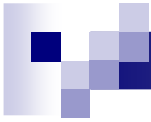


Tema 1

Requisitos

Proyecto de Análisis y Diseño de Software
2º Ingeniería Informática
Universidad Autónoma de Madrid



Indice

■ Introducción.

- Fases y Ciclo de Vida del Software.

■ Requisitos

- Técnicas de Captura de Requisitos.
- Técnicas de Representación de Requisitos.

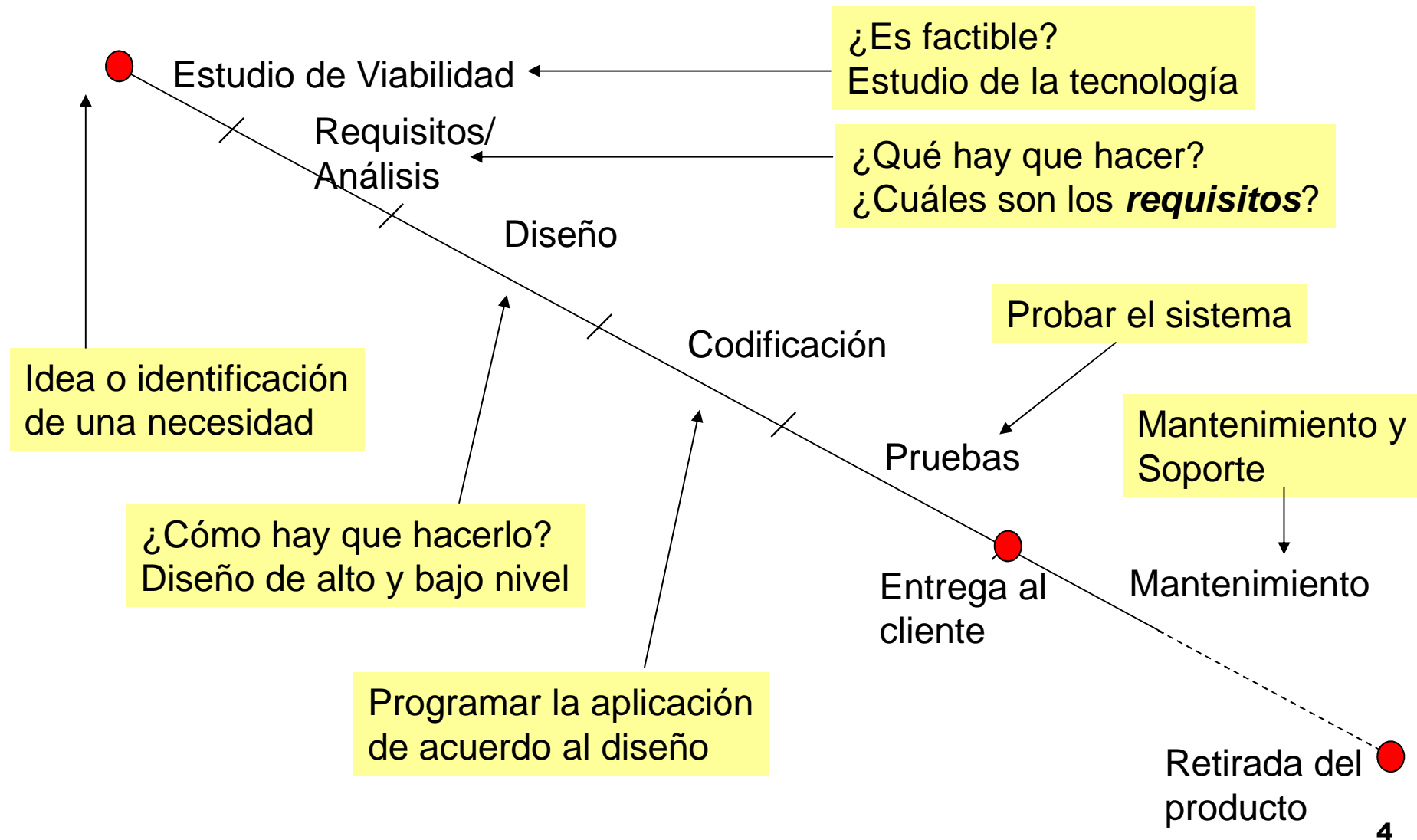


Fases en el desarrollo de Software

Ciclo de Vida

- Construir Software no consiste sólo en programar
- Fases adicionales: estudio de viabilidad, requisitos, análisis, diseño, codificación, pruebas, mantenimiento.
- Todas estas fases se desarrollan conforme a un plan de proyecto.
- Similitud con otras disciplinas:
 - construir un coche no consiste sólomente en soldar chapa y apretar tornillos,
 - construir una casa no es sólo poner ladrillos.
 - ...

Ciclo de Vida del Software

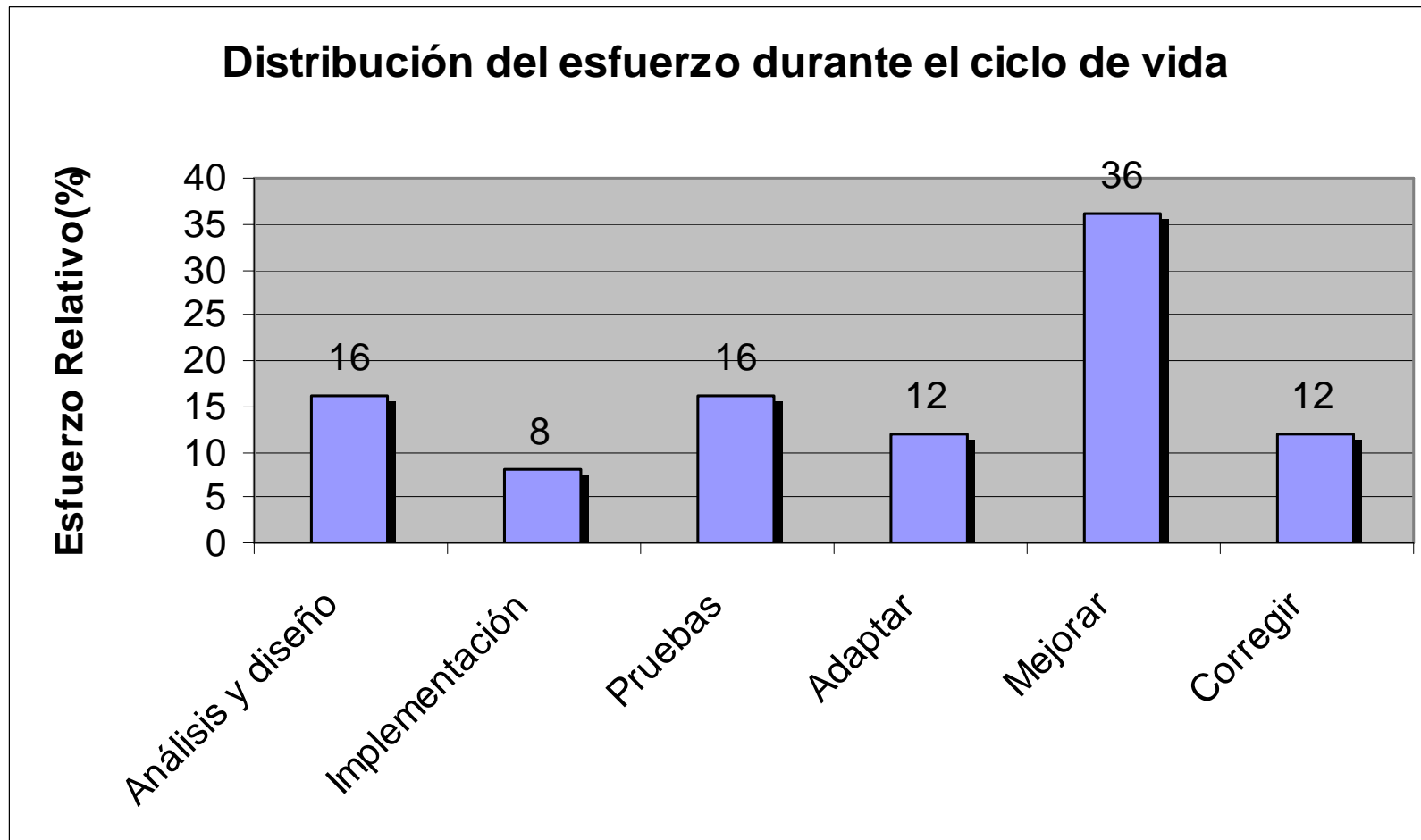


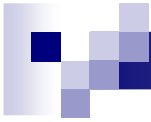


Modelos de ciclo de vida

- Las fases no necesariamente se llevan a cabo de manera lineal.
- Iteraciones:
 - Iteraciones de análisis/diseño.
- Incrementos:
 - Realizar la parte más básica de la aplicación primero.
 - Incrementos con funcionalidad secundaria.

Ciclo de Vida del Software





Indice

■ Introducción.

- ☐ Fases y Ciclo de Vida del Software.

■ Requisitos

- ☐ Técnicas de Captura de Requisitos.
- ☐ Técnicas de Representación de Requisitos.



Requisitos

- Características que ha de tener el software para que sea aceptado por el cliente.
- **Funcionales:**
 - Características útiles para el usuario.
- **No Funcionales:**
 - Operacionales (ej. recuperación, back-ups)
 - Seguridad (ej. niveles de acceso, protección)
 - Mantenibilidad y Portabilidad (ej. ejecutable en Windows y Linux)
 - Recursos (ej. memoria, almacenamiento)
 - Rendimiento (ej. tiempo de respuesta, nº usuarios)
 - Interfaz y Usabilidad (ej. tamaño de los elementos en pantalla)
 - Fiabilidad y Verificación (sobre situaciones anómalas o de error)
 - ...



Análisis de Requisitos

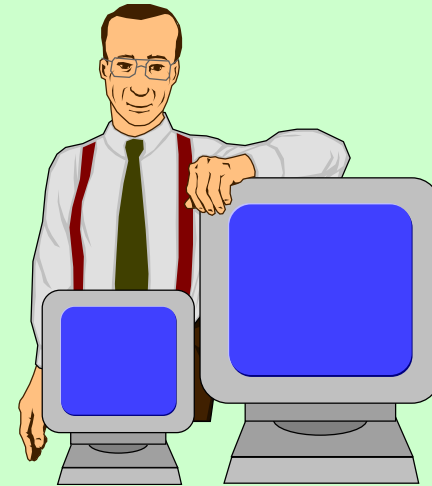
- Fase en la que hay que responder a las preguntas:
 - ☐ ¿Qué hay que hacer?
 - ☐ ¿Qué funcionalidad hay que implementar?
 - ☐ ¿Cuáles son los requisitos no funcionales (rendimiento, fiabilidad, etc)?

“Análisis del problema y especificación completa del comportamiento externo que se espera del sistema software que se va a construir, así como de los flujos de información y control.”

Análisis de Requisitos



Los clientes y usuarios plantean el problema actual, el resultado que esperan obtener y las condiciones que esperan.



El ingeniero del software pregunta, analiza, asimila y presenta la solución adecuada.

Salida: Documento de especificación de requisitos



Análisis de Requisitos

Importancia

- Un programa que resuelve un problema incorrecto no es útil para el cliente.
- Entender qué quiere el cliente antes de empezar a construir el software.
- Si falta algún requisito:
 - La aplicación final será de menor calidad.
 - El cliente estará descontento con la aplicación.
 - Habrá que añadirlo en la fase de mantenimiento: coste exponencial.
- Si un requisito es incorrecto:
 - Detectarlo cuanto antes (coste exponencial según avanza el proyecto).
- Si la aplicación no cumple con los requisitos:
 - El cliente no la aceptará.
 - Modificarla en la fase de pruebas es muy costoso.



Análisis de Requisitos

Tareas

- Captura de requisitos.
 - Identificar los requisitos que se obtienen de los usuarios y clientes.
- Análisis del problema y de los requisitos.
 - Razonar sobre los requisitos, combinar requisitos relacionados, establecer prioridades entre ellos, determinar su viabilidad, etc.
- Representación (modelización).
 - Registrar los requisitos de alguna forma, incluyendo lenguaje natural, lenguajes formales, modelos, maquetas, etc.
- Validación.
 - Examinar inconsistencias entre requisitos, determinar la corrección, ambigüedad, etc. Establecer criterios para asegurar que el software reúna los requisitos cuando se haya producido. El cliente, usuario y desarrollador se deben poner de acuerdo.



Captura de Requisitos

- También llamada Elicitación, Educción de requisitos.

- Técnicas:
 - ☐ Entrevistas.
 - ☐ Brainstorming.
 - ☐ JAD (Joint Application Design).



Captura de Requisitos

1. Análisis del problema.
2. Identificar usuarios potenciales.
3. Identificar fuentes relevantes de conocimiento/información.
4. Recopilar información y hechos.
5. Preparar y preguntar cuestiones concisas y directas.
6. Analizar la información obtenida.
7. Evaluar hallazgos (en función de coste, técnicas).
8. Priorizar.
9. Comprobar el resultado con los usuarios.
10. Sintetizar la información en forma de especificaciones.
11. Determinar aspectos no analizados todavía.



Proyecto

- Capturar los requisitos del proyecto que se desarrollará a lo largo del curso.
- Entregar un documento con el formato que puedes encontrar en Moodle.



Representación de Requisitos

- **Análisis estructurado.**
 - Técnicas de análisis orientadas a datos.
 - Técnicas de análisis orientadas a funciones.
 - Técnicas de análisis orientadas a estados.
- **Análisis orientado a objetos.**
- **Lenguajes formales.**
- **Maquetas.**

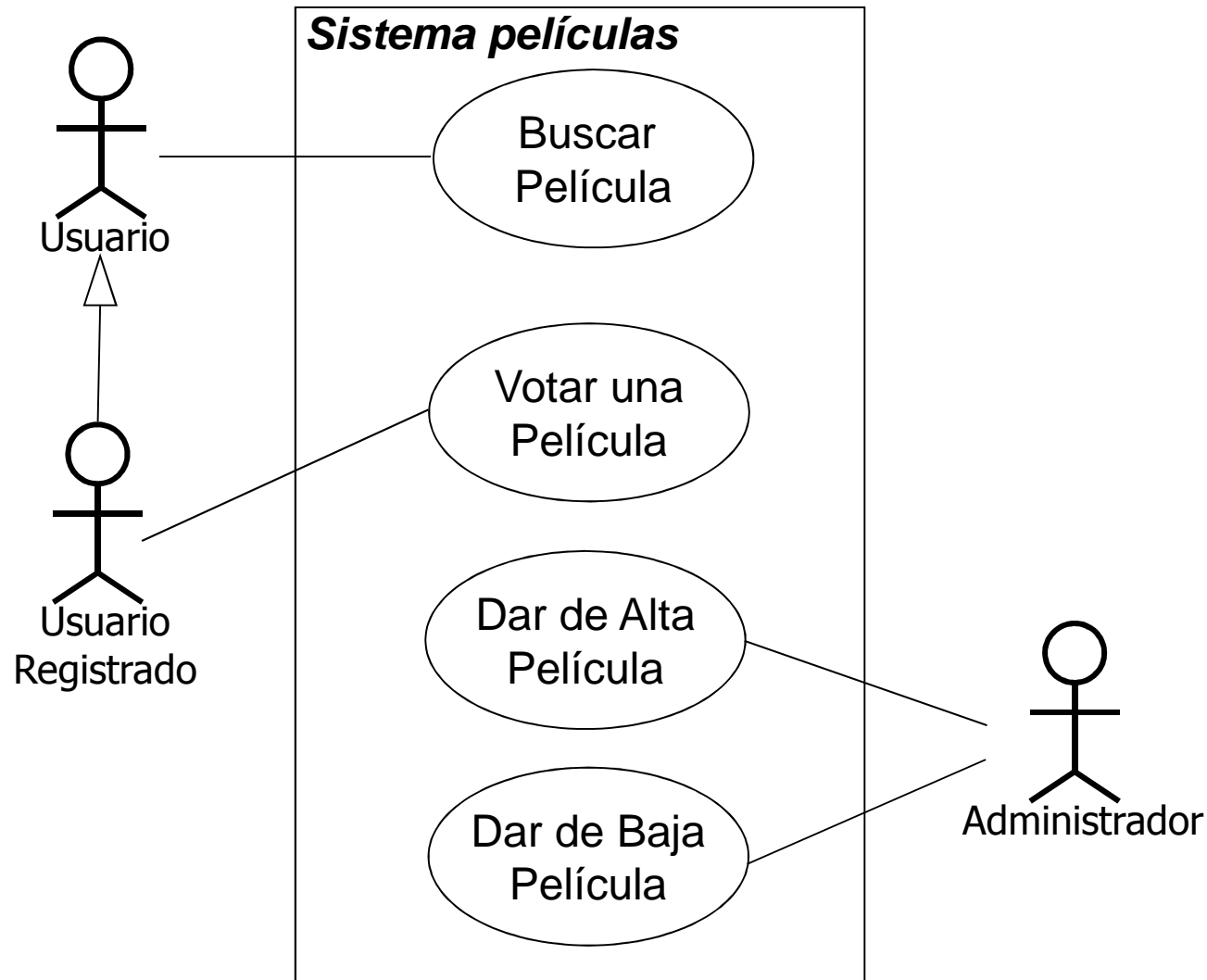


Análisis Orientado a Objetos

- Casos de Uso: conjunto de escenarios que describen distintas formas de usar el software, desde el punto de vista de cada tipo de usuario.
- Escenarios: Secuencias de interacciones que describen condiciones de éxito o fracaso (ej. errores)
- Actores: Elementos activos externos (usuarios, otro sistema) que interacciona con el sistema.

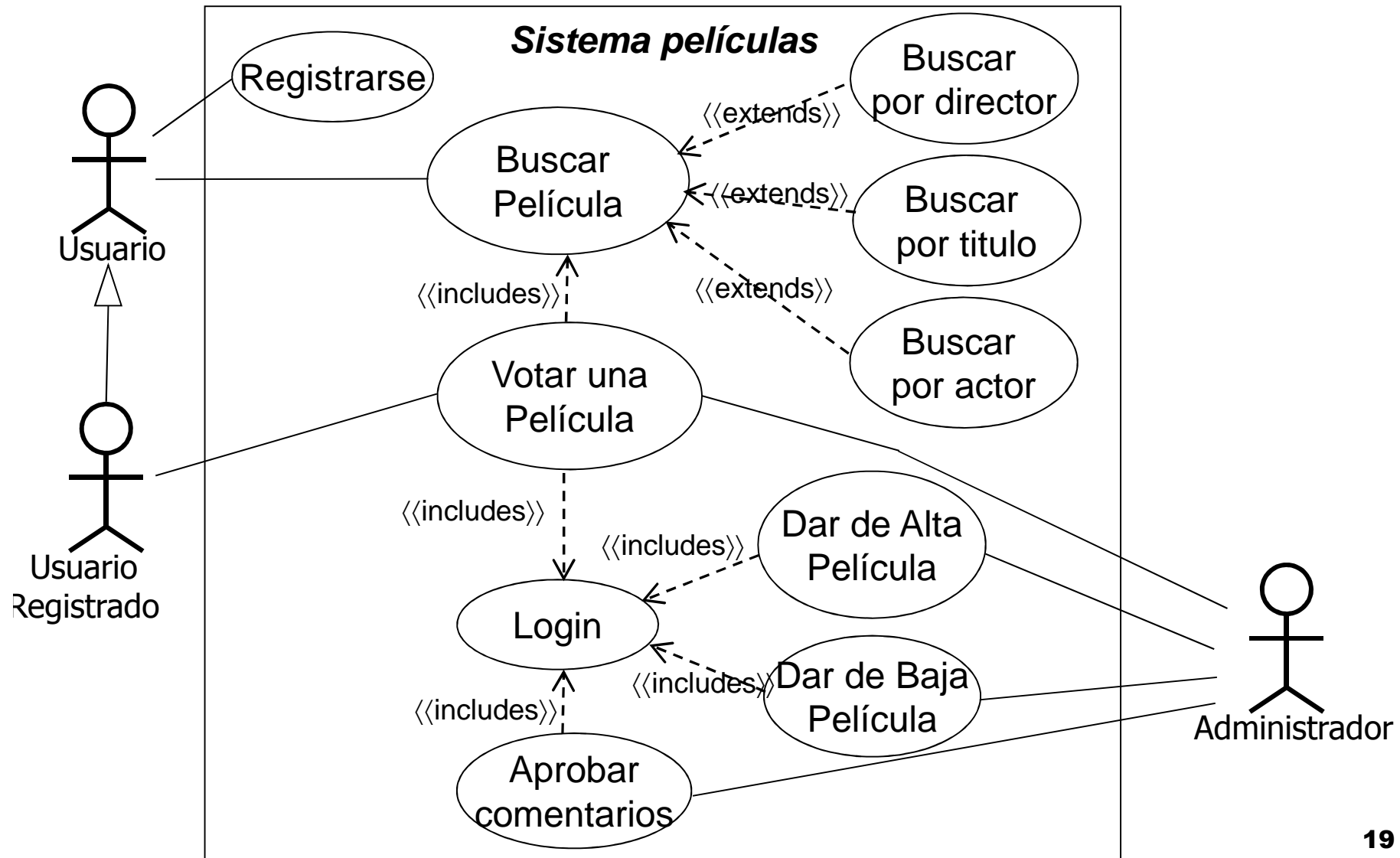
Casos de Uso

Un sistema on-line de votación de películas



Casos de Uso

Relaciones entre casos de uso





Casos de Uso

CASO DE USO 1: Votar una Película

Actor Primario: Usuario registrado.

Interesados y Objetivos:

- **Usuario Registrado:** Quiere introducir una evaluación y/o un comentario para una película que ha buscado previamente.
- **Administrador:** Quiere revisar los comentarios de los usuarios, para aprobarlos o vetarlos.

Precondiciones: El usuario se ha identificado y autenticado, ha buscado una película.

Garantía de éxito (Postcondiciones): Se registra el voto en el sistema, así como el comentario (si lo hay). Se recalcula la puntuación media de la película. Se manda una notificación al administrador, para que apruebe (o no) el comentario.

Escenario principal de Exito:

1. El usuario selecciona “valorar película”.
2. El usuario introduce una valoración entre 0 y 10.
3. El sistema guarda dicha valoración y actualiza la media de puntuación.
4. El sistema ofrece la posibilidad de introducir un comentario.
5. El usuario introduce un comentario.
6. El sistema guarda el comentario y notifica al administrador para su aprobación.



Casos de Uso

Extensiones (flujos alternativos):

5a. El usuario no introduce un comentario

5a.1. El sistema no guarda el comentario ni notifica al administrador.

Requisitos Especiales:

- Respuesta y actualización rápidas (<0.1 seg) al introducir la valoración.
- Posibilidad de internacionalización del nombre e información de las películas, así como diferentes versiones de la aplicación en distintos idiomas.

Lista de variaciones de tecnología y datos:

- Se puede plantear la posibilidad de subir documentos (pdf, word) con los comentarios.

Frecuencia de ocurrencia:

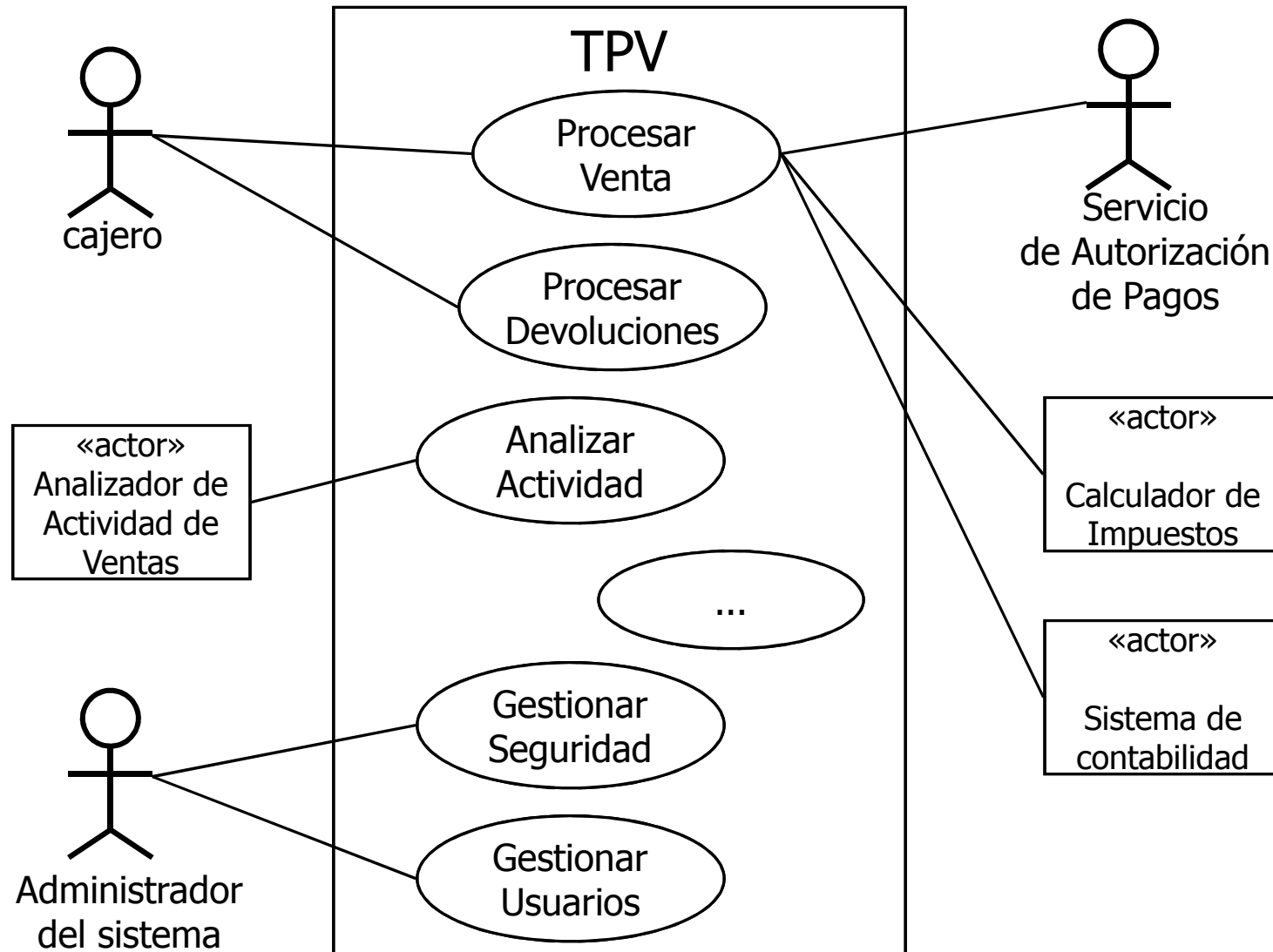
- Muy alta, del orden de miles de usuarios concurrentes.

Temas abiertos:

- Posibilidad de valorar distintos aspectos de la película (guión, dirección, actores) en el paso 2.
- ¿Se podrían obtener comentarios y valoraciones de manera automática de diversos medios, como por ejemplo críticas en periódicos o revistas especializadas?

Otro ejemplo: Terminal de Punto de Venta

Representación: Casos de Uso





Otro ejemplo: Terminal de Punto de Venta

Representación: Casos de Uso

CASO DE USO 1: Procesar venta

Actor Primario: Cajero.

Precondiciones: El cajero se ha identificado y autenticado.

Garantía de éxito (Postcondiciones): Se registra la compra en el sistema. Se calcula el impuesto aplicable. Se actualizan los sistemas de inventario y de contabilidad. Se registran las comisiones. Se genera un recibo. Se registran las aprobaciones de pago por tarjeta.

Escenario principal de Exito:

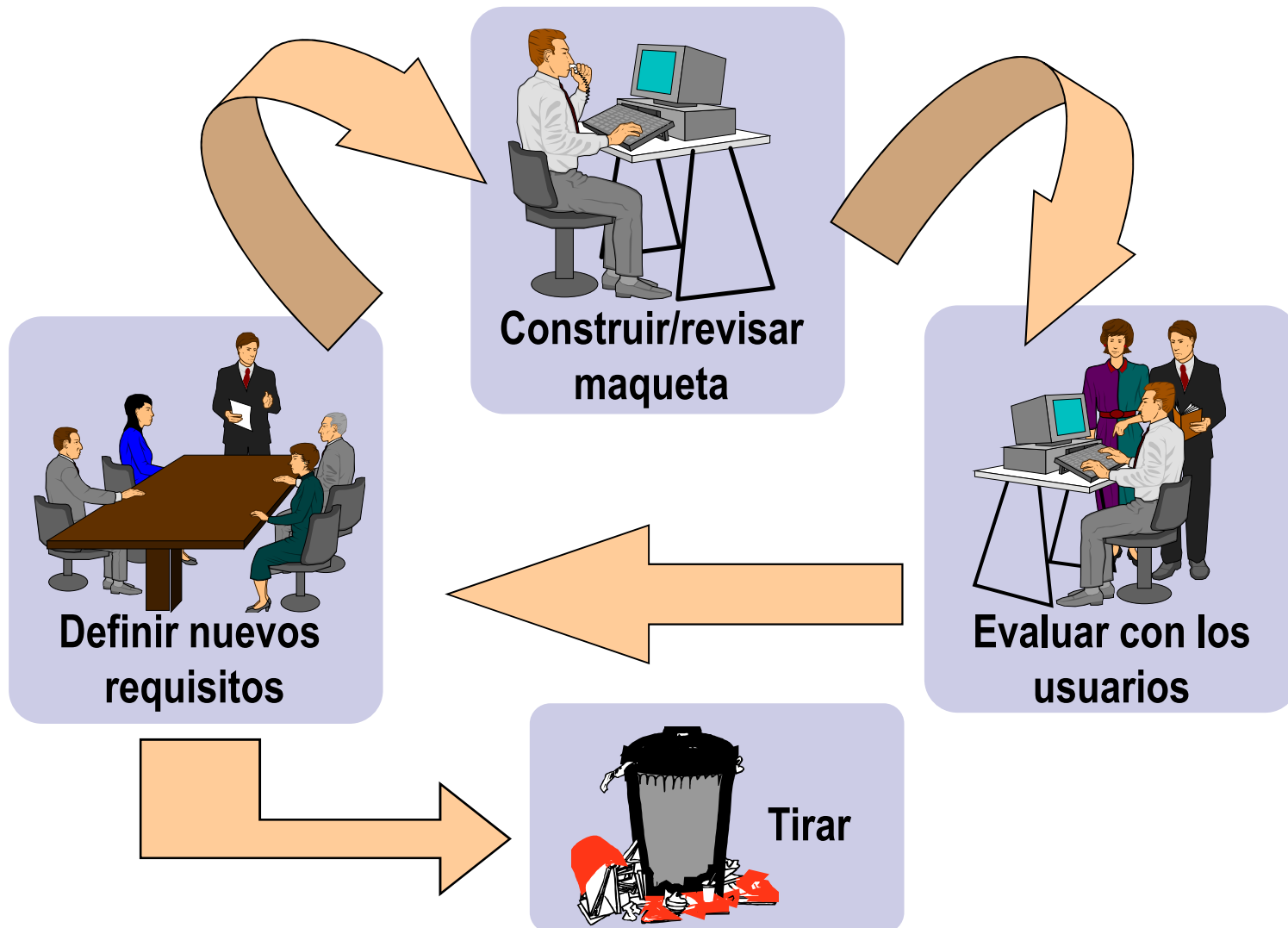
1. Llega un cliente al TPV con bienes o servicios que comprar.
 2. El cajero comienza una nueva compra.
 3. El cajero introduce un identificador de producto.
 4. El sistema registra el elemento y presenta una descripción del mismo, su precio y total actual. Se calcula el precio de una lista de reglas.
El cajero repite los pasos 3-4 hasta que no hay más elementos.
 5. El sistema presenta el total con los impuestos calculados.
 6. El cajero le dice el total al cliente, y le pide que pague.
 7. El cliente paga y el sistema procesa el pago.
 8. El sistema registra la venta completada y manda la información a los sistemas externos de inventario y contabilidad.
 9. El sistema genera el recibo.
 10. El cliente se va.
- (además hay que especificar excepciones y flujos alternativos)

Maquetas

- Captura de requisitos en forma de interfaces que permitan un mejor entendimiento con el usuario.
- Desde programas de dibujo (powerpoint, visio) hasta aplicaciones especializadas (ej.: Mockupscreens, Balsamiq, etc).



Maquetas: Ciclo de vida



Ejemplo

Ventana Inicial

The Internet Movie Database (IMDb) - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://www.imdb.es/

Más visitados Comenzar a usar Firef... Últimas noticias Personal Docente e In...

INFO-PELIS

Buscador de películas

Búsqueda: Luis García Berlanga

☒ Director
☐ Título
☐ Actor

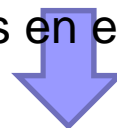
Inicia una sesión

Usuario:

Clave:

¿Usuario nuevo?
[Regístrate](#)

Pulsamos en el botón Buscar



Ejemplo

Resultado de una búsqueda



The screenshot shows a Mozilla Firefox browser window with the title 'The Internet Movie Database (IMDb) - Mozilla Firefox'. The address bar shows 'http://www.imdb.es/'. The page content is divided into two main sections.

Left Section (Login):

Inicia una sesión
Usuario:

Clave:

¿Usuario nuevo?
[Regístrate](#)

Right Section (Search Results):

INFO-PELIS

24 películas encontradas con director Luis García Berlanga

1. El sueño de la maestra (2002)	5,9/10 41 votos
2. París Tombuctú (1999)	5,3/10 247 votos
3. Blasco Ibáñez (1997)	4,2/10 21 votos
4. Todos a la cárcel (1993)	5,7/10 170 votos
5. Moros y Cristianos (1987)	5,5/10 133 votos
6. La vaquilla (1985)	7,3/10 413 votos
7. Nacional III (1982)	6,5/10 94 votos
8. Patrimonio Nacional (1981)	6,7/10 146 votos
....



Algunas herramientas para hacer maquetas

- De pago, pero con versión de evaluación:
 - Balsamiq: <http://www.balsamiq.com/products/mockups>
 - Mockupscreens: <http://mockupscreens.com/>

- Gratis:
 - Lumzy (on-line): <http://lumzy.com/app>
 - Mockingbird (on-line): <http://gomockingbird.com/mockingbird/>
 - Pencil: <http://pencil.evolus.vn/>
 - Cacao (on-line): <https://cacao.com/>

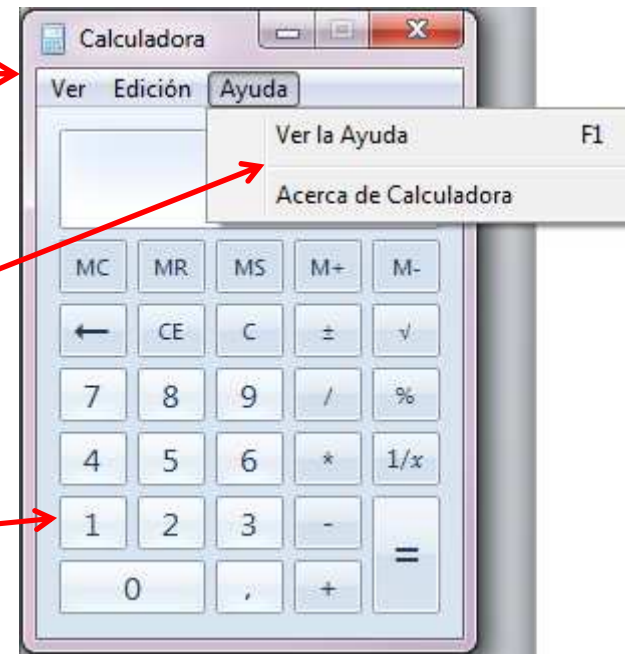
Interfaces gráficas de usuario

Algunos elementos gráficos comunes

■ Ventana

■ Menú desplegable

■ Botones



Interfaces gráficas de usuario

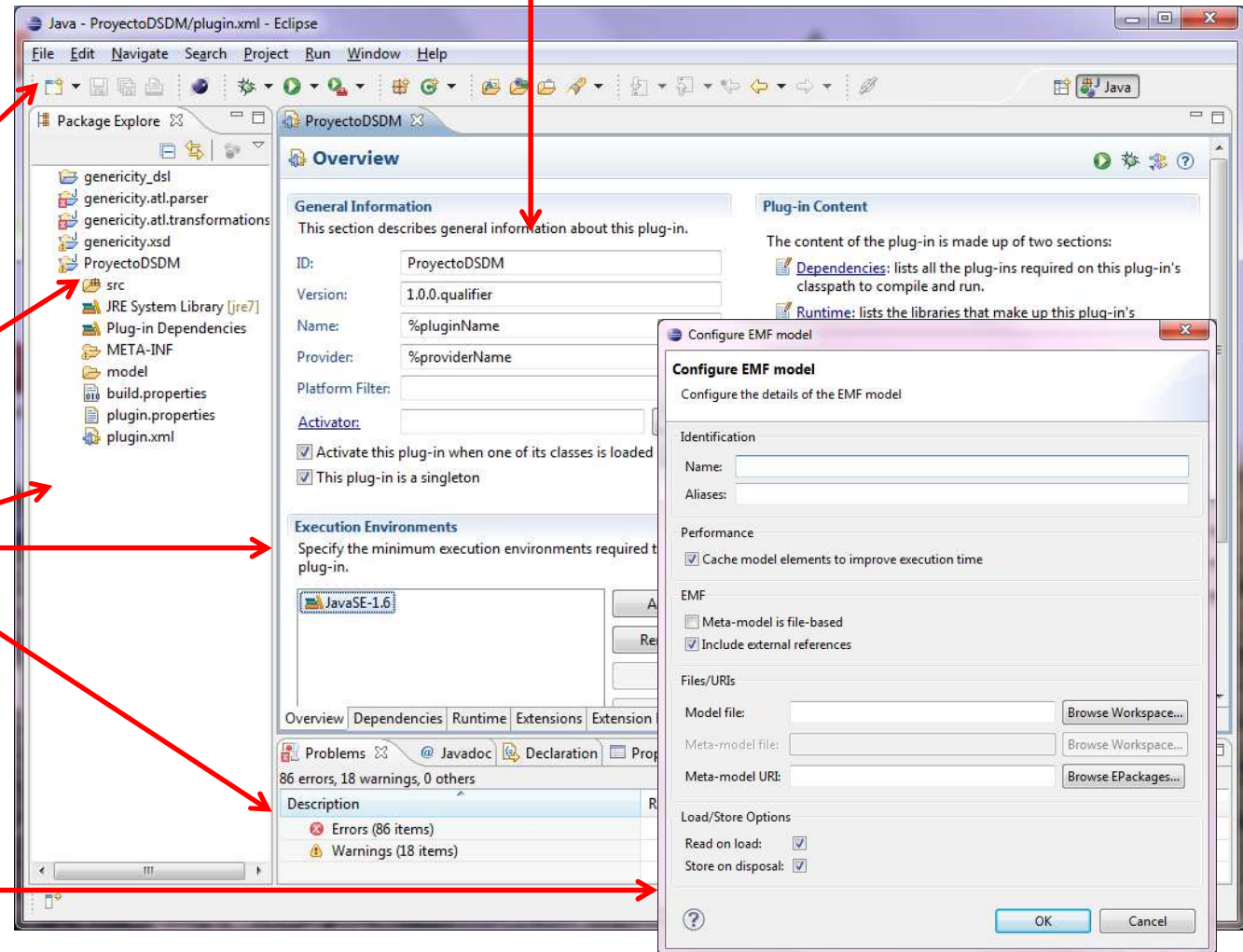
Algunos elementos gráficos **Formulario** comunes

Barra de herramientas

Árbol

Paneles

Ventana de diálogo



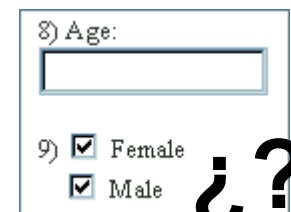
Usabilidad

Algunas normas de usabilidad

- Consistencia en la presentación de información (nomenclatura, formato, colores, etc).
- Minimizar la posibilidad de error: usar el control adecuado.

- ☐ Usar una lista en vez de un campo de texto para elegir de entre una serie cerrada de opciones.
- ☐ Usar *checkboxes* para elecciones binarias.
- ☐ Usar *radiobuttons* para elecciones excluyentes.

☒ This plug-in is a singleton



8) Age:

9) ☒ Female
☒ Male

¿?

- Minimizar la necesidad de memorización del usuario:
 - ☐ No hacerle recordar información en una pantalla que tenga que introducir en otra.



Usabilidad

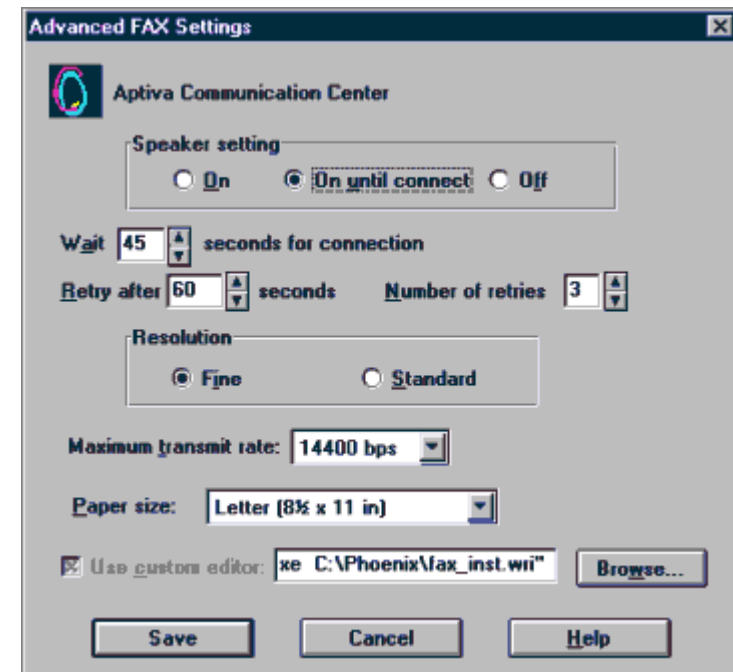
Las 8 reglas de oro...

- Consistencia.
- Usabilidad Universal.
- Feedback informativo.
- Diseñar adecuadamente diálogos y secuencias de acciones (ej. tipo wizard).
- Prevenir errores.
- Permitir deshacer acciones.
- Que el usuario controle la aplicación (y no al revés).
- Reducir la necesidad de memorizar.

Usabilidad

Ejemplos de malas prácticas

- Es difícil seguir el flujo de información.
 - Alinear las opciones verticalmente.
 - Asegurarse de que los campos son suficientemente largos para mostrar la información.
 - Agrupar las etiquetas a la izquierda y los campos en un grupo.
 - Asignar atajos a letras significativas ('a' para Wait? 'l' para Fine?).



Usabilidad

Ejemplos de malas prácticas



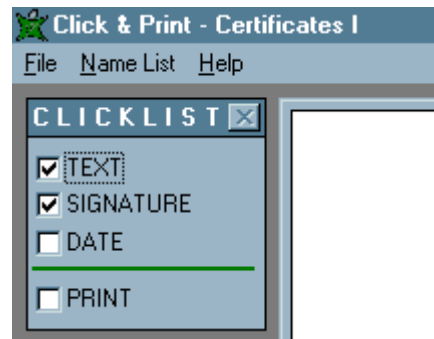
Select your State

<input type="radio"/> Alabama	<input type="radio"/> Kansas
<input type="radio"/> Alaska	<input type="radio"/> Kentucky
<input type="radio"/> Arizona	<input type="radio"/> Maine
<input type="radio"/> Arkansas	<input type="radio"/> Maryland
<input type="radio"/> California	<input type="radio"/> Massachusetts
<input type="radio"/> Colorado	<input type="radio"/> Michigan
<input type="radio"/> Connecticut	<input type="radio"/> Minnesota
<input type="radio"/> Delaware	<input type="radio"/> Mississippi
<input type="radio"/> Florida	<input type="radio"/> Missouri
<input type="radio"/> Georgia	<input type="radio"/> M

- ¿Podemos usar otro elemento mejor que este?
- Sí: Una lista desplegable (combo-box).

Usabilidad

Ejemplos de malas prácticas

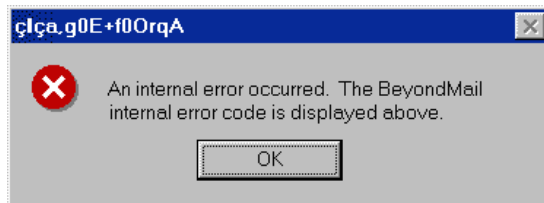


- Al seleccionar “PRINT” se inicia la impresión.
- Mala práctica, usar comandos (un botón, una opción de menú) para iniciar acciones.

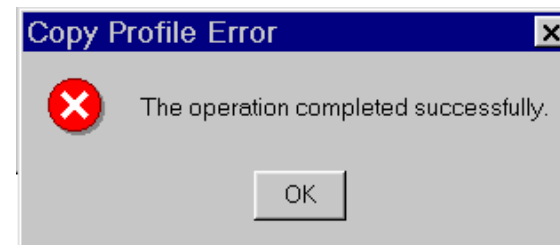
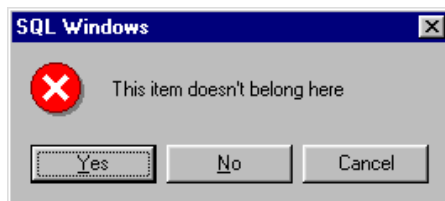
Usabilidad

Mensajes de error

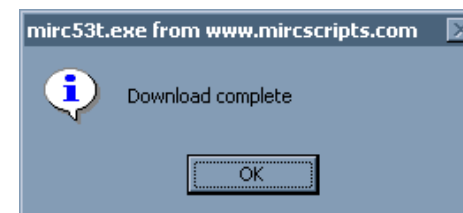
- Mostrar texto informativo (válido para el usuario) en los mensajes de error.



- No confundir al usuario con botones innecesarios o mensajes confusos o contradictorios.



- No presentar cuadros de diálogo innecesarios (que ralenticen o interrumpen el flujo de trabajo).



Widgets Java básicos

■ Componentes de acción

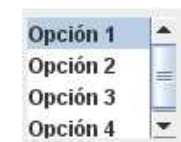
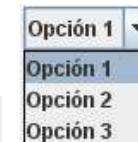


■ Entrada de datos básica



☐ Seleccionar opción

☐ Seleccionar opción



■ Entrada de datos complejos



■ Presentación estructurada

Id	Tipo	Nombre arti...	Precio	Stock
dato1.1	dato1.2	dato1.3	dato1.4	dato1.5
dato2.1	dato2.2	dato2.3	dato2.4	dato2.5
dato3.1	dato3.2	dato3.3	dato3.4	dato3.5
dato4.1	dato4.2	dato4.3	dato4.4	dato4.5
dato5.1	dato5.2	dato5.3	dato5.4	dato5.5
dato6.1	dato6.2	dato6.3	dato6.4	dato6.5
dato7.1	dato7.2	dato7.3	dato7.4	dato7.5
dato8.1	dato8.2	dato8.3	dato8.4	dato8.5
dato9.1	dato9.2	dato9.3	dato9.4	dato9.5
dato10.1	dato10.2	dato10.3	dato10.4	dato10.5





Proyecto

- Realiza el diagrama de casos de uso.
- Describe los 3 casos de uso que consideres más relevantes, con el formato mostrado en estas transparencias.
- Realiza una maqueta de la aplicación.
- Utiliza el formato de documento que puedes encontrar en Moodle.



Bibliografía

■ Bibliografía básica:

- Software engineering a practitioner's approach, 7ªed. Roger Pressman. McGraw Hill Higher Education, 2010. INF/681.3.06/PRE. También disponible en castellano.
- Software engineering, 9ª ed. Addison Wesley. Ian Sommerville. INF/681.3.06/SOM. También disponible en castellano.

■ Bibliografía recomendada:

- Writing Effective Use Cases. A. Cockburn. Addison-Wesley Professional, 2000. INF/D5000/COC.
- 830-1998 IEEE Recommended Practice for Software Requirements Specifications (Disponible en IEEE Xplore desde una IP de la escuela).
- Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition). Addison Wesley. Shneiderman, Plaisant , Cohen, Jacobs.