caso de uso finaliza.

Casos de uso extendidos: Extiende el caso de uso **Registrar socio.** Cuando el socio quiere informar **modalidad**



Basic Flow - Registrar Socio

Curso básico:

1. El caso de uso comienza cuando un socio quiere registrarse en el sistema.
2. El Sistema solicita los datos personales.
3. El Socio ingresa su nombre, apellido, email, dirección **extensión**.
4. El Sistema solicita un identificador único.
5. El Socio ingresa su número de documento.
6. El Sistema verifica el identificador ingresado.
7. Si el Socio quiere informar la modalidad, el sistema presenta las modalidades de asistencia a la pileta disponibles:
 - *Curso de natación
 - *Pileta libre
8. El Socio selecciona una modalidad de asistencia.
9. El Sistema verifica la existencia de datos previos

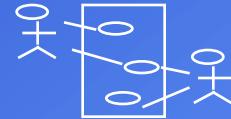
Suposición (Assumption)



• An assumption is a state that must be reached before the Use Case can be performed, but does not constitute an actual precondition of this Use Case, and is not specifically included in the Flow of Events

- The extra behavior must be handled by another use case
- A given Use Case does NOT have to have any associated assumptions. Often, a Use Case will not have any assumptions
- It seems to represent an “order activation” among use cases

Suposición (Assumption)



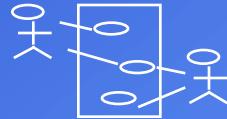
Curso básico: Flujo de ejecución ideal

Cursos alternativos: Desviaciones del flujo de ejecución ideal

Precondición: Algo que debe ser verdadero (chequeado por el caso de uso) antes de desarrollar su funcionalidad

Postcondición: Algo que debe ser verdadero (chequeado por el caso de uso) después de desarrollar su funcionalidad

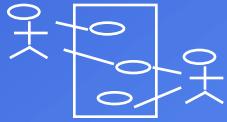
Suposiciones: Verificaciones que se suponen ya están comprobadas (el caso de uso no tiene que chequearla)



Basic Flow - Registrar Socio

Curso básico:

1. El Caso de Uso comienza cuando un socio quiere registrarse en el sistema.
2. El Sistema solicita los datos personales.
3. El Socio ingresa su nombre, apellido, email, dirección y edad.
4. El Sistema solicita un identificador único.
5. El Socio ingresa su número de documento.
6. El Sistema verifica el identificador ingresado.
8. [PtoExt]
9. El Sistema guarda la información del nuevo socio
10. El Caso de Uso finaliza



Basic Flow - Registrar Socio

Pre-condición:

1. Que no haya otro socio con el mismo identificador

Pos-condición:

9. Que la información haya sido almacenada

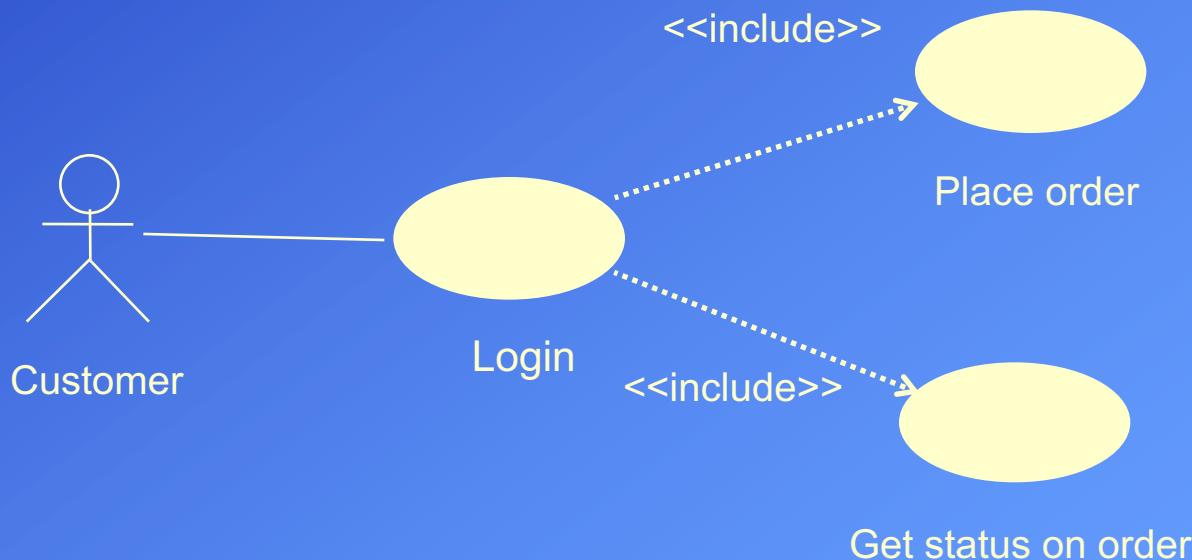
Suposición:

Que el Socio esté logueado

Login Use Case



- Login use case that includes all of the other use cases that are secure
- In addition to updating the diagram, it is also necessary to write the text of the login use case. None of the other use cases change.





Login Use Case

1. The use case begins when the customer starts the application
2. The system prompts the customer to enter a username and password
3. The customer enter a username and password
4. The system verifies that this is a valid user
5. While the customer does not select exit, do the following steps in any order
6. The customer may choose to place an order (include Place Order)
7. The customer may choose to get the status on an order (include Get Status on Order)
end loop.
8. The use case ends

Login Use Case



Benefits

- The use case diagram looks like what the designer expect: The customer logs in, and from there he she can access any of the allowed system functions

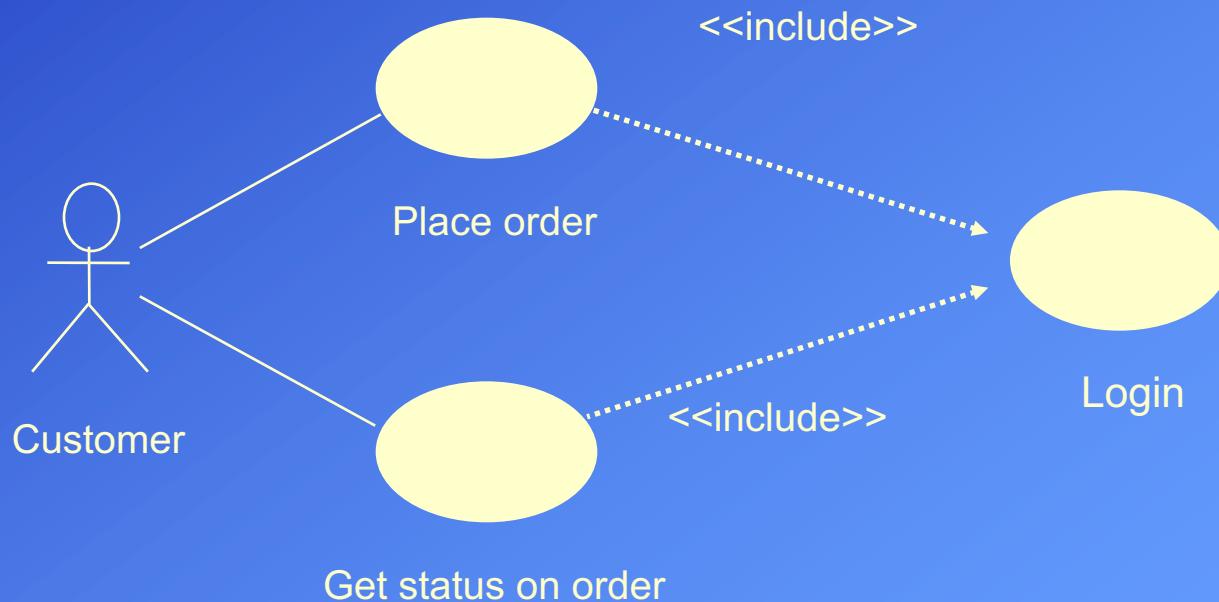
Drawback

- The text of the use case is difficult to maintain in case a new use case is needed in the system. It is possible to forget to make the change to login
- It is not good that the login has the knowledge of all other parts of the system, Use cases have to be independent of each other

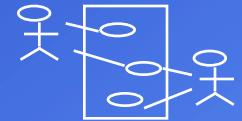
Login Use Case



- The other use cases include login
- It is necessary to update the use case diagram and the text



Login Use Case



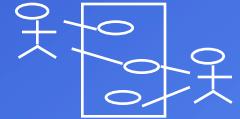
Login Use Case

1. The use case begins when the customer starts the application
2. The system prompts the customer to enter a username and password
3. The customer enter a username and password
4. The system verifies that this is a valid user
5. The use case ends

Partial Place Order Use Case

1. The use case begins when the customer logs in to the system (include login)

Login Use Case



Benefits

- The login use case describes login and nothing else

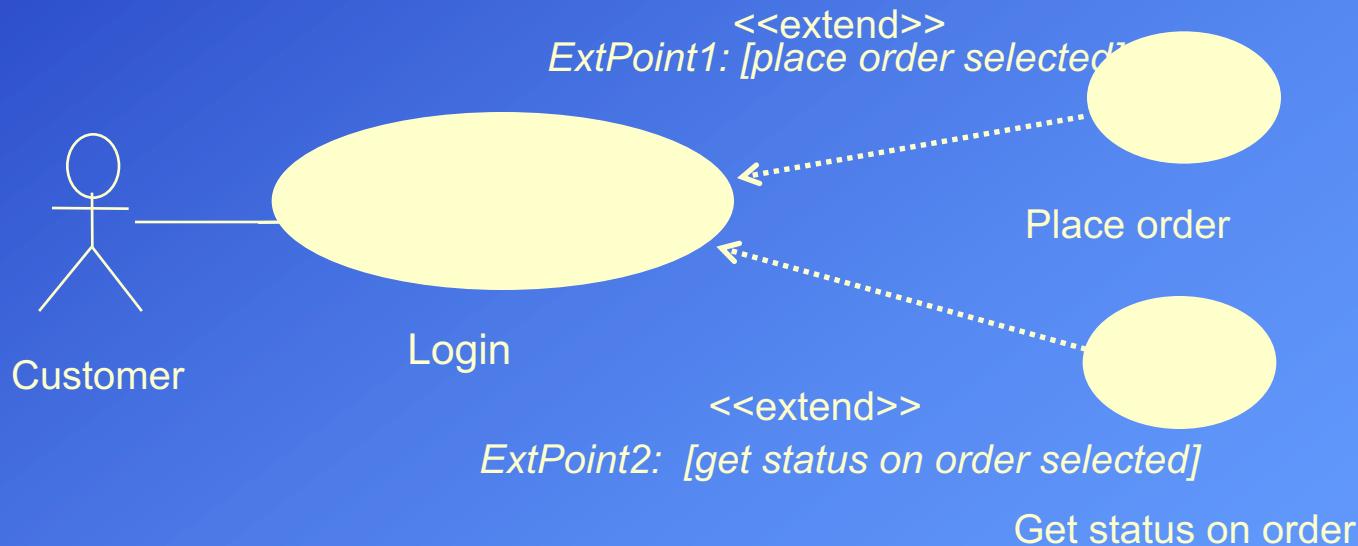
Drawbacks

- The Place Order use case and the diagram make it appear that the customer has to log in every time s/he wants to do something

Login Use Case



- The other use cases extend the login use case
- It is necessary to update the use case diagram and the text



Login Use Case



- 1. The use case begins when the customer starts the application**
- 2. The system prompts the customer to enter a username and password**
- 3. The customer enter a username and password**
- 4. The system verifies that this is a valid user**
- 5. While the customer does not select exit**
- 6. Extension point: Do It**
- 7. The use case ends**

Login Use Case



Benefits

- The customer logs one time and from there gets access to the resto of the system. Login adds an extension point, but that is all, it is no necessary to change it when adding new use cases

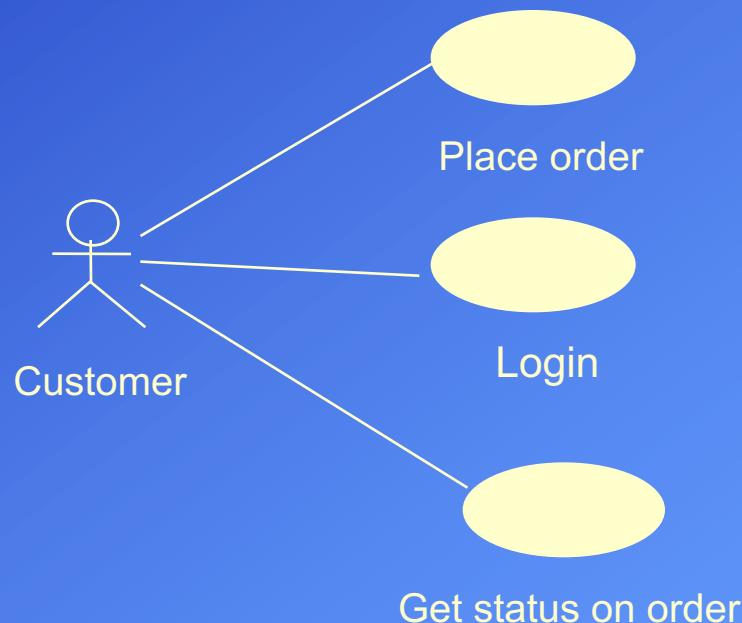
■ Drawbacks

- The description of the extensio. It can be a difficult relationship to explain, especially to people who are not developers

Login Use Case



- The login use case can be completely independent of the other use cases, but a precondition is included in the other use cases that a valid user be logged in before they can be executed
- It is necessary to update the use case diagram and the text



Login Use Case

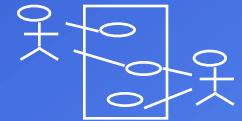


1. The use case begins when the customer starts the application
2. The system prompts the customer to enter a username and password
3. The customer enter a username and password
4. The system verifies that this is a valid user
5. The use case ends

Partial Place Order Use Case

Precondition: A valid user has logged into the system

Login Use Case



Benefits

- The login only describes login and nothing else
- The diagram and the text are clear and easy to understand and the system is more flexible
- Place Order does not require login to execute, but just a valid user. Executing login is one way to get a valid user, but there may be other validation methods as well

Login Use Case



- The last approach seem to be the easiest to read and it has the most flexibility
- However, all the approaches are correct, so the developer can pick the one that works best for his/her project

Documenting Create, Read, Update, Delete

- This kind of use cases are often necessary in applications that include maintaining data
- Two approaches:
 - Create a separate use case for each kind of access to the data
 - Create one use case for a data type that embeds all of the CRUD functions
- When the use case is named by the behavior that the user expects, they are not called Create Order, Read Order, Update Order and Delete Order. The developer uses names that makes sense to the users of the system

Documenting Create, Read, Update, Delete

- It is possible to make one use case called something like Maintain Orders, which have been in charge of anything having to do with orders. This does not really make sense because different actors use separate CRUD functions, so it makes more sense to make them separate use cases
- On the other hand, in some applications combining them may be sensible, for example, if the application is for a database administrator to update tables in a database
- The best way is to begin with separate use cases and then analyze if they can be merged to make them easier to read and maintain