

UNIDAD DIDÁCTICA I

TEMA 2 – Inicio

1. INICIO

El objetivo de la etapa de inicio es hacer la investigación justa para formar una opinión racional y justificable del propósito global y la viabilidad del nuevo sistema potencial, y decidir si merece la pena invertir en un estudio más profundo. La fase de inicio debería ser relativamente corta con una duración de una a unas pocas semanas.

En el desarrollo iterativo y el UP, los planes y estimaciones de la fase de inicio no deben considerarse fiables, proporcionan una percepción del orden de magnitud del grado de esfuerzo, para ayudar en la decisión de continuar o no.

El propósito de la fase de inicio es establecer una visión común inicial de los objetivos del proyecto, determinar si es viable y decidir si merece la pena llevar a cabo algunas investigaciones serias en la fase de elaboración. Si se ha decidido de antemano que el proyecto se hará y es claramente viable, la fase de inicio será especialmente breve. Podría incluir los primeros talleres de requisitos, planificación de la primera iteración y, entonces, cambiar a la elaboración.

1.1. Artefactos en la fase de inicio

Tabla 4.1. Ejemplo de artefactos de la fase de inicio.

<i>Artefacto</i>	<i>Comentario</i>
Visión y Análisis del Negocio	Describe los objetivos y las restricciones de alto nivel, el análisis del negocio y proporciona un informe para la toma de decisiones.
Modelo de Casos de Uso	Describe los requisitos funcionales y los no funcionales relacionados.
Especificación Complementaria	Describe otros requisitos.
Glosario	Terminología clave del dominio.
Lista de Riesgos & Plan de Gestión del Riesgo darles respuesta.	Describe los riesgos del negocio, técnicos, recursos, planificación, y las ideas para mitigarlos o darles respuesta.
Prototipos y pruebas-de-conceptos	Para clarificar la visión y validar las ideas técnicas.
Plan de Iteración	Describe qué hacer en la primera iteración de la elaboración.
Fase Plan de & Plan de Desarrollo de Software	Estimación de poca precisión de la duración y esfuerzo de la fase de elaboración. Herramientas, personas, formación y otros recursos.
Marco de Desarrollo	Una descripción de los pasos del UP y los artefactos adaptados para este proyecto. El UP siempre se debe adaptar al proyecto.

Estos artefactos sólo se completan parcialmente en esta fase, y se refinarán en iteraciones posteriores (los artefactos deberían ser ligeros). En la fase de inicio se podrían realizar algunas tareas de programación con el objeto de crear prototipos de “pruebas de conceptos”, para clarificar requisitos mediante prototipos orientados a la interfaz de usuario.

Lo importante de un artefacto no es el documento o el diagrama en sí mismo, sino el pensamiento, análisis y disposición activa (y entonces se registra). Los artefactos del UP de los proyectos anteriores se pueden utilizar en otros posteriores (lo que da muchas similitudes entre los proyectos en cuanto a los artefactos de riesgos, gestión del proyecto, pruebas y entorno). Todos los proyectos UP organizarán los artefactos de la misma manera, con los mismos nombres, lo que simplifica la búsqueda de artefactos reutilizables.

1.2. No es fase de inicio si...

- La duración es mayor de “unas pocas” semanas.
- Se intenta definir la mayoría de los requisitos.
- Se espera que los planes y estimaciones sean fiables.
- Se define la arquitectura.

- No hay artefacto de Análisis del Negocio o Visión.
- No se identificaron la mayoría de los nombres de los casos de uso y los actores.
- Se escribieron todos los casos de uso en detalle.
- Ninguno de los casos de uso se escribió en detalle.

2. COMPRENSIÓN DE LOS REQUISITOS

Los **requisitos** son capacidades y condiciones con las cuales debe ser conforme el sistema. El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar (registrar) lo que se necesita realmente, de manera que tenga un significado claro para el cliente y los miembros del equipo de desarrollo. El UP fomenta un conjunto de buenas prácticas, una de las cuales es la *gestión de requisitos*: “un enfoque sistemático para encontrar, documentar, organizar y seguir la pista de los requisitos cambiantes de un sistema” (*cambiantes* → el UP acepta el cambio en los requisitos como un motor fundamental del proyecto, *encontrar* → incitar mediante técnicas como escritura de casos de uso y talleres de requisitos).

2.1. Tipos de requisitos

En el UP, los requisitos se clasifican con el modelo FURPS+:

- **Funcional (Functional)**: características, capacidades y seguridad.
- **Facilidad de uso (Usability)**: factores humanos, ayuda, documentación.
- **Fiabilidad (Reliability)**: frecuencia de fallos, capacidad de recuperación de un fallo y grado de previsión.
- **Rendimiento (Performance)**: tiempos de respuesta, productividad, precisión, disponibilidad, uso de los recursos.
- **Soporte (Supportability)**: adaptabilidad, facilidad de mantenimiento, internacionalización, configurabilidad.

El '+' en FURPS+ indica requisitos adicionales:

- **Implementación**: limitación de recursos, lenguajes y herramientas, hardware...
- **Interfaz**: restricciones impuestas para la interacción con sistemas externos.
- **Operaciones**: gestión del sistema en su puesta en marcha.
- **Empaquetamiento**
- **Legales**: licencias, etc.

Algunos de estos requisitos se denominan **atributos de calidad** o **requisitos de calidad**, que comprenden facilidad de uso (*usability*), fiabilidad (*reliability*), rendimiento (*performance*) y soporte (*supportability*). Lo normal es dividir los requisitos en **funcionales** (comportamiento) y **no funcionales** (lo demás).

Los requisitos funcionales se estudian y recogen en el Modelo de Casos de Uso y en la lista de características del sistema del artefacto Visión. Los otros requisitos se pueden recoger en los casos de usos con los que están relacionados, o en el artefacto Especificación Complementaria. El artefacto Visión resume los requisitos de alto nivel. El Glosario agrupa y clarifica los términos que se utilizan en los requisitos. El Glosario en el UP también comprende el concepto de **diccionario de datos**, que reúne los requisitos relacionados con los datos, como reglas de validación, valores aceptables, etc. Los requisitos de calidad influyen fuertemente en la arquitectura de un sistema.

3. MODELO DE CASOS DE USO: ESCRITURA DE REQUISITOS EN CONTEXTO

Los casos de uso son un mecanismo para descubrir y registrar los requisitos. El UP define el **Modelo de Casos de Uso** en la disciplina Requisitos, el conjunto de todos los casos de uso; es un modelo de la funcionalidad y entorno del sistema.

3.1. Objetivos

Los clientes y los usuarios finales tienen objetivos (*necesidades*) y quieren sistemas informáticos que les ayuden a conseguirlos. Los casos de uso son un mecanismo para ayudar a mantenerlo simple y entendible para todo el personal involucrado, son historias del uso de un sistema para alcanzar los objetivos. La esencia es descubrir y registrar los requisitos funcionales, mediante la escritura de historias del uso de un sistema, para ayudar a cumplir los objetivos de varias de las personas involucradas.

3.2. Casos de uso y valor añadido

Un **actor** es algo con comportamiento, como una persona (un rol), sistema informatizado u organización. Un **escenario** es una secuencia específica de acciones e interacciones entre los actores y el sistema objeto de estudio (también **instancia de caso de uso**), una historia particular del uso de un sistema, o un camino a través del caso de uso. Así, un **caso de uso** es una colección de escenarios con éxito y fallo relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo. Los casos de uso sitúan las características y funciones en un contexto orientado al objetivo, trabaja con los requisitos centrándose en cómo puede un sistema añadir valor y cumplir los objetivos.

3.3. Casos de uso y requisitos funcionales

Los casos de uso son requisitos; ante todo funcionales que indican qué hará el sistema. En términos FURPS +, los casos de uso se refieren fundamentalmente a la “F”, pero también pueden utilizarse para otros tipos. Los casos de uso son requisitos (aunque no todos los requisitos). Los casos de uso son documentos de texto, no diagramas, y el modelado de casos de uso es una acción de escribir texto. UML define un diagrama de casos de uso para ilustrar los nombres de casos de uso y actores, y sus relaciones.

3.4. Tipos de casos de uso y formatos

Casos de uso de caja negra y las responsabilidades del sistema

Los **casos de uso de caja negra** no describen el funcionamiento interno del sistema, sus componentes o diseño, sino el sistema en base a las *responsabilidades* que tiene. A través de la definición de las responsabilidades del sistema con casos de uso de caja negra, es posible especificar *qué* debe hacer el sistema (los requisitos funcionales) sin decidir *cómo* lo hará (el diseño).

Tipos de formalidad

Los casos de uso se escriben con formatos diferentes, dependiendo de la necesidad. Además del tipo de *visibilidad*, de caja negra frente a caja blanca, se escriben con varios grados de *formalidad*:

- **Breve:** resumen conciso de un párrafo, normalmente del escenario principal con éxito.
- **Informal:** formato de párrafo en un estilo informal. Múltiples párrafos que comprenden varios escenarios.
- **Completo:** se escriben con detalle todos los pasos y variaciones, y hay secciones de apoyo como precondiciones y garantías de éxito. Muestran más detalles y están estructurados y son útiles para entender en profundidad los objetivos, tareas y requisitos.

Existe un formato en dos columnas o conversacional, que establece una interacción entre los actores y el sistema. Aunque no existe un mejor formato, la clave es escribir los detalles del escenario principal de éxito y sus extensiones de alguna forma.

3.5. Secciones de un caso de uso

Elementos del prólogo

Incluye sólo los elementos que son importantes que se lean antes del escenario principal de éxito, el material de “encabezamiento” ajeno, queda al final de los casos de uso.

Actor principal: El actor principal que recurre a los servicios del sistema para cumplir un objetivo.

Personal involucrado y lista de intereses

Sugiere y delimita qué es lo que debe hacer el sistema. El caso de uso, como contrato de comportamiento, captura *todo* y *sólo* el comportamiento relacionado con la satisfacción de los intereses del personal involucrado. El punto de vista del interés del personal involucrado proporciona un procedimiento metódico y completo para el descubrimiento y registro de todos los comportamientos requeridos.

Personal involucrado e intereses:

- Cajero: quiere entradas precisas, rápidas, y sin errores de pago, ya que las pérdidas se deducen de su salario.
- Vendedor: quiere que las comisiones de las ventas estén actualizadas.
- ...

Precondiciones y garantías de éxito (postcondiciones)

Las **precondiciones** establecen lo que *siempre debe* cumplirse antes de comenzar un escenario en el caso de uso, *no* se prueban en el caso de uso, se asumen que son verdad. Normalmente, implican un escenario de otro caso de uso que se ha completado con éxito. Hay condiciones que deben ser verdad; pero no tienen un valor práctico para que se escriban y comunican suposiciones importantes de las que el escritor del caso de uso piensa que los lectores deberían ser avisados.

Las **garantías de éxito (postcondiciones)** establecen qué debe cumplirse cuando el caso de uso se completa con éxito (o algún camino alternativo) y deberían satisfacer las necesidades de todo el personal involucrado.

Precondiciones: El cajero se identifica y autentica.
Garantías de éxito (Postcondiciones): Se registra la venta. El impuesto se calcula de manera correcta. Se actualizan la Contabilidad y el Inventario. Se registran las comisiones. Se genera el recibo.

Escenario principal de éxito y pasos (Flujo Básico)

Describe el camino de éxito típico que satisface los intereses del personal involucrado. A menudo *no* incluye ninguna condición o bifurcación. El escenario recoge los pasos, que pueden ser de tres tipos:

1. Una interacción entre actores.
2. Una validación (normalmente a cargo del sistema).
3. Un cambio de estado realizado por el sistema.

El primer paso de un caso de uso no está incluido en esta clasificación, sino que indica el evento que desencadena el comienzo del escenario.

Escenario principal de éxito (o Flujo Básico):
1. El Cliente llega a un terminal PDV con mercancías para comprar.
2. El Cajero comienza una nueva venta.
3. El Cajero introduce el identificador del artículo.
4. ...
El Cajero repite los pasos 3-4 hasta que se indique.
5. ...

Extensiones (Flujos Alternativos)

Indican todos los otros escenarios tanto de éxito como de fracaso. La combinación del camino feliz y los escenarios de extensión deberían satisfacer “casi” todos los intereses del personal involucrado (algunos intereses se pueden capturar mejor como requisitos no funcionales escritos en la Especificación Complementaria). Son bifurcaciones del escenario principal de éxito y pueden ser etiquetados de acuerdo con él.

Extensiones (o Flujos Alternativos):
3a. Identificador no válido:
1. El Sistema señala el error y rechaza la entrada.
3b. Hay muchos artículos de misma categoría y tener en cuenta una única identidad del artículo no es importante (ej. 5 paquetes de hamburguesas vegetales):
1. El Cajero puede introducir el identificador de la categoría del artículo y la cantidad.

Una extensión tiene dos partes: la condición y el manejo. El manejo de la extensión se puede resumir en un paso, o incluir una secuencia. Al final del manejo de la extensión, por defecto, el escenario se une de nuevo con el escenario principal de éxito, a menos que la extensión indique otra cosa. Algunas veces, un punto de extensión particular es bastante complejo, lo que puede ser motivo para expresarlo como un caso de uso aparte.

Requisitos especiales

Si un requisito no funcional, atributo de calidad o restricción se relaciona de manera específica con un caso de uso, se recoge en el caso de uso. Incluye cualidades como rendimiento, fiabilidad y facilidad de uso, y restricciones de diseño que son obligados o se consideran probables.

Requisitos especiales:

- Interfaz de usuario con pantalla táctil en un gran monitor de pantalla plana. El texto debe ser visible a un metro de distancia.
- Tiempo de respuesta para la autorización de crédito de 30 segundos el 90% de las veces.
- Internacionalización del lenguaje del texto que se muestra.
- Reglas de negocio que se puedan añadir en ejecución en los pasos 3 y 7.

Lista de tecnología y variaciones de datos

Son decisiones o restricciones de diseño anticipadas, que en general deben evitarse; pero algunas veces son obvias o inevitables, especialmente en relación con las tecnologías de entrada/salida. También es útil registrar las variaciones en los datos que podrían capturarse en un paso particular. No debería contener múltiples pasos para representar la variación de comportamiento en diferentes casos.

Lista de tecnología y variaciones de datos

- 3a. El identificador del artículo se introduce mediante un escáner láser de código de barras (si está presente el código de barras) o a través del teclado.
- 3b. El identificador del artículo podría ser cualquier esquema de código UPC, EAN, JAN o SKU.
- 7a. La entrada de la información de la cuenta de crédito se lleva a cabo mediante un lector de tarjetas o el teclado.
- 7b. La firma de los pagos a crédito se captura en un recibo de papel. Pero en dos años, pronosticamos que muchos clientes querrán que se capture la firma digital.

3.6. Objetivos y alcance de un caso de uso

Hay casos de uso a *diferentes niveles*, dependiendo de los límites del sistema, actores y objetivos. Para el análisis de requisitos de una aplicación informática, se toman los casos de uso al nivel de **procesos del negocio elementales** (EBPs, Elementary Business Processes): una tarea realizada por una persona en un lugar, en un instante, como respuesta a un evento del negocio, que añade un valor cuantificable para el negocio y deja los datos en un estado consistente.

Aunque los casos de uso “base” de una aplicación deberían satisfacer la guía EBP, es útil crear “sub” casos de uso separados que representan subtareas, o pasos en un solo caso de uso base.

Casos de uso y objetivos

Un caso de uso de nivel EBP se denomina caso de uso de nivel de **objetivo de usuario**, para remarcar que sirve (o debería servir) para satisfacer un objetivo de un usuario del sistema, o el actor principal, lo que lleva a:

1. Encontrar los objetivos de usuario.
2. Definir un caso de uso para cada uno.

La guía EBP se puede aplicar igualmente para decidir si un objetivo o un caso de uso se encuentran a un nivel adecuado.

Objetivos y casos de uso de subfunción

El **objetivo de subfunción** es un objetivo de nivel más bajo que dan soporte a un objetivo de usuario. Sólo deberían escribirse casos de uso de manera ocasional para estos objetivos de subfunción. No es ilegal; pero no siempre es útil, ya que añade complejidad al modelo de casos de uso. El número y granularidad de los casos de uso influyen en el tiempo y la dificultad para entender, mantener y gestionar los requisitos. El motivo válido más común para representar un objetivo de subfunción como un caso de uso es cuando la subfunción se repite o es una precondition en muchos casos de uso de nivel de objetivos de usuario.

Objetivos y casos de uso compuestos

Los objetivos son compuestos desde el nivel de empresa con objetivos intermedios a nivel de uso de la aplicación y objetivos de subfunción dentro de las aplicaciones. Los casos de uso se pueden escribir a niveles diferentes para satisfacer estos objetivos y pueden estar compuestos de casos de uso de nivel inferior.

3.7. Descubrimiento de actores principales, objetivos y casos de uso

Los casos de uso se definen para satisfacer los objetivos de usuario de actores principales, luego hay que:

1. Elegir los límites del sistema.

2. Identificar los actores principales: aquellos que tienen objetivos de usuario que se satisfacen mediante el uso de servicios del sistema.
3. Para cada uno, identificar sus objetivos de usuario.
4. Definir los casos de uso que satisfagan los objetivos de usuario.

Paso 1: elegir el límite del sistema

Si no está clara la definición de los límites, se puede aclarar definiendo lo que está fuera (actores principales y de apoyo).

Pasos 2 y 3: identificar los actores principales y objetivos

Algunas veces, los objetos ponen de manifiesto a los actores, o viceversa. Hay que centrar la discusión en los actores principales en primer lugar para establecer el marco para las investigaciones posteriores. Los *actores principales* tienen objetivos de usuario que se satisfacen mediante el uso de los servicios de sistema. Los *actores de apoyo* proporcionan servicios al sistema que se está diseñando.

Los actores principales y sus objetivos de usuario, en términos de los artefactos del UP, debe corresponderse con una sección del artefacto Visión y puede recogerse en una **lista actor-objetivo**.

<i>Actor</i>	<i>Objetivo</i>	<i>Actor</i>	<i>Objetivo</i>
Cajero	procesar ventas procesar alquileres gestionar las devoluciones abrir caja cerrar caja ...	Administrador del Sistema	añadir usuarios modificar usuarios eliminar usuarios gestionar seguridad gestionar las tablas del sistema ...
Director	poner en marcha suspender operación ...	Sistema de Actividad de Ventas	analizar los datos de ventas y rendimiento
...

El actor principal y los objetivos de usuario dependen del límite del sistema. También se pueden identificar actores y objetivos por medio del análisis de eventos externos. Un grupo de eventos pertenecen al mismo objetivo o caso de uso de nivel EBP.

Paso 4: definir los casos de uso

Los casos de uso de nivel EBP se definen, por lo general, por cada objetivo de usuario. Una excepción típica es agrupar objetivos separados CRUD (crear, recuperar, actualizar, eliminar) en un caso de uso CRUD (por convención, *Gestionar<X>*).

3.8. Escritura de casos de uso en un estilo esencial independiente de la interfaz de usuario

Investigar y preguntar acerca de los objetivos, en lugar de las tareas y procedimientos fomenta que se centre la atención en la esencia de los requisitos.

Escritura en estilo esencial

La narración se expresa el nivel de las *intenciones* de los usuarios y *responsabilidades* del sistema, en lugar de sus acciones concretas. Son independientes de los detalles acerca de la tecnología y los mecanismos, especialmente aquellos relacionados con la UI. Utiliza el formato en dos columnas (se puede escribir en una columna).

... Intención del Actor 1. El Administrador se identifica 3. ...	Responsabilidad del Sistema 2. Autenticar la identidad	... 1. El Administrador se identifica 2. El Sistema autentica la identidad 3. ...
--	--	--

Estilo concreto

En los **casos de uso concreto** se incluye en el texto del caso de uso de las decisiones acerca de la interfaz de usuario. Puede mostrar instantáneas de las pantallas de las ventanas, discutir la navegación de las ventanas, la manipulación de los elementos de la GUI, etc. Se han de evitar durante el trabajo de requisitos inicial.

... 1. El Administrador introduce su ID y contraseña en el cuadro de diálogo (ver Dibujo 3). 2. El Sistema autentica al Administrador. 3. El Sistema muestra la ventana de "edición de usuarios" (ver Dibujo 4). 4. ...

3.9. Actores

Un **actor** es cualquier cosa con comportamiento, incluyendo el propio sistema que se está estudiando (SuD, *System under Discussion*) cuando solicita los servicios de otros sistemas. Los actores principales y de apoyo aparecen en los pasos de acción del texto del caso de uso. No solamente son roles que juegan personas, sino también organizaciones, software y máquinas. 3 tipos de actores externos:

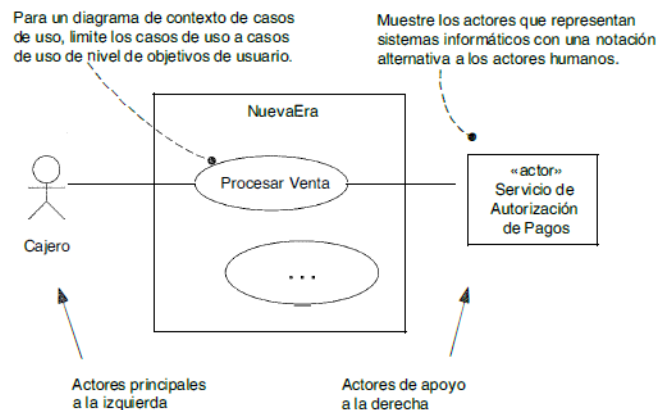
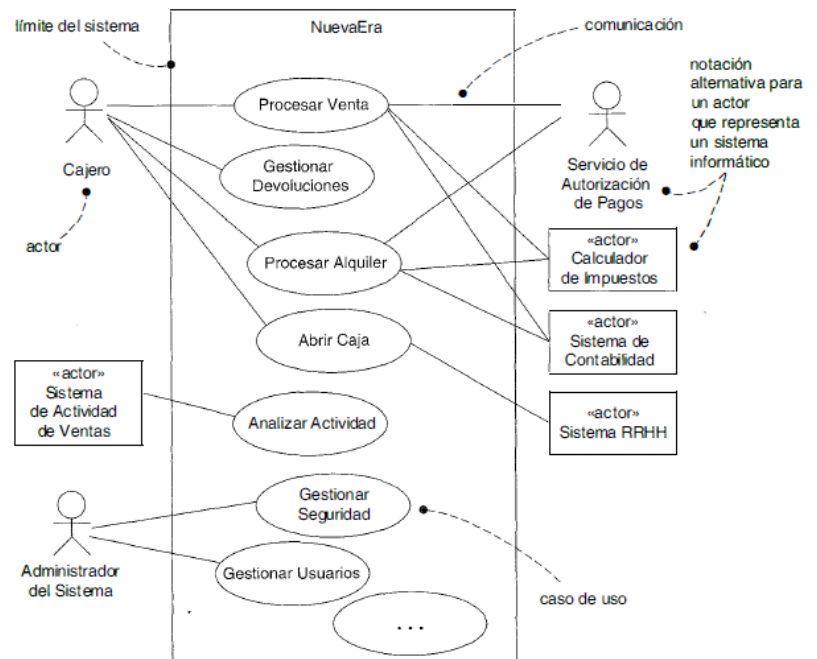
- **Actor principal:** tiene objetivos de usuario que se satisfacen mediante el uso de los servicios del SuD. Se identifica para encontrar los objetivos de usuario, los cuales dirigen los casos de uso.
- **Actor de apoyo:** proporciona un servicio al SuD. Normalmente se trata de un sistema informático, pero podría ser una organización o una persona. Se identifica para clarificar las interfaces externas y los protocolos.
- **Actor pasivo:** está interesado en el comportamiento del caso de uso, pero no es principal ni de apoyo. Se identifica para asegurar que *todos* los intereses necesarios se han identificado y satisfecho.

3.10. Diagramas de casos de uso

UML proporciona notación para los diagramas de casos de uso con el fin de ilustrar los nombres de los casos de uso y los actores y las relaciones entre ellos. Son secundarios en el trabajo con los casos de uso. Los casos de uso son documentos de texto y significan escribir texto.

Un diagrama de casos de uso proporciona un conciso diagrama de contexto visual del sistema, que muestran los actores externos y cómo utilizan el sistema. Los diagramas de casos de uso se pueden escribir junto con la lista actor-objetivo.

Un diagrama de casos de uso conforma un buen **diagrama de contexto**, muestra los límites de un sistema, lo que permanece fuera de él, y cómo se utiliza. Resume el comportamiento de un sistema y sus actores.



3.11. Requisitos en contexto y listas de características de bajo nivel

Una motivación clave de la idea de caso de uso es considerar y organizar los requisitos en el contexto de los objetivos y escenarios de uso de un sistema. Los casos de uso no son los únicos artefactos de requisitos necesarios. Algunos requisitos no funcionales se capturan mejor en la Especificación.

Una idea detrás de los casos de uso es sustituir las listas de características de bajo nivel por los casos de uso agrupadas por áreas funcionales, de las que se puede aprovechar contenido:

- Listas de funciones largas y detalladas no relacionan los requisitos en un contexto cohesivo, aparecen progresivamente como una lista inconexa de elementos. Los casos de uso sitúan los requisitos en el contexto de las historias y objetivos de uso del sistema.
- Si se utilizan casos de uso y listas de características detalladas, puede haber duplicación, lo que supone más problemas de consistencia y sincronización. Pueden sustituirse de las listas detalladas de características de bajo nivel por casos de uso.

Es típico y útil resumir la funcionalidad del sistema con una lista breve de características de alto nivel, denominadas **características del sistema** (tiende a incluir sólo unas pocas docenas de elementos). La lista proporciona un resumen muy conciso de la funcionalidad del sistema, independiente de la vista del caso de uso. Algunas aplicaciones exigen un punto de vista dirigido por las características. Los casos de uso no se ajustan de manera natural a estas aplicaciones o al modo en que necesitan evolucionar en términos de las fuerzas del mercado.

No hay nada orientado a objetos en los casos de uso; aunque incrementen su utilidad como método de requisitos. Sin embargo, los casos de uso son una entrada fundamental en las actividades clásicas de A/DOO.

3.12. Casos de uso en el UP

El **desarrollo dirigido por casos de uso** implica:

- Los requisitos se recogen principalmente en casos de uso, otras técnicas de requisitos son secundarias, si es que se utilizan.
- Los casos de uso son una parte importante de la planificación iterativa. Son una entrada clave para hacer estimaciones.
- Las **realizaciones de los casos de uso** dirigen el diseño.
- Los casos de uso influyen en la organización de los manuales de usuario.

El UP diferencia entre casos de uso del sistema y del negocio. Los **casos de uso del negocio** se escriben con menos frecuencia. Se crean en la disciplina Modelado de Negocio, como parte de un esfuerzo de reingeniería de los procesos de negocio a gran escala, o para ayudar a entender el contexto de un nuevo sistema en el negocio. Describen una secuencia de acciones de un negocio como un todo para cumplir un objetivo de **actor del negocio**.

Casos de uso y especificación de requisitos a lo largo de las iteraciones

El UP y el desarrollo iterativo miden el tiempo y el nivel de esfuerzo de la especificación de requisitos a lo largo de las iteraciones. El equipo técnico comienza construyendo los fundamentos de la producción del sistema cuando sólo se han detallado el 10% de los requisitos. La diferencia clave entre proceso iterativo y el proceso en cascada está en que el desarrollo con calidad de producción de los fundamentos del sistema empieza rápidamente, antes de conocer todos los requisitos.

Cerca del final de la primera iteración de elaboración, hay un segundo taller de requisitos, durante el que el 30% de los casos de uso se escriben. Se beneficia de la retroalimentación a partir de la construcción de un poco del núcleo del software. La retroalimentación incluye la evaluación del usuario, pruebas y “conocimiento de lo que se conoce”.

Momento de creación de los artefactos del UP

El Modelo de Casos de Uso comienza en la fase de inicio. La mayoría se escriben increíblemente a lo largo de las iteraciones de la fase de elaboración, proporcionando una base realista

Tabla 6.1. Muestra del esfuerzo de los requisitos a lo largo de las primeras iteraciones; no es una receta.

Disciplina	Artefacto	Comentarios y nivel de esfuerzo de los requisitos				
		Inicio 1 semana	Elab 1 4 semanas	Elab 2 4 semanas	Elab 3 3 semanas	Elab 4 3 semanas
Requisitos.	Modelo de Casos de Uso.	2 días de taller de requisitos. Se identifican por el nombre la mayoría de los casos de uso y se resumen en un párrafo breve. Sólo el 10% se escribe en detalle.	Cerca del final de esta iteración, tiene lugar un taller de requisitos de 2 días. Se obtiene un mejor entendimiento y retroalimentación a partir del trabajo de implementación, entonces se completa el 30% de los casos de uso en detalle.	Cerca del final de esta iteración, tiene lugar un taller de requisitos de dos días. Se obtiene una mejor comprensión y retroalimentación a partir del trabajo de implementación, entonces se completa el 50% de los casos de uso en detalle.	Repetir, se completa el 70% de todos los casos de uso en detalle.	Repetir con la intención de clarificar y escribir en detalle del 80-90% de los casos de uso. Sólo una pequeña parte de éstos se construyen durante la elaboración; el resto se aborda durante la construcción.
Diseño.	Modelo de Diseño.	Nada.	Diseño de un pequeño conjunto de requisitos de alto riesgo significativos desde el punto de vista de la arquitectura.	Repetir.	Repetir.	Repetir. Deberían ahora estabilizarse los aspectos de alto riesgo significativos para la arquitectura.
Implementación.	Modelo de Implementación (código, etc.)	Nada.	Implementar esto.	Repetir. Se construye el 5% del sistema final.	Repetir. Se construye el 10% del sistema final.	Repetir. Se construye el 15% del sistema final.
Gestión del Proyecto.	Plan de Desarrollo de SW.	Estimación muy imprecisa del esfuerzo total.	La estimación comienza a tomar forma.	Un poco mejor...	Un poco mejor...	Ahora se pueden establecer racionalmente la duración global del proyecto, los hitos más importantes, estimación del coste y esfuerzo.

para hacer una estimación precisa hasta el final del proyecto.

Tabla 6.2. Muestra de los artefactos UP y evolución temporal. c - comenzar; r – refinar.

Disciplina	Artefacto Iteración →	Inicio I1	Elab. E1...En	Const. C1...Cn	Trans. T1...T2
Modelado del Negocio	Modelo del Dominio		c		
Requisitos	Modelo de Casos de Uso	c	r		
	Visión	c	r		
	Especificación Complementaria	c	r		
	Glosario	c	r		
Diseño	Modelo de Diseño		c	r	
	Documento de Arquitectura SW		c		
	Modelo de Datos		c	r	
Implementación	Modelo de Implementación		c	r	r
Gestión del Proyecto	Plan de Desarrollo SW	c	r	r	r
Pruebas	Modelo de Pruebas		c	r	
Entorno	Marco de Desarrollo	c	r		

Casos de uso en la fase de inicio

No todos los casos de uso se escriben en formato completo durante la fase de inicio. Después, entre el 10-20% de los casos de uso que representan las funciones complejas principales se escriben en formato complejo. La intención del trabajo de inicio es adquirir una idea de poca fidelidad (propensa a errores) acerca del alcance, riesgo, refuerzo, viabilidad técnica, y análisis del negocio, para decir avanzar, dónde comenzar si se hace, o si parar.

Casos de uso en la elaboración

Es una fase de múltiples iteraciones de duración fija en las que se construyen incrementalmente partes del sistema arriesgadas, de alto valor o significativas desde el punto de vista de la arquitectura, y se identifican y clarifican la “mayoría” de los requisitos. La retroalimentación influye e informa del conocimiento de los requisitos que se refina de manera iterativa y adaptable.

En cada siguiente taller de requisitos breve, es el momento de adaptar y refinar la visión de los requisitos principales, que serán inestables en las primeras iteraciones y se irán estabilizando en las últimas. Durante cada taller de requisitos se refinan los objetivos de usuario y la lista de casos de uso. La elaboración conlleva programar partes del sistema. Al final de esta etapa, se debería tener una mejor definición de los casos de uso y algo de software ejecutable de calidad.

Casos de uso en la construcción

La etapa de construcción está compuesta de iteraciones de duración fija que se centra en completar el sistema, una vez que las principales cuestiones arriesgadas e inestables se han establecido en la elaboración. Tiene lugar todavía la escritura de casos de uso menores y quizás talleres de requisitos. La mayoría de los requisitos funcionales y no funcionales principales deberían haberse estabilizado de manera iterativa y adaptable.

3.13. Artefactos UP y contexto del proceso

Los casos de uso influyen en muchos artefactos UP.

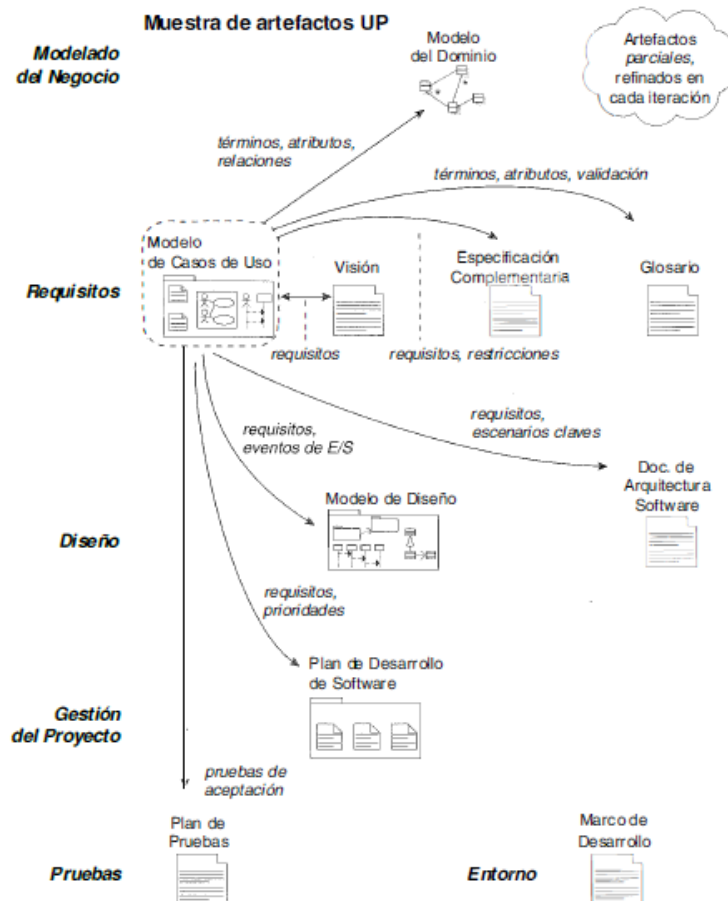


Figura 6.5. Muestra de la influencia entre los artefactos UP.

4. IDENTIFICACIÓN DE OTROS REQUISITOS

No es suficiente escribir casos de uso. Existen otros tipos de requisitos que son necesarios identificar, como los relacionados con documentación, empaquetado, soporte, licencia, etc. Se recogen en la **Especificación Complementaria**. El **Glosario** almacena los términos y definiciones y puede también jugar el rol de diccionario de datos.

La **Visión** resume la “visión” del proyecto. Sirve para comunicar de manera concisa las grandes ideas acerca de por qué se propuso el proyecto, cuáles son los problemas, las personas involucrada, qué necesitan, y cuál podría ser la apariencia de la solución propuesta. Define la vista que tienen las personas involucradas del producto que se va a desarrollar, especificada en términos de las necesidades y características claves de dichas personas y proporciona la base contractual para los requisitos técnicos más detallados.

4.1. Especificación Complementaria

La **Especificación Complementaria** captura otros requisitos, información y restricciones que no se recogen fácilmente en los casos de uso o el Glosario, que comprende los atributos o requisitos de calidad “URPS+” de todo el sistema. Los requisitos específicos de un caso de uso pueden escribirse en primer lugar con el caso de uso, en una sección *Requisitos Especiales*, pero también reunirlos en la Especificación Complementaria. Los elementos de la Especificación Complementaria comprenden:

- Requisitos FURPS+ (funcionalidad, facilidad de uso, fiabilidad, rendimiento y soporte).
- Informes.
- Restricciones de SW y HW.
- Restricciones de desarrollo (herramientas de proceso y desarrollo).
- Otras restricciones de diseño e implementación.
- Cuestiones de internacionalización (unidades, idiomas...).
- Documentación (usuario, instalación, administración) y ayuda.
- Licencia y otras cuestiones legales.
- Empaquetado.
- Estándares (técnico, seguridad, calidad).

- Cuestiones del entorno físico (por ejemplo, calor o vibración).
- Cuestiones operacionales.
- Reglas del dominio o negocio
- Información en los dominios de interés.

Las **restricciones** no son comportamientos, sino otro tipo de restricciones del diseño o el proyecto. También son requisitos.

Atributos de calidad

Algunos requisitos se denominan **atributos de calidad** que incluyen facilidad de uso (usability), fiabilidad (reliability), etc., y se refieren a cualidades del sistema. Hay dos tipos:

1. Observables en la ejecución (funcionalidad, facilidad de uso, fiabilidad, rendimiento, etc.).
2. No observables en ejecución (soporte, pruebas, etc.).

La funcionalidad se especifica en los casos de uso, como otros atributos de calidad relacionados con casos de uso específicos. Otros atributos de calidad FURPS+ del sistema se describen en la Especificación Complementaria. Los atributos de calidad de todo el sistema son especialmente interesantes porque el análisis y el diseño de la arquitectura tienen que ver en gran medida con la identificación y resolución de los atributos de calidad en el contexto de los requisitos funcionales. Entre los atributos de calidad existen interdependencias, lo que implica compromisos.

Reglas del dominio

Las reglas del dominio dictan el modo en el que podría operar un dominio o negocio. No son requisitos de ninguna aplicación, aunque los requisitos de una aplicación se ven afectados por las reglas de dominio. Las políticas de la compañía, leyes físicas y leyes gubernamentales son reglas de dominio típicas. Las **reglas del negocio** son el tipo más común. Resulta útil identificar y registrar aquellas reglas de dominio que afectan a los requisitos, materializados en los casos de uso, porque pueden clarificar el contenido de un caso de uso ambiguo o incompleto.

Información en los dominios de interés

Es útil escribir explicaciones de dominio relacionados con el sistema para presentar el contexto y ayuda a la comprensión por parte del equipo de desarrollo y puede contener referencias importantes a expertos o a la literatura, fórmulas, leyes u otras referencias.

4.2. Visión

Durante el trabajo de requisitos en la fase de inicio, hay que definir un enunciado del problema conciso, reduciendo la posibilidad de que el personal involucrado esté intentando solucionar problemas ligeramente diferentes.

El problema de	...
afecta	...
el impacto de lo cual es	...
una solución de éxito sería	...

Los objetivos de alto nivel y problemas claves del personal involucrado pueden resumirse en una tabla a un nivel más alto que los casos de uso de nivel de tarea, y muestra importantes objetivos no funcionales y de calidad que podrían pertenecer a un caso de uso o abarcar a muchos.

Algunas veces, el personal involucrado llega a soluciones que no son las más apropiadas o que no tratan el problema subyacente más importante y esencial. El analista del sistema necesita estudiar la cadena de problemas y objetivos para aprender los problemas subyacentes, y su importancia e impacto relativos, y de este modo priorizar y solucionar las cuestiones más relevantes con una solución ingeniosa.

Existen técnicas útiles de ayuda al grupo para el descubrimiento de los problemas y objetivos esenciales y para ayudar a la generación de ideas y su priorización: mapas mentales, diagramas causa-efecto, diagramas Pareto, tormenta de ideas, multivotación, votación por puntos, técnica de grupo normal, escribir ideas en silencio, diagrama de afinidad...

Características del sistema - requisitos funcionales

Los casos de uso no son necesariamente el único modo para expresar los requisitos funcionales porque:

- Son detallados
- Listan los nombres de los casos de uso y todavía podría ser demasiado extensa. Los nombres pueden ocultar funcionalidad interesante.

- Algunas funcionalidades importantes se expresan como sentencias cortas que no se corresponden con los nombres de los casos de uso u objetivos de nivel de Procesos del Negocio Elementales (EBP).

Un modo alternativo complementario para expresar las funciones del sistema es mediante las **características del sistema**, sentencias concisas, de alto nivel, que resumen las funcionalidades del sistema. En el UP, una característica del sistema es “un servicio observable externamente proporcionado por el sistema que cumple directamente una necesidad del personal involucrado” → *El sistema deberá hacer <característicaX>*. Es complementario a los casos de uso, ya que son concisas. La mayoría de las características del sistema se expresarán de manera detallada en el texto de los casos de uso.

Es habitual organizar una jerarquía de dos niveles de características del sistema. Pero en el documento de Visión, más de dos niveles llevaría a un detalle excesivo. Algunas veces, estas características de segundo nivel son básicamente equivalentes a los nombres de los casos de uso. Es deseable un documento de Visión con menos de 50 características.

Otros requisitos en la Visión

En la Visión, las características del sistema resumen brevemente los requisitos funcionales, que se expresan en detalle en los casos de uso. La Visión *puede* resumir otros requisitos que se detallan en la sección de *Requisitos Especiales* de los casos de uso, y en la Especificación Complementaria (SS, Supplementary Specification). Sin embargo, hay peligro de duplicaciones inútiles.

La secuencia recomendada es:

1. Escribir un primer borrador breve de la Visión.
2. Identificar los objetivos de usuario y los casos de uso de apoyo.
3. Escribir algunos casos de uso y comenzar la Especificación Complementaria.
4. Refinar la Visión, resumiendo la información a partir de éstos.

4.2. Glosario (Diccionario de Datos)

El **Glosario** es una lista de los términos relevantes y sus definiciones. Se utiliza para reducir los problemas de comunicación y los requisitos ambiguos. Hay que comenzarlo cuanto antes. El objetivo no es recoger todos los posibles términos, sino aquellos que no están claros, son ambiguos o que requieren algún tipo de elaboración relevante.

El Glosario como diccionario de datos

En el UP, el Glosario también juega el rol de **diccionario de datos**, un documento que recoge los datos sobre los datos (**metadatos**). Durante la fase de inicio, el glosario debe ser un documento sencillo de términos y descripciones. Durante la elaboración, podría ampliarse a un diccionario de datos. Los atributos de los términos podrían contener:

- Alias.
- Descripción.
- Formato (tipo, longitud, unidad).
- Relaciones con otros elementos.
- Rango de valores.
- Reglas de validación.

Se deben tener en cuenta las unidades (moneda, medidas, ...), especialmente en esta era de aplicaciones software internacionalizadas. El Glosario no está destinado sólo a términos atómicos y puede y debe incluir términos compuestos y alias utilizados para describir una colección de datos que se transmiten entre los actores en los casos de uso.

4.3. Otras consideraciones

Los requisitos escritos pueden dar la impresión de que se han entendido los requisitos reales y están bien definidos, y pueden utilizarse (pronto) para estimar y planificar el proyecto de manera fiable. Lo que realmente importa es construir SW que pase las pruebas de aceptación definidas por los usuarios y el resto de personas implicadas en el proyecto, y que satisfaga sus verdaderos objetivos. Escribir la Visión y Especificación Complementaria pone en claro una primera aproximación de lo que se quiere; pero no son una especificación fiable.

Puede haber artefactos digitales recogidos sólo on-line a través del sitio web del proyecto, en lugar de ser documentos estáticos planos.

El objetivo de la fase de inicio es recopilar sólo la información suficiente para establecer una visión común, decidir si es viable avanzar y si merece la pena una investigación seria del proyecto en la fase de elaboración. No son necesarios más

diagramas que los simples diagramas de casos de uso UML. La mayoría de los diagramas UML se realizarán en la siguiente fase (elaboración).

4.4. Otros artefactos de requisitos en el UP

Todos los artefactos de requisitos se comienzan en la fase de inicio, se trabaja sobre ellos sobre todo durante la elaboración.

Inicio

No debería ser que estos artefactos de requisitos se terminen en la fase de inicio, apenas se habrán empezado. Durante la fase de inicio, la Visión resume la idea del proyecto de manera que ayude a determinar si merece la pena continuar, y dónde comenzar.

La mayoría del trabajo de requisitos ocurre durante la elaboración, la Especificación Complementaria debería desarrollarse ligeramente durante la fase de inicio. La información de entrada a estos artefactos podría generarse durante un taller de requisitos de la fase de inicio. No se escribirá un borrador legible de los artefactos, sino que lo elaborará después el analista del sistema.

Tabla 7.1. Muestra de los artefactos UP y evolución temporal. c – comenzar; r – refinar.

<i>Disciplina</i>	<i>Artefacto Iteración →</i>	<i>Inicio I1</i>	<i>Elab. E1...En</i>	<i>Const. C1...Cn</i>	<i>Trans. T1...T2</i>
Modelado del Negocio	Modelo del Dominio		c		
Requisitos	Modelo de Casos de Uso <i>Visión</i> <i>Especificación Complementaria</i> <i>Glosario</i>	c	r		
		c	r		
		c	r		
		c	r		
Diseño	Modelo de Diseño Documento de Arquitectura SW Modelo de Datos		c	r	
			c		
			c	r	
Implementación	Modelo de Implementación		c	r	r
Gestión del Proyecto	Plan de Desarrollo SW	c	r	r	r
Pruebas	Modelo de Pruebas		c	r	
Entorno	Marco de Desarrollo	c	r		

Elaboración

A través de las iteraciones de la elaboración, se refina la “visión” y la Visión, en base a la retroalimentación a partir de la construcción incremental de partes del sistema. Mediante la investigación de requisitos continuada y el desarrollo iterativo, los otros requisitos llegarán a verse más claros y se recogerán en la SS.

La mayoría de los términos se descubren y elaboran en el Glosario durante esta fase. Llegando al final de la elaboración, es viable contar con casos de uso, una Especificación Complementara y un documento de Visión que reflejan las características estabilizadas más importantes y otros requisitos que se deben completar. La Especificación Complementaria y la Visión no se congelan o “se dan por concluidas”.

- En el desarrollo iterativo y el UP, se entiende que no importa cuanta atención merecida se presta a la especificación de requisitos, algún cambio es inevitable. Podría ser una mejora oportunista que incorpora una novedad importante o un cambio debido a una visión mejorada.
- En el desarrollo iterativo, es un valor central contar con un compromiso continuo por parte del personal involucrado para evaluar, proporcionar retroalimentación, y dirigir el proyecto como realmente quieren.

Construcción

En la construcción, los requisitos importantes, tanto funcionales como de otro tipo, deberían estar estabilizados, no terminados, aunque dispuestos a cambios menores.

4.5. Artefactos UP y contexto del proceso

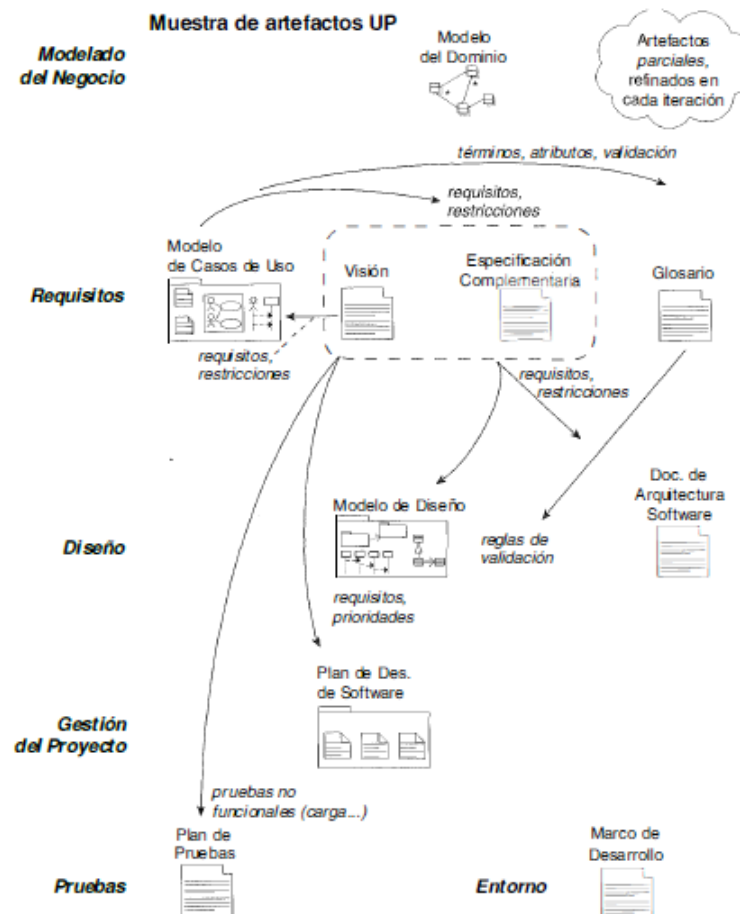


Figura 7.1. Muestra de la influencia entre los artefactos UP.

5. DEL INICIO A LA ELABORACIÓN

5.1. Resumen de la etapa de inicio

En la etapa de inicio, los artefactos creados deberían ser breves e incompletos, la etapa rápida, y la investigación ligera. No es la fase de requisitos del proyecto, sino una etapa breve para determinar la viabilidad, riesgo y alcance básicos y decidir si merece la pena más investigación seria en el proyecto, que tendrá lugar en la elaboración. Algunas de las actividades y artefactos posibles en la fase de inicio comprenden:

- Taller de requisitos.
- La mayoría de los actores, objetivos y casos de uso, con los nombres.
- La mayoría de los casos de uso escritos en formato breve
- Identificación de la mayoría de los requisitos de calidad influyentes y de riesgo.
- Escritura de la primera versión de la Visión y Especificación Complementaria.
- Lista de riesgos.
- Prototipos de pruebas de conceptos técnicos.
- Prototipos orientados a la interfaz de usuario.
- Recomendaciones sobre qué componentes comprar/construir/reutilizar.
- Arquitectura de alto nivel *candidata* y componentes propuestos.
- Plan para la primera iteración.
- Lista de herramientas candidatas.

5.2. Perspectivas de cara a la elaboración

La elaboración es la serie inicial de iteraciones durante las que el equipo lleva a cabo un estudio serio, implementa el núcleo central de la arquitectura, aclara la mayoría de los requisitos, y aborda las cuestiones de alto riesgo. Consta de entre

2 y 4 iteraciones (cada iteración dura entre 2 y 6 semanas). Si es probable que el equipo no cumpla con la fecha, algunos requisitos se colocan en la lista de tareas futuras, de manera que la iteración pueda concluir a tiempo.

No se corresponde con una fase de diseño o una fase en la que se desarrollan completamente los modelos preparándolos para que se implementen en la etapa de construcción. Durante esta fase, uno no está creando prototipos desechables; sino que el código y el diseño son porciones del sistema final con calidad de producción → **prototipo de la arquitectura**. En el UP, significa un subconjunto de producción del sistema final → **arquitectura ejecutables o base de la arquitectura**.

Las ideas claves y buenas prácticas en esta fase son:

- Llevar a cabo iteraciones breves, de duración fija, dirigidas por el riesgo.
- Comenzar a programar pronto.
- Diseñar, implementar y probar, de manera adaptable, las partes básicas y arriesgadas de la arquitectura.
- Probar desde el principio, a menudo y de manera realista.
- Adaptar en base a la retroalimentación.
- Escribir la mayoría de los casos de uso y otros requisitos en detalle.

Las iteraciones iniciales construyen y prueban el núcleo central de la arquitectura:

- Empleo de diseño e implementación “ancho y superficial”: identificación de los diferentes procesos, capas, paquetes, y subsistemas, y sus responsabilidades e interfaces de alto nivel.
- Refinamiento de las interfaces locales y remotas inter-módulos.
- Integración de los componentes existentes.
- Implementación de escenarios simplificados de caminos de éxito y fracaso que impulsan el diseño, la implementación y las pruebas, a través de muchos componentes importantes.

Las pruebas en la fase de elaboración son importantes, para obtener retroalimentación, adaptar y probar que el núcleo es robusto.

5.3. Planificación de la siguiente iteración

Se organizan los requisitos y las iteraciones según el riesgo, grado de cobertura y naturaleza crítica:

- **Riesgo:** comprende tanto la complejidad técnica como otros factores, como incertidumbre del esfuerzo o facilidad de uso.
- **Cobertura:** implica que todas las partes importantes del sistema se tratan, al menos, en las primeras iteraciones.
- **Naturaleza crítica:** se refiere a las funciones de alto valor para el negocio.

Estos criterios se utilizan para priorizar el trabajo a través de las iteraciones. La clasificación se realiza antes de la Iteración 1, pero después otra vez antes de la Iteración 2, etc., cuando nuevos requisitos y nuevas interpretaciones influyan en el orden.

En términos de los artefactos del UP:

- Los requisitos elegidos para la siguiente iteración se listan brevemente en un **Plan de Iteración**. No se trata de un plan de todas las iteraciones, sólo un plan de la siguiente.
- Si la breve descripción del Plan de Iteración es insuficiente, se podría escribir con más detalle una tarea o requisitos en una **Solicitud de Cambio**, y dársela al equipo responsable.
- La clasificación de requisitos se recoge en el **Plan de Desarrollo de Software**.

5.4. Requisitos y énfasis de la Iteración 1: habilidades de A/DOO fundamentales

La Iteración 1 de la fase de elaboración hace hincapié en una gama de habilidades de A/DOO fundamentales y comunes, utilizadas en la construcción de sistemas de objetos, como la asignación de las responsabilidades a los objetos.

Es normal continuar con el trabajo de varios escenarios o características del mismo caso de uso en varias iteraciones y gradualmente ampliar el sistema para, al final, tratar toda la funcionalidad requerida. Los casos de uso simples y cortos podrían completarse en una iteración.

Los artefactos del UP creados en la fase de elaboración son:

<i>Artefacto</i>	<i>Comentario</i>
Modelo del Dominio	Es una visualización de los conceptos del dominio; es similar al modelo de información estático de las entidades del dominio.
Modelo de Diseño	Es el conjunto de diagramas que describen el diseño lógico. Comprende los diagramas de clases software, diagramas de interacción, diagramas de paquetes, etcétera.
Documento de la Arquitectura Software	Una ayuda de aprendizaje que resume las cuestiones claves de la arquitectura y cómo se resuelven en el diseño. Es un resumen de las ideas destacadas del diseño y su motivación en el sistema.
Modelo de Datos	Incluye los esquemas de bases de datos, y las estrategias de transformación entre representaciones de objetos y no objetuales.
Modelo de Pruebas	Una descripción de lo que se probará y cómo.
Modelo de Implementación	Se corresponde con la implementación real —el código fuente, ejecutables, base de datos, etcétera—.
Guiones de Caso de Uso, Prototipos UI	Descripción de la interfaz de usuario, caminos de navegación, modelos de facilidad de uso, etcétera.

5.5. No es fase de elaboración si...

- La duración es superior a “unos pocos” meses.
- Sólo comprende una iteración.
- La mayoría de los requisitos se definieron antes de la elaboración.
- Los elementos arriesgados y el núcleo de la arquitectura no se abordan.
- El resultado no es una arquitectura *ejecutable*.
- Se considera fundamentalmente una fase de requisitos.
- Se intenta llevar a cabo un diseño completo.
- No se llevan a cabo pruebas realistas en las primeras etapas.
- La arquitectura se termina de forma especulativa antes de la programación.
- Se considera una etapa para realizar la programación de pruebas de conceptos.
- No se realizan varios talleres de requisitos breves que adaptan y refinan los requisitos en base a la retroalimentación.