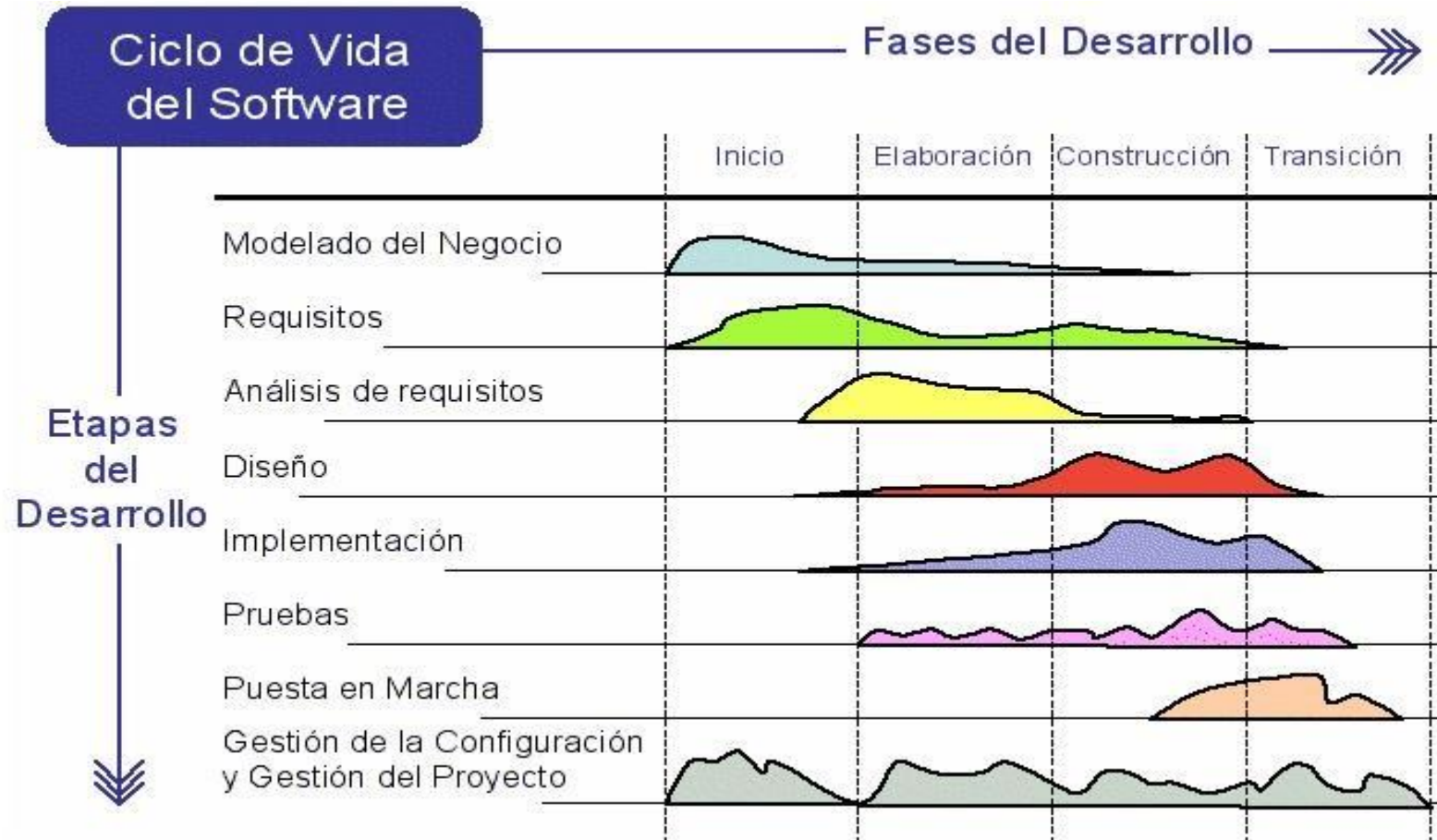


Importancia de modelar

- Caso
- Como nos organizamos?
- Elegimos un proceso de desarrollo, una guía de las actividades a seguir.
 - PUDS? Fase de Inicio, Fase de Elaboración, Fase de Construcción, Fase de Transición.
 - Complementamos con nuestra experiencia:
 - En ese proceso comenzamos definiendo los objetos o los requerimientos?
- Modelamos:
 - Que modelamos? Dominio, Negocio, Sistema ?
 - Que diagramas usamos: Diagrama de contexto, diagrama de flujo, diagrama de actividad, ..

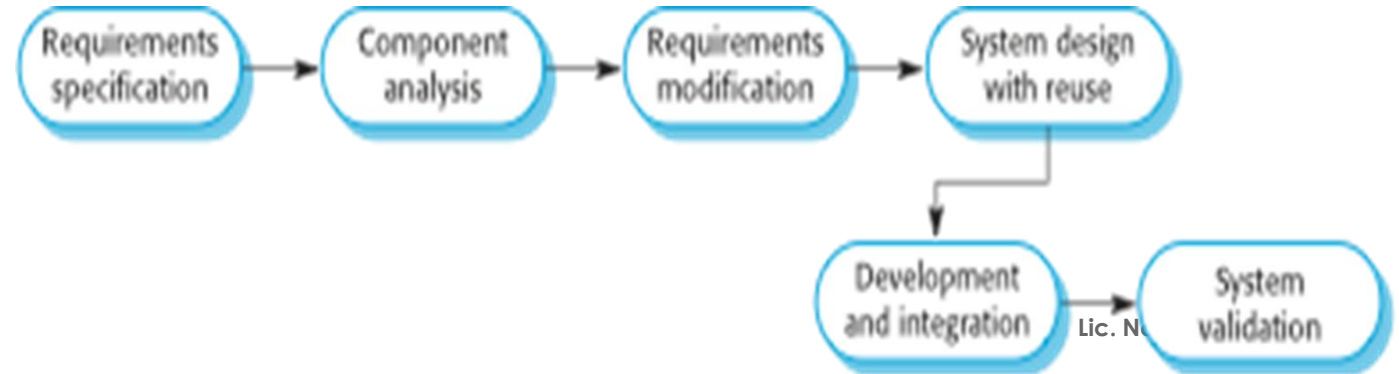
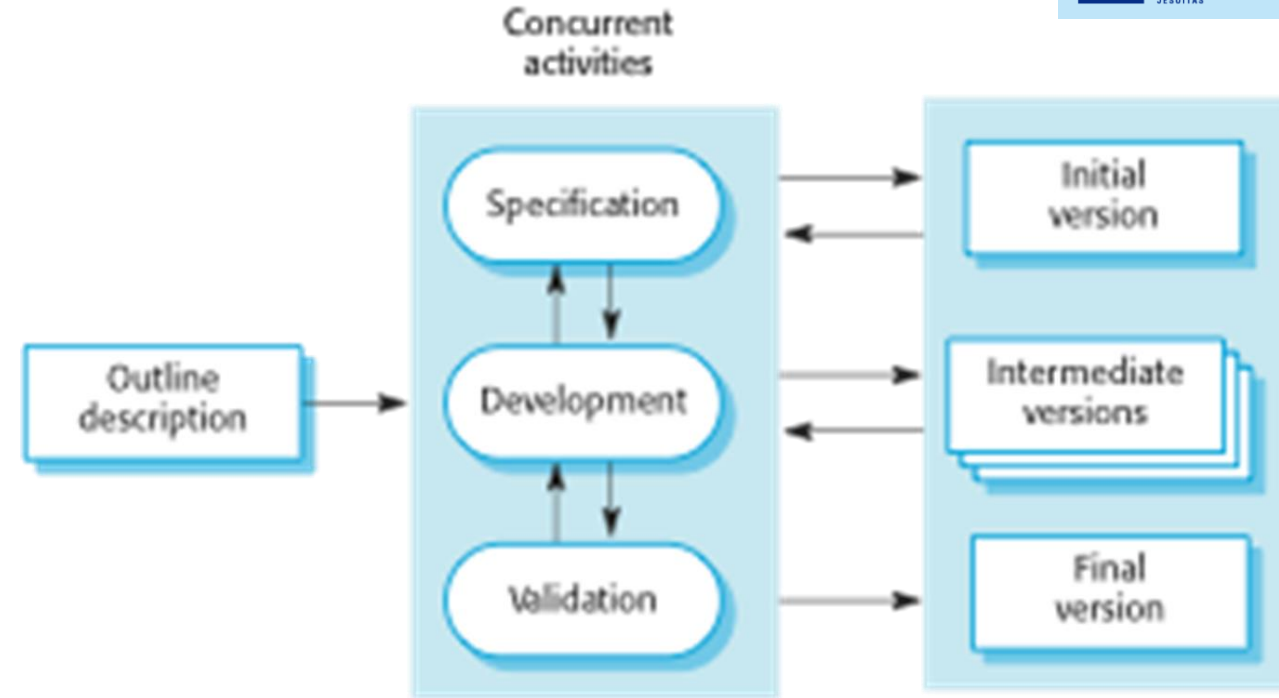
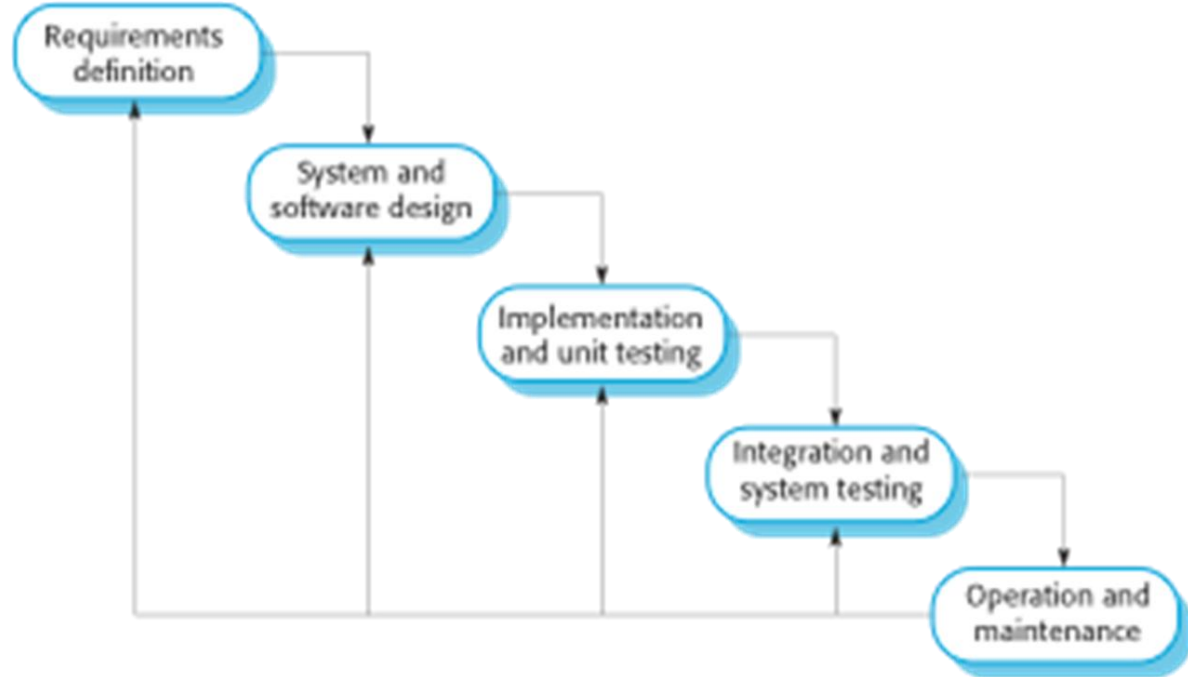
Fases



Procesos de desarrollo - Tipo de sistemas

- a) Un juego que pueda ser utilizado en cualquier dispositivo móvil.
- b) Un sistema que registre los controles de calidad de un producto que se acopla a un sistema de gestión de producción existente.
- c) Un sistema de monitoreo de sensores de movimiento y señales de alarma.
- d) Una sistema web para ventas online y envíos a domicilio.
- e) Un sistema de contabilidad doméstica o pequeño negocio para Windows más funcional, práctico y sencillo de manejar del mercado.
- f) Software embebido que acciona los comandos de lavado según los programas establecidos y registra los ciclos de lavado.
- g) Software de diseño tridimensional que permite proyectar objetos en el espacio, modificar escalas, perspectivas y tamaños; admitiendo también la visualización simultanea desde varias perspectivas por medio de ventanas
- h) Aplicación para el celular que permita conocer los horarios y recorridos de los transportes publicos cercanos a la ubicación del individuo que lo utilice.
- i) Programa didáctico y entretenido para aprender trigonometría. Contiene: funciones trigonométricas, ángulos asociados, funciones circulares, resolución de triángulos rectángulos, polígonos regulares, relación entre ángulos, triángulos isósceles, teoremas del seno y del coseno, etc.

Proceso de desarrollo



Criterios a considerar

Criterio 1: Nivel de flexibilidad de los requerimientos

- Desarrollo web -- Desarrollo Evolutivo -- Agil
- Cambios permanentes que deben ser introducidos frecuentemente – Desarrollo Evolutivo
- Desarrollo de aplicaciones clásicas, ambiente estable, no hay cambios en el dominio, los equipos de trabajo son estratificados, las fases de desarrollo son estables se respetan todas las etapas planificadas, entorno predecible. --Cascada

Criterio 2: considerar la escala y el alcance del desarrollo

- Proyectos grandes con mucha cantidad de integrantes, hay que realizar planes de gestión bien elaborados. – Cascada

Criterio 3: que conviene más teniendo en cuenta lo que exige el cliente

- Desarrollar software a través de sprint facilitan el lanzamiento de sistemas realmente complejos y genera una impresión de progresión rápida. Para clientes exigentes que quieren ver avances o que no confían en lo que se está desarrollando. Clientes que no se hacen una idea del sistema que necesitan, tienen que ver funcionando algo para tener una visión de lo que realmente quieren. – Desarrollo Evolutivo
- Contamos con períodos de desarrollos a largo plazo, con fechas de entrega que no se aproximan rápidamente. --Cascada

Criteria to consider

Criterion 4: Define the final users clearly

- A controlled group of final users probably has a set of fixed requirements with which to work, which makes the waterfall method ideal. But if the final objectives are dispersed, there will be a need to take care of the feedbacks after the launch of the application by requesting the inclusion of new functions

Criterion 5: Consider where the development team is located

- If the development team is dispersed all over the world, there will be a need for a higher level of coordination and responsibility. In this case, a more rigid project management regime is the best option, and this is the case where the waterfall method shines more. The method requires more frequent contact and more united teams, because a dispersed development team could work with a lot of confusion and errors in the development process if the chosen model is Agile.

What other criteria suggest for choosing a development process?? Propose at least 3 more.

Modelado de Negocio

- la perspectiva de cómo se mueven los procesos en el lugar en donde se construirá una solución de software (institución, empresa dominio)
- Los diagramas de proceso del negocio, son representaciones de los procesos que se dan en la organización y que son de interés para el sistema.
- Se trata de representar la relación existente entre un proceso y las unidades funcionales (como pueden ser los departamentos o personas) responsables del proceso.
- Es un diagrama de flujo extendido, en donde además de las operaciones, se detalla las personas que intervienen en un proceso

Importancia del objetivo..

- OBJETIVO: Los objetivos deben estar planteados desde la perspectiva del negocio.
 - Deben definir porque se construye el sistema para tal fin.
 - Que beneficios obtiene el negocio en la construcción del sistema.
 - No debe describir funcionalidades
- Esto nos permite ubicarnos desde la perspectiva del cliente y del usuario y no solamente como desarrollador.
- **Ej: Un Sistema contra incendios y detección de intrusos para un edificio de oficinas.**
- ~~*OBJETIVO INCORRECTO: Construir un sistema de alarma contra incendios e intrusos para el edificio que proporcione avisos de fuego y de avisos de accesos no autorizadas tanto internas como externas.~~
- *OBJETIVO CORRECTO: Asegurar que el funcionamiento normal de los trabajos realizados en el edificio no se interrumpa por eventos como el fuego y accesos no autorizados.

Con que arranco primero? Con requerimientos o con objetos?

- Procesos funcionales
- Areas funcionales
- Lista de requerimientos (varias versiones)
- Enfoque de casos de usos (proceso de encontrar requerimientos)

- Conceptos relevantes
- Objetos del dominio
- ...

- **Con cuanto nivel de detalle?**

Veamos el ejemplo

- **Objetivo del sistema:**
- Proveer contenido digital, propio y de terceros, mediante streaming a través de la suscripción paga mensualmente.
- **Subsistemas que lo componen**
 - Subsistemas de Cuentas
 - Subsistemas de Contenido
 - Subsistemas de Pago

The Netflix logo is displayed in a bold, red, sans-serif font. The letters are slightly irregular and have a hand-drawn feel, with the 'N' and 'X' being particularly prominent.

Algunos requerimientos

- “El sistema debe permitir buscar contenido de películas”
- “El sistema debe poder transmitir streaming en vivo de campeonatos de videojuegos”
- “El sistema debe permitir configurar la Home mostrando contenido relevante para el usuario”
- “El sistema debe ofrecer una prueba gratis durante 1 mes con acceso limitado a ciertas secciones con posibilidad de ingresar una tarjeta al registrarse o luego del mes”
- “El sistema debe permitir ofrecer contenido interactivo”

Ejercicio Grupal

- En grupos trabajen sobre el sistema de Spotify definiendo:
 - Objetivo del sistema.
 - Identificación de subsistemas que lo componen.
 - Identificar requerimientos funcionales y no funcionales.



Modelado de requerimientos: Casos de uso

- **¿Qué es un Modelo?**

- Representación en un cierto medio de algo → Ayuda a **visualizar** el sistema
- Tratamos de **captar** lo importante → Definimos un **punto de vista** → **Simplificamos** la realidad.
- Busca reducir y atacar la **complejidad** inherente en los sistemas

- **¿Por qué Modelamos?**

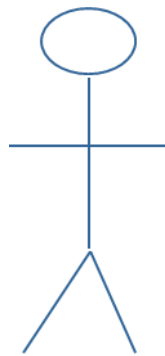
- Conocimiento del dominio, captar requisitos, diseñar, captar decisiones, usabilidad, explorar soluciones, organizar el desarrollo, definir estructura, definir comportamiento...
 - **Abstrae** → **Específica** → **Describe**

Diagramas de Casos de Uso

- Especifica **requerimientos funcionales** del Sistema.
- Se utiliza a nivel de Sistema o Negocio.
- El proceso de desarrollo está dirigido por los **Casos de Uso**.
- El diagrama trata de identificar:
 - **Actores**, quiénes interactúan con el sistema, ya sea de forma directa o indirecta.
 - **Casos de uso**, que representan la funcionalidad del sistema.
 - **Relaciones entre actores y casos de uso**, determina funcionalidades que pueden realizar.
 - **Relaciones entre casos de uso y casos de uso**, para expandir las funcionalidades del sistema.

Modelado de requerimientos: Casos de uso

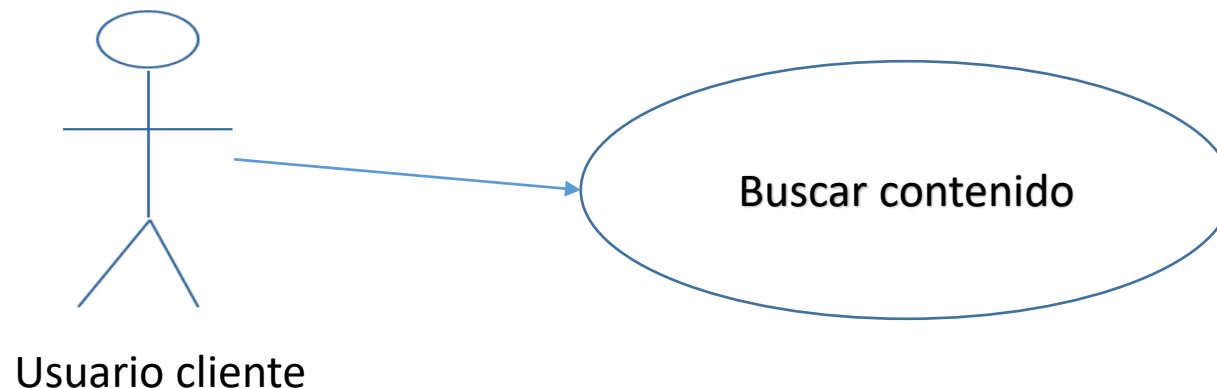
- Identificando los **Actores**
 - Representa un rol que interactúa con el sistema y ejecuta las funcionalidades.
 - El nombre del actor debe describir el rol que desempeña.
 - Existen Actores Principales y Actores Secundarios.



Modelado de requerimientos: Casos de uso

- Identificando los **Casos de Uso**

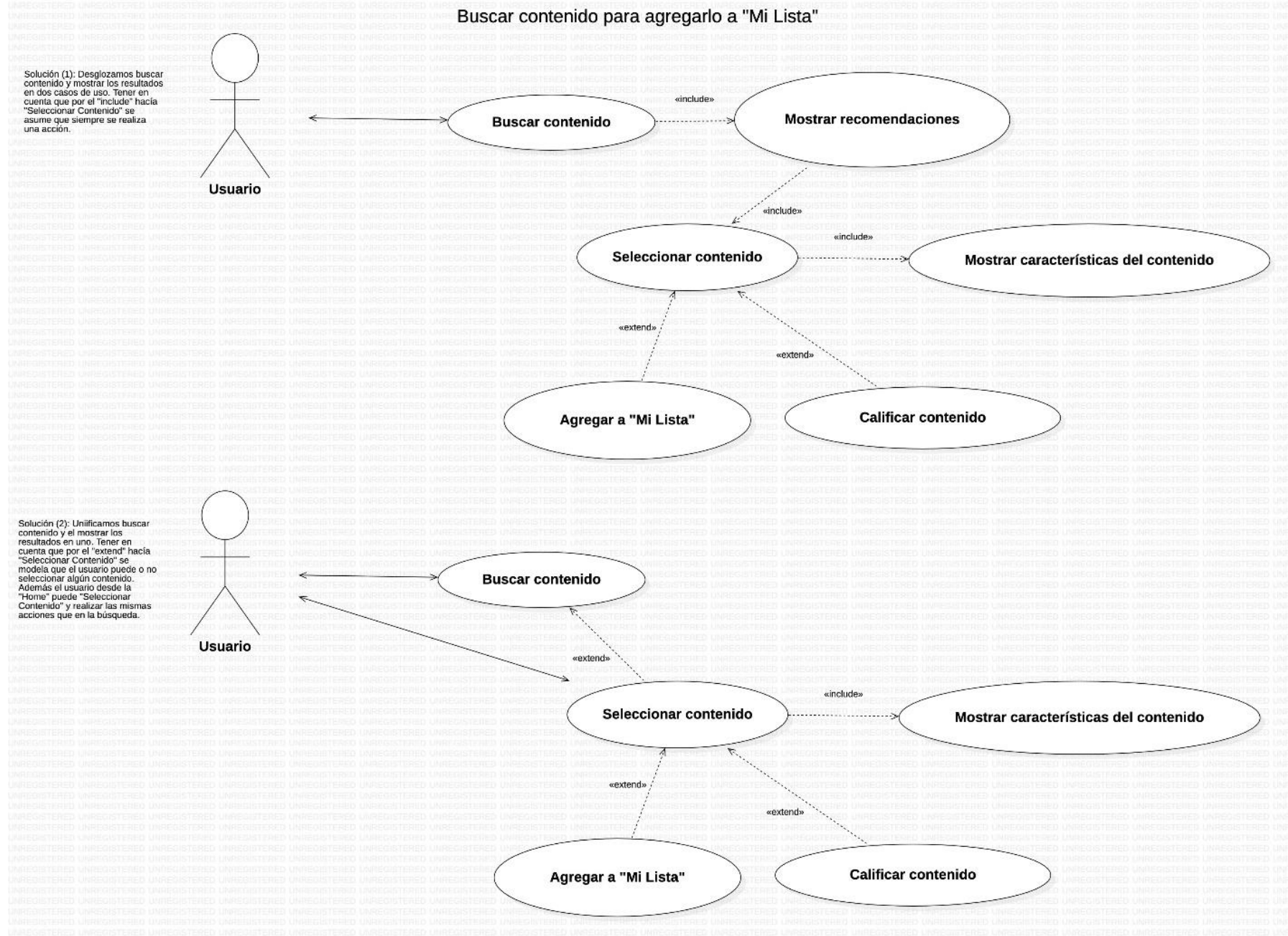
- Describen la funcionalidad y requerimientos del sistema.
- Deben ser descriptos con una acción en infinitivo al principio.
- Deben ser equilibrados entre el tamaño y su descripción.
- Evitar CU muy minimalista ó abarcativos.
- Un requerimiento o funcionalidad se puede descomponer y luego relacionar para formar un todo.



Modelado de requerimientos: Casos de uso

- Identificando las **Relaciones**

- **Include:** debe ejecutarse siempre y ayudan a complementar la función de quién lo llama.
- **Extend:** agrega funcionalidad a otro caso de uso, que puede o no ejecutarse siempre, es optativo.
- **Generalización:** permite especificar comportamiento partiendo de la funcionalidad base.



Aplico la tecnica de casos de uso

- Actores:
 - Usuario Cliente
 - Gestor de contenido
 - Gestor de pagos
- Casos de uso:
 - Subsistema de Contenido
 - Buscar contenido
 - Agregar a mi lista
 - Calificar contenido
 - ...
 - Subsistema de cuentas
 - Crear Cuenta
 - Crear perfil
 - ...
 - Subsistema de Pagos
 - Genera facturacion
 - Registrar Pago
 - ...

Objetos o conceptos del dominio

- Contenido
- Serie
- Pelicula
- Actores
- Lista personalizada
- Usuario
- Pago

Ejercicio grupal

- En grupos trabajen sobre el requerimiento de Spotify: “Armar una playlist a partir de música que busco”
 - Identificar actores.
 - Identificar casos de uso.
 - Identificar relaciones entre actores y casos de uso.
 - Realizar el modelado de casos de uso.

