

# CASOS DE USO

## 2.0

La guía para ser exitoso con los casos de uso

**Ivar Jacobson**  
**Ian Spence**  
**Kurt Bittner**

**Traducción de:**  
**Luis Antonio Salazar Caraballo**  
**Carlos Mario Zapata Jaramillo**

Acerca de esta Guía	3
Cómo leer esta Guía	3
¿Qué son los Casos de Uso 2.0?	4
Primeros Principios	5
Principio 1: Mantenerlos simples al narrar historias	5
Principio 2: Entender el panorama general	5
Principio 3: Enfocarse en el Valor	7
Principio 4: Construir el sistema por partes	8
Principio 5: Entregar el sistema en incrementos	10
Principio 6: Adaptarse para cubrir las necesidades del equipo	11
Contenido de los Casos de Uso 2.0	12
Cosas con las cuales trabajar	12
Productos de Trabajo	17
Cosas Por Hacer	21
Emplear los Casos de Uso 2.0	28
Casos de Uso 2.0: Aplicables para todos los tipos de sistema	28
Casos de Uso 2.0: Manejando todo tipo de requisitos	29
Casos de Uso 2.0: Aplicable para todos los enfoques de desarrollo	29
Casos de Uso 2.0: Escalando para cubrir todas tus necesidades: escalando hacia adentro, escalando horizontal y verticalmente	37
Conclusión	38
Apéndice 1: Productos de Trabajo	39
Información de Soporte	40
Casos de Prueba	42
Modelo de Casos de Uso	44
Narrativa del Caso de Uso	45
Realización del Caso de Uso	47
Glosario de Términos	49
Agradecimientos	51
General	51
Personas	51
Bibliografía	52
Acerca de los Autores	53
Acerca de los Traductores	54

---

# Acerca de esta Guía

Esta guía describe cómo aplicar casos de uso de una manera ágil y escalable. Se construye con el actual nivel de desarrollo para presentar una evolución de la técnica de casos de uso, que denominamos Casos de Uso 2.0. El objetivo es entregarle una base para ayudarlo a obtener lo mejor de sus casos de uso, algo que sea aplicable no solo para desarrollos de multi-sistemas complejos con pequeños equipos ágiles locales sino también con equipos grandes distribuidos.

Esta guía presenta los elementos necesarios del desarrollo dirigido por casos de uso como una práctica accesible y reusable. También proporciona una introducción a la idea de casos de uso y su aplicación. La guía se mantuvo liviana de manera deliberada. No es una guía detallada sobre todos los aspectos de los casos de uso ni un tutorial sobre modelado de casos de uso. Quizás no sea suficiente para que usted adopte la práctica. Por ejemplo, no intenta enseñarle como modelar; para esto lo remitimos a libros publicados en la materia.

## Cómo leer esta Guía

La guía se estructura en cuatro capítulos principales:

- ¿Qué son los Casos de Uso 2.0?: Una introducción de una página a la práctica.
- Primeros Principios: Una introducción a los casos de uso basada en seis principios que actúan como base para la práctica.
- Contenido de los Casos de Uso 2.0: La práctica misma presentada como un conjunto de conceptos clave, actividades, productos de trabajo y las reglas que las unen en conjunto.
- Empleando los Casos de Uso 2.0: Un resumen de cuándo y cómo aplicar la práctica.

Esta breve introducción antecede estos capítulos que finalizan con una corta conclusión.

Si usted recién comienza a trabajar con casos de uso, entonces debería leer los capítulos ¿Qué son los Casos de Uso 2.0?, “Primeros Principios” y “Usando Casos de Uso 2.0” para entender los conceptos básicos. Luego, puede profundizar en el “Contenido de los Casos de Uso 2.0” cuando comience a aplicar la práctica.

Si le son familiares los elementos básicos de los casos de uso, entonces usted podría preferir ir directamente a los capítulos “Contenido de los Casos de Uso 2.0” y “Empleando los Casos de Uso 2.0” una vez que lea el capítulo ¿Qué es Casos de Uso 2.0? Esto lo ayudará a comparar Casos de Uso 2.0 con su propia experiencia y a entender lo que cambió.

Alternativamente, siempre puede leer todos los capítulos en el orden de presentación.

---

# ¿Qué son los Casos de Uso 2.0?

**Caso de Uso:** Un caso de uso expresa todas las formas de usar un sistema para alcanzar una meta particular para un usuario. En conjunto, los casos de uso le proporcionan todos los caminos útiles de usar el sistema e ilustran el valor que este provee.

**Casos de Uso 2.0:** Es una práctica escalable y ágil que emplea los casos de uso para capturar un conjunto de requisitos y conducir el desarrollo incremental de un sistema que los realice.

Los Casos de Uso 2.0 conducen el desarrollo de un sistema ayudándole, en principio, a entender cómo se empleará el sistema y luego ayudándole a evolucionar un sistema apropiado que apoye a los usuarios. Esta técnica se puede usar junto con prácticas administrativas y técnicas seleccionadas para apoyar el desarrollo exitoso de software y otras formas de sistemas. Cómo se verá más adelante, los Casos de Uso 2.0 son:

- Livianos
- Escalables
- Versátiles
- Fáciles de usar

Los casos de uso dejan claro lo que hará un sistema y, por omisión intencional, lo que no hará. Estos posibilitan una visión efectiva, el manejo del alcance y el desarrollo incremental de sistemas de cualquier tipo y cualquier tamaño. Los casos de uso se vienen empleando para conducir el desarrollo de sistemas de software desde su presentación inicial en la OOPSLA de 1987. Con los años, se convirtieron en la base de muchos métodos diferentes y en parte integral del Lenguaje de Modelado Unificado (UML). Se emplean en muchos contextos y entornos diferentes y para muchos tipos distintos de equipos. Por ejemplo, los casos de uso pueden beneficiar tanto a equipos pequeños de desarrollo ágil que producen aplicaciones de uso intensivo como a grandes proyectos que producen sistemas complejos de sistemas interconectados, tales como sistemas corporativos, líneas de productos y sistemas en la nube.

La estrategia de los casos de uso tiene un alcance mucho más amplio que solo la captura de requisitos. Como se mencionó antes, se pueden y se deberían emplear para conducir el desarrollo, lo que significa que los Casos de Uso 2.0 apoyan el análisis, diseño, planeación, estimación, seguimiento y prueba de sistemas. No establecen como se debería planear o manejar el trabajo de desarrollo o como se debería diseñar, desarrollar o probar el sistema. Sin embargo, proporcionan una estructura para la adopción exitosa de sus prácticas seleccionadas de administración y desarrollo.

Los Casos de Uso 2.0 existen como una práctica probada y bien definida. Aunque el término Casos de Uso 2.0 sugiere una nueva versión de los casos de uso, no se refiere a una actualización del Lenguaje de Modelado Unificado, sino más bien a cambios acumulativos en la forma como los desarrolladores de software y analistas del negocio aplican los casos de uso. Un caso de uso es, aún, un caso de uso, pero las formas en que los presentamos, dirigimos y manejamos evolucionaron para ser más efectivas. Los cambios no son teóricos sino cambios pragmáticos basados en 20 años de experiencia en todo el mundo y en todas las áreas del desarrollo de software.

---

# Primeros Principios

Hay seis principios básicos en el corazón de cualquier aplicación exitosa de los casos de uso:

1. Mantenerlos simples al narrar historias
2. Entender el panorama general
3. Enfocarse en el valor
4. Construir el sistema por partes
5. Entregar el sistema en incrementos
6. Adaptarse para cubrir las necesidades del equipo

En este capítulo miraremos estos principios en más detalle y los usaremos para presentar los conceptos de modelado de los casos de uso y el desarrollo dirigido por casos de uso.

## Principio 1: Mantenerlos simples al narrar historias

La narración de historias permite a las culturas sobrevivir y progresar; es el camino más simple y más efectivo para pasar el conocimiento de una persona a otra. Es la mejor manera de comunicar lo que un sistema debe hacer y hacer que todo el mundo trabaje en el sistema sobre los mismos objetivos.

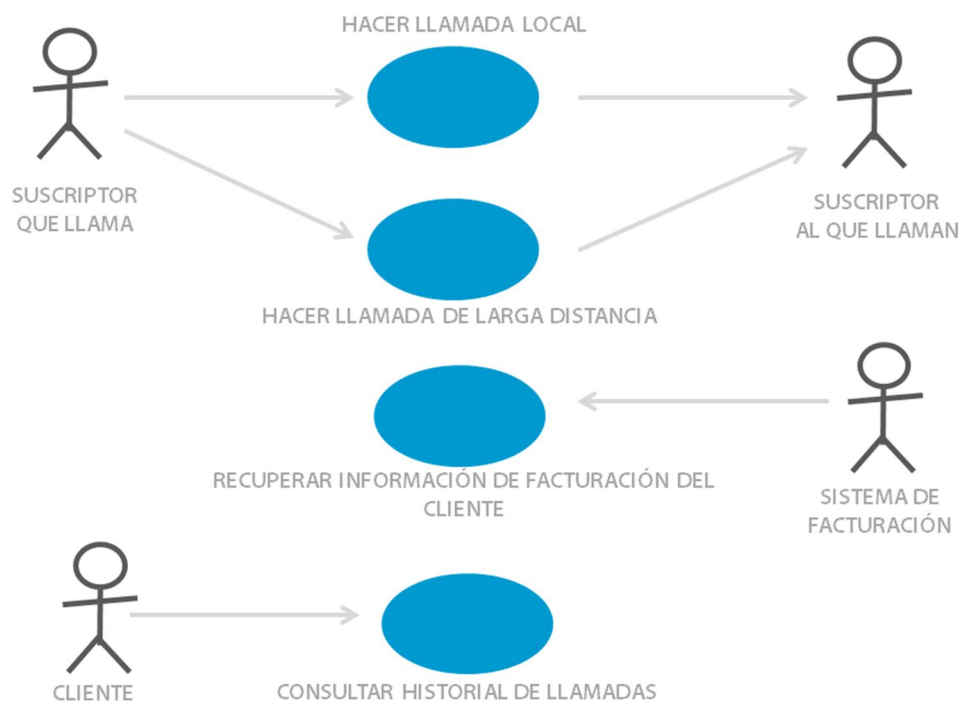
Los casos de uso capturan las metas del sistema. Para entender un caso de uso narramos historias. Las historias cubren la manera de alcanzar una meta exitosamente y la manera de manejar cualquier problema que ocurra en el camino. Los casos de uso proporcionan una forma de identificar y capturar todas las historias diferentes, pero relacionadas de una manera simple, aunque detallada. Esto posibilita que los requisitos del sistema se capturen, compartan y entiendan fácilmente.

Como un caso de uso se enfoca en el logro de una meta particular, este proporciona un medio para narrar historias. En vez de intentar describir el sistema de una sola vez, podemos hacerlo un caso de uso a la vez. Los resultados de la narración de historias se capturan y presentan como parte de la narrativa del caso de uso, que acompaña a cada caso de uso.

Cuando usamos la técnica de la narración de historias para comunicar los requisitos, es esencial asegurarnos que las historias se capturen de una forma que las haga accionables y verificables. Un conjunto de casos de prueba acompañan cada narrativa de casos de uso para completar la descripción del caso de uso. Los casos de prueba son la parte más importante de la descripción de un caso de uso, más importante aún que la narrativa del caso de uso. Esto es así porque los casos de prueba hacen realidad las historias y su uso puede demostrar sin ambigüedad que el sistema hace lo que se supone que haga. Son los casos de prueba los que definen lo que significa implementar exitosamente el caso de uso.

## Principio 2: Entender el panorama general

Sin importar si el sistema que está desarrollando es grande o pequeño, o si es un sistema de software, uno de hardware o un nuevo negocio, es esencial que usted entienda el panorama general. Sin entender el sistema como un todo, será imposible tomar las decisiones correctas acerca de lo que incluye el sistema, lo que estará por fuera, lo que costará y el beneficio que proporcionará. Esto no significa capturar todos los requisitos de una sola vez. Tan solo es necesario crear algo que resuma el sistema deseado y permita entender el alcance y el progreso a nivel del sistema.



**Figura 1:** El diagrama de casos de uso para un sistema telefónico simple

Un diagrama de casos de uso es una forma simple de presentar una visión general de los requisitos de un sistema. La **Figura 1** muestra el diagrama de casos de uso para un sistema telefónico simple. De esta imagen se pueden ver todas las formas en que el sistema se puede usar, quién inicia la interacción y todas las partes involucradas. Por ejemplo, un Suscriptor que llama puede hacer una llamada local o una llamada de larga distancia a cualquiera de los Suscriptores a los que se llaman en el sistema. También es posible ver que los usuarios no tienen que ser personas sino que también pueden ser otros sistemas y, en algunos casos, ambos (por ejemplo, una máquina contestadora, y no una persona, podría desempeñar el rol de Suscriptor que llama).

Un diagrama de casos de uso es una vista del modelo de casos de uso. Estos modelos reconocen el hecho de que los sistemas apoyan muchas metas distintas de muchos interesados diferentes. En un modelo de casos de uso, los interesados que usan el sistema y contribuyen al logro de las metas se modelan como actores, mientras que las formas en las que se usará el sistema para alcanzar estas metas se modelan como casos de uso. De esta manera, el modelo de casos de uso proporciona el contexto para que los requisitos del sistema se descubran, compartan y entiendan. También, suministra un panorama general fácilmente accesible de todas las cosas que hará el sistema. En un diagrama de casos de uso como el de la **Figura 1**, los actores se muestran como monigotes y los casos de uso como elipses. Las flechas punteadas indican el iniciador de la interacción (un Actor o el Sistema), permitiéndonos ver claramente quien inicia el caso de uso.

Un modelo de casos de uso es un modelo de todas las formas útiles para emplear un sistema. Permite definir rápidamente el alcance del sistema (lo que está incluido y lo que no) y da al equipo un panorama de lo que hará el sistema. Nos permite hacer esto sin sofocarnos con los detalles de los requisitos o los aspectos internos del sistema. Con poca experiencia, es muy fácil producir modelos de casos de uso para los más complejos sistemas, creando un cuadro general fácilmente accesible que haga visible a todos los interesados el alcance y las metas del sistema.

## Principio 3: Enfocarse en el Valor

Cuando intentamos entender cómo se empleará un sistema, siempre es importante enfocarse en el valor que este prestará a sus usuarios y a otros interesados. Solo se genera valor si el sistema se usa realmente; así, es mejor enfocarse en cómo se empleará el sistema que en las largas listas de funciones o características que ofrecerá.

FLUJO BÁSICO	FLUJOS ALTERNATIVOS
1. Insertar Tarjeta	A1 Tarjeta Inválida
2. Validar Tarjeta	A2 Cantidad No-Estándar
3. Seleccionar Retiro de Efectivo	A3 Recibo Requerido
4. Seleccionar Cuenta	A4 Fondos Insuficientes en ATM
5. Confirmar Disponibilidad de Fondos	A5 Fondos Insuficientes en cuenta
6. Retornar Tarjeta	A6 Causaría Sobregiro
7. Entregar Efectivo	A7 Tarjeta Atascada
	A8 Efectivo dejado atrás
	etc..

**Figura 2: La estructura de la narrativa de un caso de uso**

Los casos de uso proporcionan este enfoque concentrándose en cómo se empleará el sistema para alcanzar las metas específicas de un usuario particular. Estos casos de uso encierran muchas formas de usar el sistema: aquellas que alcanzan exitosamente las metas del sistema y aquellas que manejan los problemas que pueden ocurrir. Para cuantificar, identificar y entregar fácilmente el valor, es necesario estructurar la narrativa del caso de uso. Para mantener las cosas simples, inicie con el camino más fácil para conseguir la meta. Luego, capture todas las formas alternativas de alcanzar la meta y cómo manejar cualquier problema que pueda ocurrir mientras trata de lograr el objetivo. Esto hará que las relaciones entre las formas de emplear el sistema estén claras. Además, permitirá que se identifiquen y evolucionen las formas de más valor y también que las de menos valor se adicionen posteriormente sin estropear o cambiar lo que ya se hizo.

En algunos casos habrá poco o ningún valor al implementar algo más allá de la forma más simple de alcanzar la meta. En otros casos, proporcionar más opciones y formas especializadas de lograr la meta suministrará los diferenciadores clave que harán su sistema más valioso que el de sus competidores.

La **Figura 2** muestra la narrativa de un caso de uso estructurada de esta manera. La forma más simple de lograr el objetivo se describe en el flujo básico. Las otras se presentan como flujos alternativos. De esta manera, se crea un conjunto de flujos que estructuran y describen las historias y ayudan a encontrar los casos de prueba que complementan su definición.

La **Figura 2** muestra la estructura de la narrativa para el caso de uso Retirar Fondos de un cajero automático. El flujo básico se presenta como un conjunto de pasos simples que capturan la interacción entre los usuarios y el sistema. Los flujos alternativos identifican todas las demás formas de usar el sistema para lograr metas como preguntar por una cantidad no-estándar, cualquier facilidad opcional que se pueda ofrecer al usuario (como entregar un recibo) y cualquier problema que pueda ocurrir en el camino como por ejemplo que la tarjeta se atasque en la máquina.

No es necesario capturar todos los flujos al mismo tiempo. Mientras se registra el flujo básico, es natural pensar en las otras formas de conseguir el objetivo y en lo que puede resultar mal en cada paso. Usted



---

captura esto como Flujos Alternativos, pero se concentra en el Flujo Básico. Siempre es posible regresar a completar los flujos alternativos a medida que se necesiten.

Esta clase de esquema con viñetas puede ser suficiente para capturar las historias y conducir el desarrollo, o puede requerir de más elaboración a medida que el equipo explora el detalle de lo que el sistema necesita hacer. El aspecto más importante es la estructura aditiva de la narrativa del caso de uso. El flujo básico es necesario si el caso de uso se va a completar exitosamente en algún momento; este flujo básico se debe implementar primero. Las alternativas, sin embargo, son opcionales y se pueden agregar al flujo básico a medida que se requieran. Esto permite que el equipo realmente se enfoque en el valor a obtener del sistema. No es necesario entregar todo el caso de uso, sino enfocarse en aquellas partes del caso de uso que ofrecen más valor. También significa que no es necesario un modelo de casos de uso completo o aún un caso de uso completo antes de empezar a trabajar en el desarrollo del sistema. Si se identificó el caso de uso más importante y se entendió su flujo básico, entonces ya se tiene algo de valor que se podría adicionar al sistema.

Esta estructura hace que las historias sean fáciles de capturar y su plenitud fácil de validar, a la vez que facilita filtrar las formas potenciales de emplear el sistema que ofrezcan poco o ningún valor real a los usuarios. Este foco constante en el valor permitirá asegurar que cada entrega del sistema sea tan pequeña como sea posible, a la vez que tenga valor real para los usuarios del sistema y para los interesados que financian el desarrollo.

## Principio 4: Construir el sistema por partes

La mayoría de los sistemas requieren mucho trabajo antes de que sean usables y estén listos para uso operacional. Tienen muchos requisitos, la mayoría de los cuales dependen de otros requisitos que se están implementando antes de que se puedan satisfacer y entregar con valor. Siempre es un error intentar construir un sistema de una sola vez. El sistema se debería construir por partes, cada una de las cuales tiene un valor claro para sus usuarios.

La receta es bien simple. Primero, identifique el aspecto más útil que el sistema debe hacer y enfóquese en eso. Luego, tome ese aspecto y divídalo en porciones más pequeñas. Defina los casos de prueba que representen la aceptación de cada una de esas porciones. Hágalos a un lado por el momento. Seleccione la porción más importante que viaje por todo el concepto de principio a fin, o lo más cercano a eso que se pueda. Estime el desarrollo en equipo (las estimaciones no tienen que estar “correctas”, son simples estimados) y comience a construir.

Esta es la estrategia que siguen Casos de Uso 2.0, donde los casos de uso se dividen para proveer elementos de trabajo de tamaño adecuado y donde el sistema mismo evoluciona porción por porción.

### Dividir los casos de uso

La mejor forma de encontrar las porciones correctas es pensar en las historias y en cómo las capturamos. Cada historia es una buena porción candidata. Cada historia se define con una parte de la narrativa del caso de uso y uno o más de los casos de prueba acompañantes. Son los casos de prueba la parte más importante de la descripción de esta porción del caso de uso, porque son los que hacen reales las historias.

Aplicando nuestra receta de arriba, los casos de uso identifican las cosas útiles que el sistema hará. Seleccione el caso de uso más útil para encontrar la cosa más útil que el sistema hace. Para encontrar la porción más útil usted necesitará hacer a un lado todas las formas menos importantes de lograr el objetivo y de manejar los problemas. Esto se puede hacer enfocándose en la historia que describe el flujo básico. Una porción basada en el flujo básico se garantiza que viaja por todo el concepto, de principio a fin, puesto que este será el camino más simple para el usuario de alcanzar su meta.



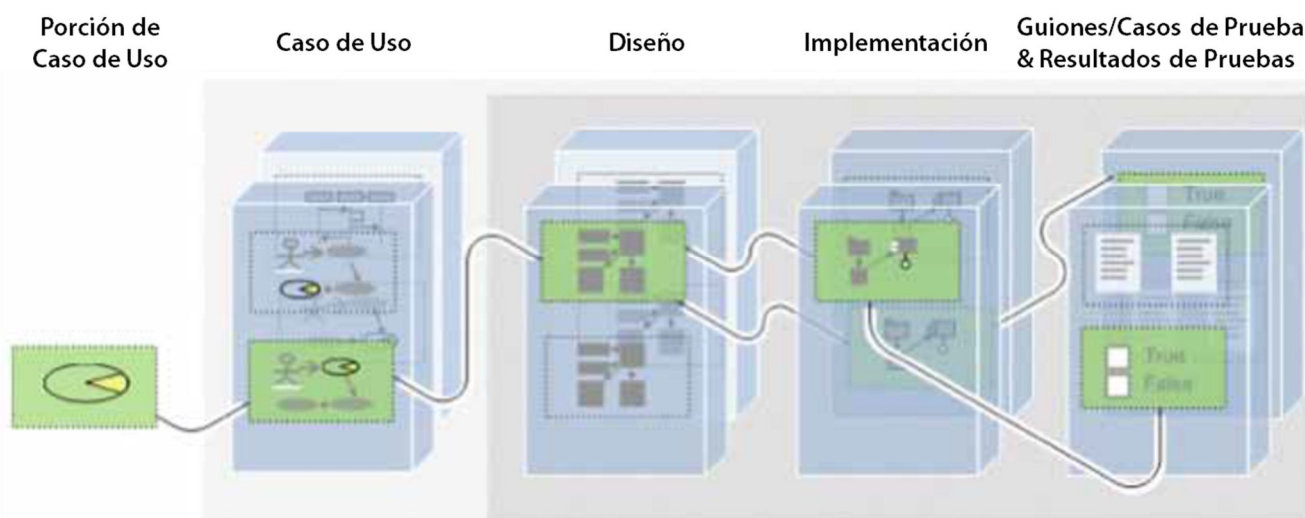
Estime la porción y comience a construirla. Las porciones adicionales se pueden tomar del caso de uso hasta que haya suficientes porciones para entregar una solución usable a este usuario en particular. Lo mismo se puede hacer para cualquier otro caso de uso que sea necesario para completar un sistema usable.

Una porción de caso de uso no necesariamente contiene un flujo completo y todos sus casos de prueba: la primera porción puede ser tan solo el flujo básico y un caso de prueba. Porciones adicionales se pueden agregar luego para completar el flujo y cubrir todos los casos de prueba. El mecanismo de partición es muy flexible, permitiendo que se creen porciones tan grandes o tan pequeñas como sea necesario para conducir el desarrollo.

### Las porciones son más que solo requisitos y casos de prueba

Cuando construimos el sistema por partes, no es suficiente con solo dividir los requisitos. Aunque tradicionalmente los casos de uso se usan para ayudar a entender y a capturar los requisitos, siempre fueron más que esto. Como se muestra en la **Figura 3**, las porciones de casos de uso dividen más que los requisitos, también dividen todos los otros aspectos del sistema y su documentación.

En el lado izquierdo de la **Figura 3** se puede ver la porción del caso de uso, que se tomó de uno de los casos de uso que se muestran en la siguiente columna. La porción luego continúa hacia el diseño, mostrando los elementos de diseño involucrados, y hacia la implementación, donde se puede ver cuales piezas del código realmente implementan la porción. Finalmente, la porción se corta en activos de prueba, no solo abarcando los casos de prueba, sino también los guiones de prueba usados para ejecutar los casos de prueba y los resultados de prueba generados.



**Figura 3: Una porción de caso de uso es más que solo una porción de los requisitos**

Así como se provee trazabilidad desde los requisitos hasta el código y las pruebas, pensar en las porciones de esta manera ayuda a desarrollar el sistema correcto. Cuando se va a implementar una porción, es necesario entender el impacto que la porción tendrá en el diseño y en la implementación del sistema. ¿Necesita esta porción que se adicionen nuevos elementos del sistema? ¿Se puede implementar con solo hacer cambios a los elementos existentes? Si el impacto es muy grande, es posible tomar la decisión de no implementar esa porción. Si se tiene el diseño básico para el sistema, este tipo de análisis se puede hacer fácil y rápidamente, a la vez que permite una manera estupenda de entender el impacto de adicionar esa porción al sistema.

Tratando cada aspecto del sistema porción por porción, los casos de uso ayudan en todas las áreas del sistema, incluyendo la experiencia de usuario (interfaz de usuario), arquitectura, pruebas y planeación. Las porciones proporcionan una forma de vincular los requisitos con las partes del sistema que los implementan, las pruebas se emplean para verificar que los requisitos se implementen exitosamente, al igual que la versión y los planes de proyecto que conducen el trabajo de desarrollo. En los Casos de Uso

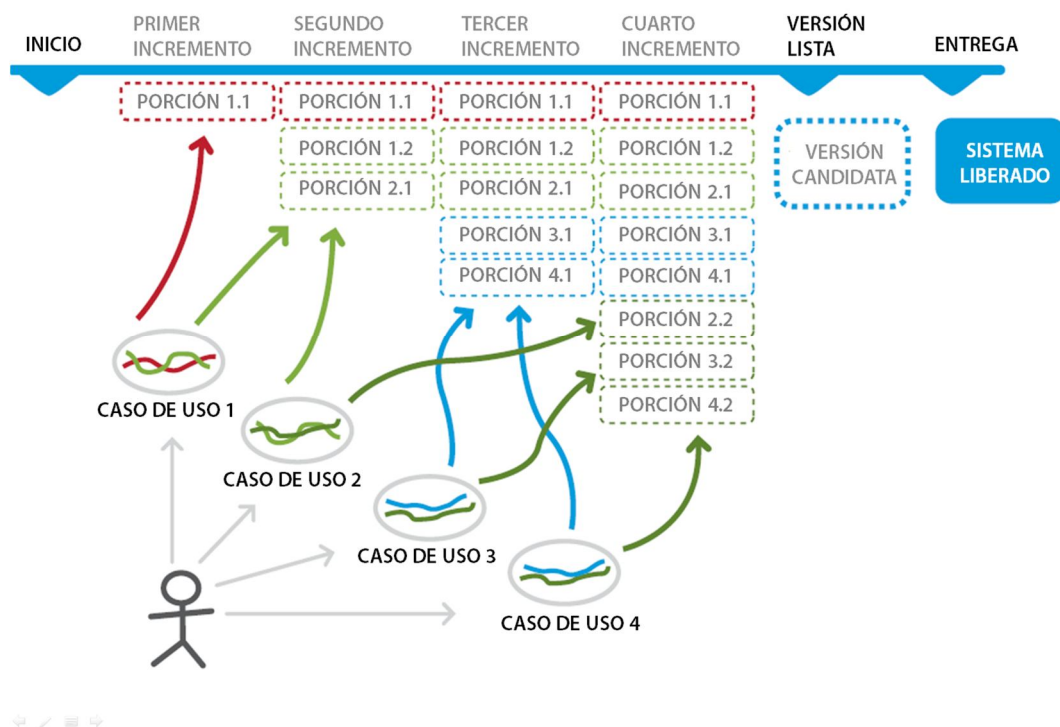
2.0 hay una construcción especial, llamada la realización del caso de uso, la cual se adiciona a cada caso de uso para registrar su impacto en los demás aspectos del sistema.

### Porciones de Casos de Uso: La parte más importante de los Casos de Uso 2.0

El concepto de una porción de caso de uso es tan integral a los Casos de Uso 2.0 como el caso de uso mismo. Son las porciones las que posibilitan que los casos de uso se dividan en piezas de tamaño apropiado para que las atienda el equipo de desarrollo. Imagine que usted hace parte de un equipo pequeño que está produciendo software funcional cada dos semanas. Un caso de uso completo es probablemente mucho para completar en un período de dos semanas. Sin embargo, una porción de caso de uso es otra cosa, porque se puede dividir en una parte tan pequeña como el equipo requiera. Las porciones de casos de uso también permiten que el equipo se enfoque en entregar un sistema usable, con valor, tan pronto como sea posible, dejando en el camino todos los requisitos innecesarios.

## Principio 5: Entregar el sistema en incrementos

La mayoría de los sistemas de software evolucionan a lo largo de muchas generaciones. No se producen de una sola vez, sino que se construyen como una serie de versiones, cada una de las cuales se implementa sobre la anterior. Aun las versiones mismas no se producen de una sola vez, sino que evolucionan luego de una serie de incrementos. Cada incremento arroja una versión demostrable o usable del sistema. Cada incremento se construye sobre el incremento previo para adicionar más funcionalidad o mejorar la calidad de lo que se hizo antes. Esta es la forma en que se deberían producir todos los sistemas.



**Figura 4: Casos de Uso, Porciones de Casos de Uso, Incrementos y Versiones**

También, puede ser demasiado considerar la entrega de todos los casos de uso al mismo tiempo. Por ejemplo, probablemente no necesitemos todas las formas de hacer una llamada local en el primerísimo incremento de un sistema telefónico. Las facilidades más básicas pueden ser suficientes para que el sistema se ponga en marcha. Las formas opcionales de hacer una llamada, como revertir los cargos o remarcar el último número, se pueden adicionar en incrementos posteriores. Al fraccionar los casos de uso, podemos lograr el control más detallado requerido para maximizar el valor en cada versión.

La **Figura 4** muestra el desarrollo incremental de una versión de un sistema. El primer incremento solo contiene una porción simple: la primera porción del caso de uso 1. El segundo incremento adiciona otra porción del caso de uso 1 y la primera porción del caso de uso 2. Se agregan porciones adicionales luego para crear el tercer y cuarto incrementos. El cuarto incremento se considera completo y suficientemente útil para liberarlo.

Los casos de uso constituyen una herramienta fabulosa para planear una versión. Trabajar a nivel de casos de uso permite que grupos completos de requisitos relacionados se difieran hasta las últimas entregas. Tomar decisiones a nivel de casos de uso permite bosquejar rápidamente el panorama general y usar esto para enfocarse en las áreas del sistema que se abordarán en la próxima versión.

Los diagramas de casos de uso, al mostrar cuales casos de uso se van a tratar en esta versión y cuales casos de uso se relegarán hasta una versión posterior, constituyen una gran herramienta para ilustrar las metas del equipo. Estos diagramas muestran claramente el tema de cada versión y lucen muy bien sujetos en la pared de su centro de operaciones para que todos lo vean.

Las porciones de los casos de uso constituyen una herramienta fabulosa para construir incrementos más pequeños mientras se avanza hacia una versión completa. Esta herramienta permite adicionar a los incrementos porciones independientes que se puedan implementar y verificar, asegurando que cada incremento sea más grande que el anterior y se construya sobre el.

## Principio 6: Adaptarse para cubrir las necesidades del equipo

Desafortunadamente, no hay una solución “única” para todos los desafíos del desarrollo de software: equipos diferentes y situaciones diferentes requieren estilos diferentes y niveles diferentes de detalle. Sin importar cuales prácticas y técnicas se seleccionan, es necesario asegurarse de que estas son lo suficientemente adaptables para cubrir las necesidades actuales del equipo.

Esto aplica tanto a las prácticas que se seleccionan para compartir los requisitos y conducir el desarrollo del software como a cualquiera otra. Por ejemplo, los requisitos livianos son increíblemente efectivos cuando hay una colaboración cercana con los usuarios, tal que el equipo de desarrollo puede obtener explicaciones personales de los requisitos y obtener a tiempo respuestas a cualquier pregunta que surja. Si esta clase de colaboración no es posible, porque los usuarios no están disponibles, entonces los requisitos deberán tener más detalle e, inevitablemente, pesarán más. Hay muchas otras circunstancias donde un equipo podría necesitar requisitos más detallados como insumo para el desarrollo. Sin embargo, lo que importa no es la lista de todas las posibles circunstancias donde una estrategia liviana no sea adecuada, sino reconocer el hecho de que es necesario escalar las prácticas.

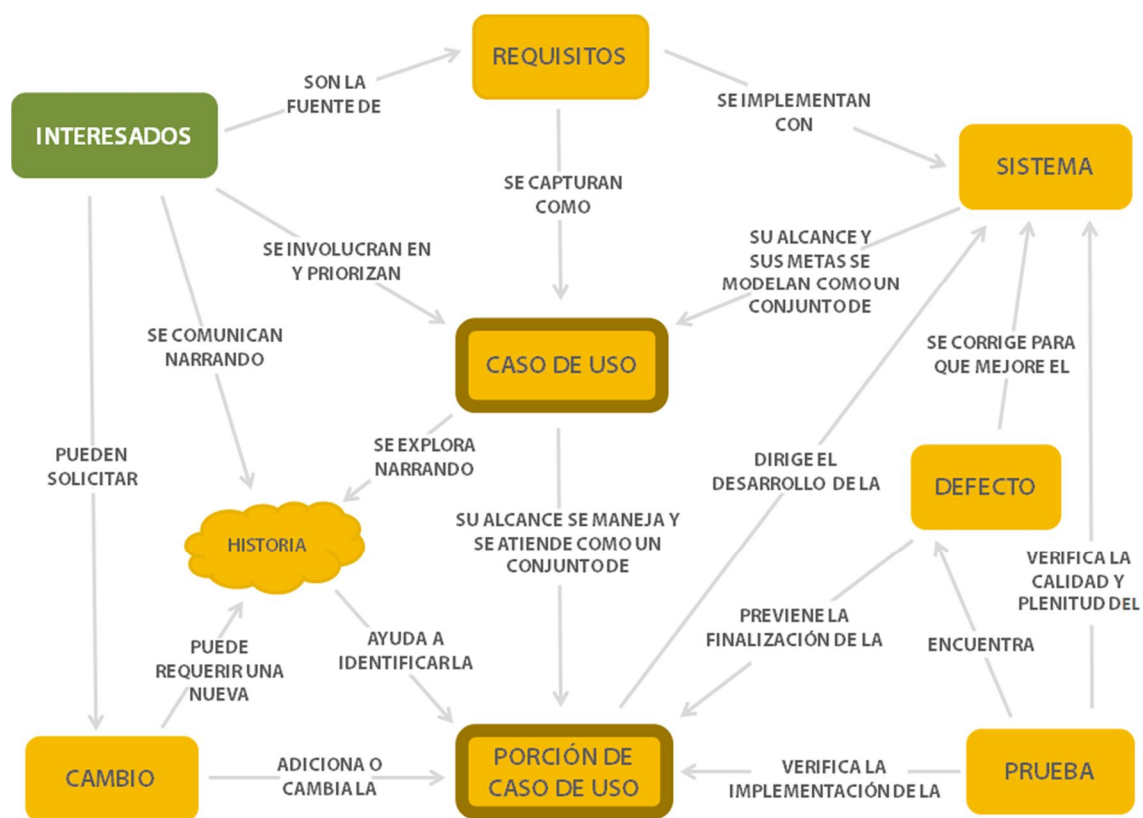
Los Casos de Uso 2.0 se diseñaron con esto en mente y son tan livianos como se requiera que sean. Equipos pequeños y colaborativos pueden tener narrativas de casos de uso muy livianas que capturen lo estrictamente necesario de las historias. Estas se pueden escribir a mano en simples tarjetas indexadas. Equipos distribuidos grandes pueden tener narrativas de casos de uso más detalladas y que se presenten como documentos. Depende del equipo decidir si es necesario ir más allá de lo esencial, adicionando detalles de una manera natural a medida que se encuentren problemas que lo esencial no pueda cubrir.

# Contenido de los Casos de Uso 2.0

Los Casos de Uso 2.0 constan de un conjunto de cosas con las cuales trabajar y un conjunto de cosas por hacer.

## Cosas con las cuales trabajar

El asunto de los Casos de Uso 2.0 lo constituyen: los requisitos, el sistema que se construya para satisfacer los requisitos y las pruebas usadas para demostrar que el sistema satisface los requisitos. En el corazón de los Casos de Uso 2.0 están el caso de uso, la historia y la porción de caso de uso. Estos elementos capturan los requisitos y conducen el desarrollo del sistema. La **Figura 5** muestra como estos conceptos se relacionan entre sí. También, muestra como los cambios y los defectos impactan el empleo de los Casos de Uso 2.0.



**Figura 5: Mapa Conceptual de Casos de Uso 2.0**

Los interesados son las personas, grupos u organizaciones que impactan o se impactan con un sistema de software. Los requisitos constituyen lo que el sistema debe hacer para satisfacer a los *interesados*. Es importante descubrir lo que es necesario del sistema de software, compartir este entendimiento con los interesados y con los miembros del equipo y usarlo para conducir el desarrollo del nuevo sistema. En los Casos de Uso 2.0, los requisitos se capturan como un conjunto de casos de uso, los cuales se manejan y se atienden como un conjunto de porciones de casos de uso. Cualquier cambio que requieran los

interesados genera adiciones o cambios al conjunto de casos de uso y a las porciones de los casos de uso.

El sistema es el sistema que se va a construir. Es, típicamente, un sistema de software, aunque los Casos de Uso 2.0 se pueden también emplear en el desarrollo de nuevos negocios (donde el negocio mismo se trata como un sistema) y combinarlos con sistemas de hardware y software (sistemas embebidos). El sistema es el que implementa los requisitos y es el sujeto del modelo de casos de uso. La calidad y plenitud del sistema se verifican mediante un conjunto de pruebas. Las pruebas también verifican si la implementación de las porciones de casos de uso fue exitosa o no. Si se encuentran defectos durante las pruebas, entonces su presencia prevendrá la finalización de la porción del caso de uso hasta que estos se corrijan y el sistema mejore.

La narración de historias reduce la brecha entre los interesados, los casos de uso y las porciones de casos de uso. Esta es la forma como los interesados comunican sus requisitos y exploran los casos de uso. Entender las historias también es el mecanismo para encontrar las porciones de caso de uso correctas para conducir la implementación del sistema.

## Casos de uso

Un caso de uso se materializa con todas las formas de emplear un sistema para alcanzar una meta particular para un usuario particular. El conjunto de todos los casos de uso nos da todas las formas útiles de emplear el sistema.

Un caso de uso es:

- Una secuencia de acciones que un sistema ejecuta y que arroja un resultado observable de valor para un usuario particular.
- Ese comportamiento específico de un sistema, que participa en una colaboración con un usuario para entregar algo de valor para ese usuario.
- La unidad más pequeña de actividad que proporciona un resultado significativo para el usuario.
- El contexto para un conjunto de requisitos relacionados.

Para entender un caso de uso narramos historias. Las historias cubren tanto la forma como se alcanza la meta de manera exitosa y la forma como se maneja cualquier problema que ocurre en el camino. Las historias nos ayudan a entender el caso de uso y lo implementan porción por porción.

Como se muestra en la **Figura 6**, un caso de uso experimenta varios cambios de estado desde su identificación inicial hasta su cumplimiento en el sistema. Los estados constituyen puntos importantes en la comprensión e implementación del caso de uso, indicando:

1. Meta Establecida: cuando se establece la meta del caso de uso.
2. Estructura de la Historia Entendida: cuando el equipo entiende la estructura de la narrativa del caso de uso lo suficiente para empezar el trabajo de identificar e implementar las porciones de los casos de uso.



Figura 6: El Ciclo de Vida de un Caso de Uso



3. La Historia Más Simple Cumplida: cuando el sistema cumple la historia más simple que permita al usuario alcanzar la meta.
4. Suficientes Historias Cumplidas: cuando el sistema cumple suficientes historias para proporcionar una solución usable.
5. Todas las Historias Cumplidas: cuando el sistema cumple todas las historias que narra el caso de uso.

Esto se logrará implementando el caso de uso porción por porción. Los estados proporcionan una forma simple de evaluar el progreso que se está logrando, entendiendo e implementando con el caso de uso.

### Porciones de Casos de uso

Los casos de uso cubren muchas historias relacionadas de importancia y prioridad variada. Muchas veces hay demasiadas historias para entregar en una sola liberación y, generalmente, son muchas para trabajar en un solo incremento. Debido a esto, necesitamos una forma de dividir los casos de uso en piezas más pequeñas que nos permitan: 1) seleccionar las piezas del caso de uso a entregar, 2) proporcionar una unidad adecuada para el desarrollo y las pruebas del equipo de desarrollo y 3) tener piezas de trabajo pequeñas y de tamaño similar que fluyan rápidamente a lo largo del desarrollo.

Una porción de caso de uso la conforman una o más historias seleccionadas de un caso de uso para formar un elemento de trabajo que proporcione valor claro para el usuario. Esta actúa como un marcador para todo el trabajo requerido para completar la implementación de las historias seleccionadas. Como vimos antes, cuando discutimos como las porciones de casos de uso son más que solo requisitos y casos de prueba, la porción de caso de uso evoluciona para incluir las porciones correspondientes a lo largo del diseño, la implementación y la prueba.

La porción de caso de uso es el elemento más importante de Casos de Uso 2.0, puesto que no solo se usa para ayudar con los requisitos, sino también para conducir el desarrollo de un sistema que los satisfaga.

Las porciones de casos de uso:

- Posibilitan que los casos de uso se dividan en unidades de trabajo más pequeñas, entregables independientemente.
- Posibilitan que los requisitos contenidos en un conjunto de casos de uso se ordenen, prioricen y traten en paralelo.
- Vinculan los distintos modelos del sistema (requisitos, análisis, diseño, implementación y pruebas) usados en el desarrollo dirigido por casos de uso.

Como lo muestra la **Figura 7**, una porción de caso de uso sufre varios cambios de estado desde su identificación inicial hasta su aceptación final. Los estados constituyen puntos importantes en el entendimiento, implementación y prueba de la porción de caso de uso, indicando:



Figura 7: El Ciclo de Vida de una Porción de Caso de Uso

1. Alcance definido: cuando se define el alcance y se clarifica la extensión de las historias cubiertas.
2. Preparada: cuando se prepara la porción, ampliando la narrativa y los casos de prueba para definir claramente lo que significa la implementación exitosa de la porción.
3. Analizada: cuando se analiza la porción de tal forma que se entiende su impacto en los componentes del sistema y las piezas afectadas están listas para codificación y pruebas del desarrollador (unitarias).
4. Implementada: cuando se mejora el sistema de software para implementar la porción, que queda lista para pruebas.
5. Verificada: finalmente, cuando se verifica la porción y queda lista para su inclusión en una versión.

Los estados proporcionan una forma simple de evaluar el progreso que usted está haciendo para entender e implementar las porciones de casos de uso. También, denotan los puntos donde la porción está potencialmente lista para pasar de una persona o equipo a otro. Para el observador casual que echa un vistazo a los estados, esto podría parecer como un proceso en cascada: definición de alcance>preparación>análisis>implementación>verificación. Sin embargo, hay una diferencia significativa. En una estrategia de cascada, todos los requisitos se preparan antes que el análisis comience y todo el análisis se completa antes de iniciar la implementación y toda la implementación se completa antes de que la verificación inicie. Aquí estamos tratando con una porción individual de un caso de uso. A lo largo del conjunto de porciones, todas las actividades podrían ir en paralelo. Mientras que una porción de caso de uso se verifica, otra porción de caso de uso se implementa, una tercera se diseña y una cuarta se analiza. En el siguiente capítulo exploraremos esto más en detalle cuando miremos el empleo de Casos de Uso 2.0 con diferentes estrategias.

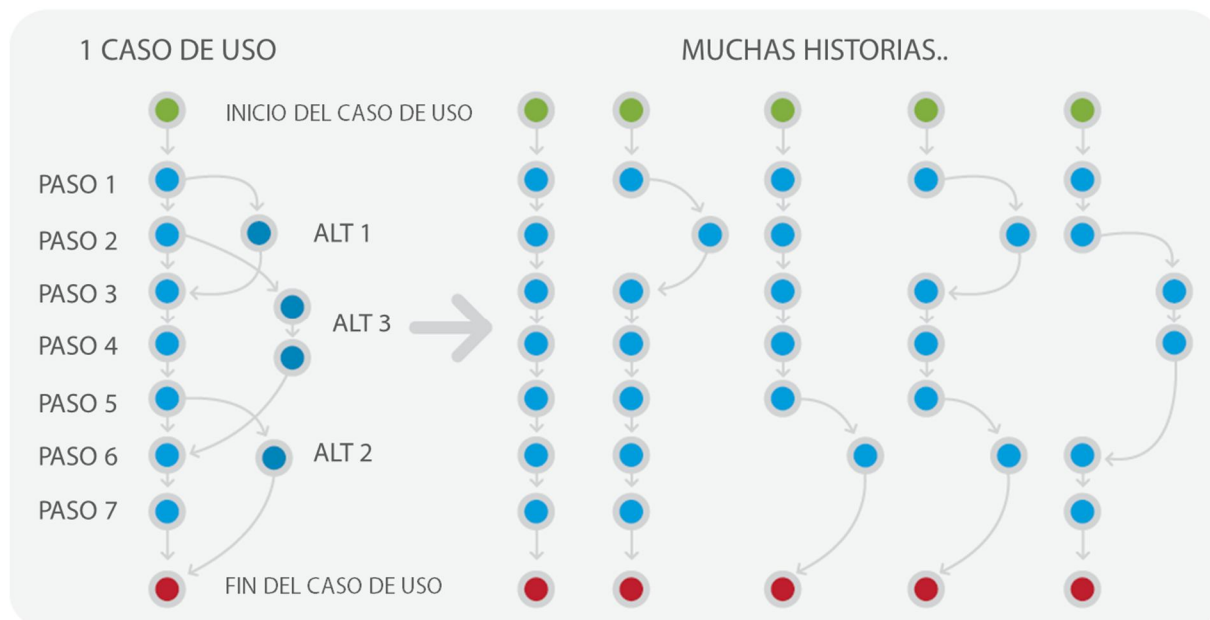
## Historias

Narrando historias es como exploramos los casos de uso con nuestros interesados. Cada historia de valor para los usuarios y otros interesados es un hilo que atraviesa uno de los casos de uso. Las historias pueden ser funcionales o no funcionales en su naturaleza.

Una historia se describe mediante parte de la narrativa del caso de uso, uno o más flujos y requisitos especiales y uno o más casos de prueba. La clave para encontrar historias es entender la estructura de la narrativa del caso de uso. La red de flujos se puede ver como un mapa que resume todas las historias necesarias para describir el caso de uso. La **Figura 8** ilustra la relación entre los flujos de la narrativa de un caso de uso y las historias que esta describe.

En el lado izquierdo de la figura, el flujo básico se presenta como una secuencia lineal de pasos y los flujos alternativos se presentan como un desvío de este conjunto de pasos. Los flujos alternativos siempre se definen como variaciones del flujo básico. En el lado derecho del diagrama se presentan algunas de las historias que los flujos cubren. Cada historia atraviesa uno o más flujos, iniciando con el caso de uso al final del flujo básico. Esto asegura que todas las historias se relacionen con la consecución de la misma meta, se completen y sean significativas y complementarias, ya que todas se construyen sobre la historia simple que describe el flujo básico. Para cada historia habrá uno o más casos de prueba.





**Figura 8: La Relación Entre los Flujos y las Historias**

Hay dos enfoques comunes para identificar las historias y crear la narrativa del caso de uso:

- **Arriba-Abajo:** a algunas personas les gusta hacer una aproximación *top-down*, donde 1) identifican el caso de uso, 2) resumen los pasos del flujo básico y 3) realizan una lluvia de ideas para identificar los flujos alternativos en el flujo básico. Esto estructura la narrativa y permite identificar las historias.
- **Abajo-Arriba:** usando la aproximación *bottom-up* empezamos por hacer una tormenta de ideas con algunas de las historias y luego las agrupamos por tema para identificar nuestros casos de uso. El conjunto de historias se revisa más tarde para ayudarnos a identificar el flujo básico y algunos de los flujos alternativos. La estructura del caso de uso nos conduce luego a identificar cualquier historia faltante y nos asegura que todas las historias se formen bien y sean complementarias.

Usted debería seleccionar el método que trabaje mejor para sus interesados. También es posible combinar los dos métodos: trabajar tanto *top-down*, desde sus casos de uso, como *bottom-up* desde cualquier nueva historia sugerida.

Las historias son herramientas útiles de pensamiento para ayudarnos a encontrar las porciones correctas de los casos de uso y, más importante aun, los casos de prueba correctos. No necesitamos rastrear el estado de las historias mismas, puesto que es la ejecución de los casos de prueba la que nos muestra que tan completo está el sistema y el progreso de los casos de uso y las porciones de los casos de uso que dirigen el desarrollo.

## Defectos y Cambios

Aunque no es una parte directa de los Casos de Uso 2.0, es importante entender como los defectos y los cambios se relacionan con los casos de uso y las porciones de casos de uso.

Los cambios que requieren los interesados se analizan contra el modelo de casos de uso actual, los casos de uso y las porciones de casos de uso. Esto posibilita la comprensión rápida del alcance del cambio. Por ejemplo, adicionar un nuevo caso de uso al sistema es un cambio mayor, puesto que esto cambia las metas generales y propósitos del sistema; mientras tanto, un cambio a un caso de uso existente es típicamente mucho más pequeño, particularmente si es a una historia que aún no se localiza, prepara, analiza, implementa o verifica en una porción.

Los defectos se manejan rastreando cuales porciones de casos de uso y, consecuentemente, cuales casos de prueba resultaron en su detección. Si se encuentran durante la implementación o verificación de una



productos de trabajo los documentan completamente. Si se requiere, se pueden listar las historias relacionadas con un caso de uso como una sección extra en la narrativa del caso de uso, pero esto no es esencial.

## Trabajar con Casos de Uso y Porciones de Caso de Uso

A medida que se crean y rastrean los productos de trabajo, también se necesitará rastrear los estados y las propiedades de los casos de uso y de las porciones de los caso de uso. Esto se puede hacer de muchas maneras y con muchas herramientas. Los estados se pueden rastrear simplemente usando notas adhesivas u hojas de cálculo. Si se requiere más formalidad, se puede emplear una de las muchas herramientas de administración de requisitos, administración de cambios o manejo de defectos comercialmente disponibles.

La **Figura 10** muestra un caso de uso y algunas de sus porciones capturadas en un conjunto de notas adhesivas.



**Figura 10: Capturar las Propiedades de un Caso de Uso y sus Porciones Usando Notas Adhesivas.**

El caso de uso mostrado es el caso de uso '7. Revisar y Comprar' de una aplicación de compra en línea. Las porciones 1 y 2 del caso de uso se basan en historias individuales del flujo básico. 'Seleccionar y Comprar 1 Producto' y 'Seleccionar y Comprar 100 Productos'. La porción 3 se basa en historias múltiples, cubriendo la disponibilidad de los diferentes sistemas de soporte involucrados en el caso de uso. Estas historias cubren varios flujos alternativos.

Las propiedades esenciales para un caso de uso son su nombre, estado y prioridad. En este caso se usa el popular esquema de priorización MoSCoW, cuyo nombre proviene de los auxiliares en inglés *Must* (Debe), *Should* (Debería), *Could* (Podría) y *Would* (Pudiera). Los casos de uso también se deberían estimar. En este caso, se utiliza un simple esquema de evaluación de su tamaño y complejidad relativos.

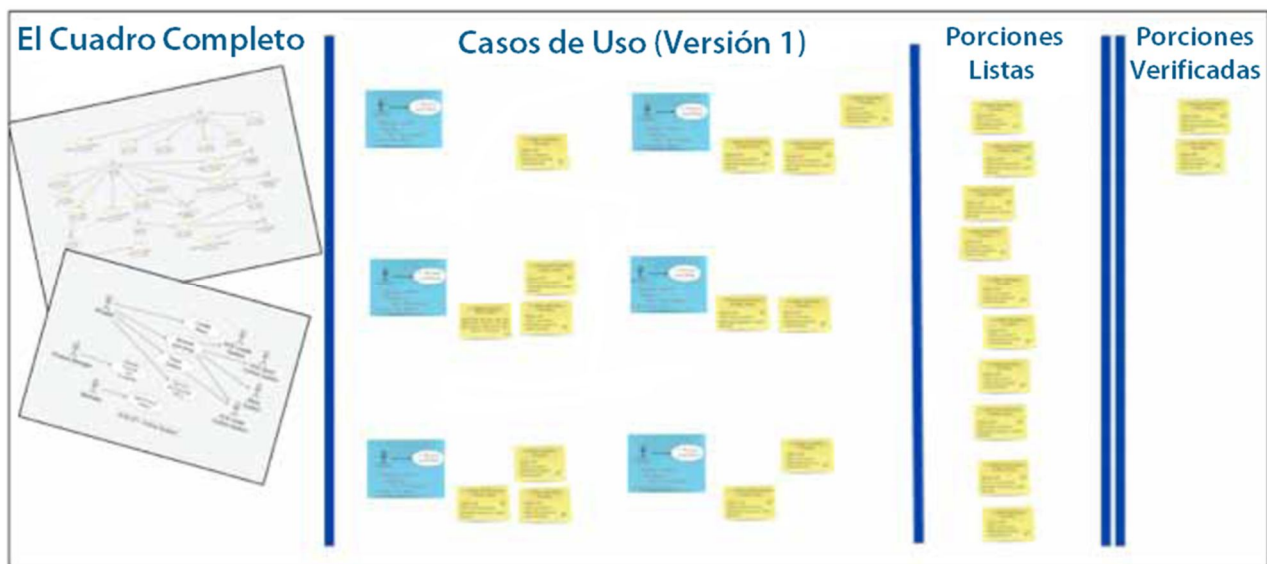
Las propiedades esenciales de una porción de caso de uso son: 1) una lista de sus historias, 2) referencias al caso de uso y los flujos que definen las historias, 3) referencias a las pruebas y casos de pruebas que se emplearán para verificar su plenitud y 4) un estimado del trabajo necesario para implementar y probar la porción. En este ejemplo, las historias se usan para nombrar la porción y las referencias al caso de uso están implícitas en los números de las porciones y lista de flujos. Los estimados se agregan más tarde, después de consultar al equipo. Estos son los números grandes que se agregan en la parte inferior derecha de la nota adhesiva. En este caso, el equipo juega un '*poker de planeación*' para crear estimados relativos usando puntos de historia; 5 puntos de historia para las porciones 7.1 y 7.2 y 13 puntos de historia para la porción 7.3 del cual el equipo cree que tomará más del doble del esfuerzo que las otras

porciones. Alternativamente, se podrían usar las técnicas de días ideales, tallas de camiseta (XS, S, M, L, XL, XXL, XXXL) o cualquier otra técnica popular de estimación.

Los casos de uso y las porciones de los casos de uso también se pueden ordenar de tal forma que los más importantes se traten primero. La **Figura 11** muestra como estas notas adhesivas se pueden usar para construir una simple lista de pendientes de producto en un tablero. Leyendo de derecha a izquierda, es posible ver 1) el cuadro completo ilustrado con diagramas de casos de uso que muestran el alcance del sistema completo y la primera versión, 2) los casos de uso seleccionados para la primera versión y algunas de sus porciones, las cuales se identificaron pero aún no se detallan ni ordenan, 3) la lista ordenada de porciones, que se incluirá en el desarrollo de la versión y, finalmente, 4) las porciones que el equipo implementa y verifica exitosamente.

La **Figura 11** se incluye solo para propósitos ilustrativos, pues hay muchas otras formas de organizar y trabajar con sus requisitos. Por ejemplo, muchos equipos se preocupan porque sus notas adhesivas no se caigan del tablero. Estos equipos muchas veces rastrean el estado de sus casos de uso y porciones de casos de uso usando una hoja de cálculo simple, tal como las que se muestran en las **Figuras 12 y 13**.

Estas ilustraciones se incluyen para ayudarle a empezar. Los casos de uso y las porciones de casos de uso son centrales a cualquier cosa que el equipo haga, así que emplee cualquier técnica que usted necesite para hacerlos tangibles y visibles al equipo. Siéntase libre de agregar otros atributos a la medida y cuando lo requiera, por ejemplo, para registrar la fuente del caso de uso o su propietario, o para asignar las porciones de caso de uso a un incremento particular dentro de la versión.



**Figura 11: Emplear Casos de Uso y Porciones de Caso de Uso Para Construir una Lista de Pendientes de Producto.**

CASO DE USO	NOMBRE	VERSIÓN	PRIORIDAD	ESTADO	TAMAÑO	COMPLEJIDAD
7	REVISAR Y COMPRAR	1	1-REQUERIDO	ESTRUCTURA DE HISTORIA ENTENDIDA	MUY GRANDE	ALTA
14	CONSEGUIR NUEVAS OFERTAS	1	1-REQUERIDO	ESTRUCTURA DE HISTORIA ENTENDIDA	MEDIANA	MEDIA
17	MANTENER PRODUCTOS Y DISPONIBILIDAD	1	1-REQUERIDO	ESTRUCTURA DE HISTORIA ENTENDIDA	GRANDE	ALTA
12	RASTREAR ORDENES	1	2-IMPORTANTE	ESTRUCTURA DE HISTORIA ENTENDIDA	GRANDE	BAJA
13	LOCALIZAR ALMACEN	1	2-IMPORTANTE	ESTRUCTURA DE HISTORIA ENTENDIDA	PEQUEÑA	BAJA
16	FIJAR OFERTAS DE PRODUCTOS	1	2-IMPORTANTE	ESTRUCTURA DE HISTORIA ENTENDIDA	MEDIANA	ALTA
11	CONSULTAR COMPRAS	1	3-OPCIONAL	ESTRUCTURA DE HISTORIA ENTENDIDA	PEQUEÑA	MEDIA
1	CONSTRUIR CASA			META ESTABLECIDA		
2	DISEÑAR INTERIORES			META ESTABLECIDA		
3	CONSTRUIR JARDIN			META ESTABLECIDA		
4	SEMBRAR EN EL JARDIN			META ESTABLECIDA		

**Figura 12: La Hoja de Cálculo del Caso de Uso de un Rastreador Simple de Casos de Uso.**



CASO DE USO	PORCIÓN	HISTORIAS	FLUJO	CASOS DE PRUEBA	ESTADO	ORDEN	ESTIMADO (PUNTOS DE HISTORIA)	VERSIÓN OBJETIVO
7 REVISAR Y COMPRAR	7.1	SELECCIONAR Y COMPRAR 1 PRODUCTO	BF	7.1.1	PREPARADO	1	13	1
7 REVISAR Y COMPRAR	7.2	SELECCIONAR Y COMPRAR N PRODUCTOS	BF	7.2.1, 7.2.3	PREPARADO	2	13	1
7 REVISAR Y COMPRAR	7.4	MANEJAR DETALLES DE PAGO Y ENTREGA	A4, A5, A6	7.3.1, 7.3.2	DEFINIDO	3	3	1
17 MANTENER PRODUCTOS Y DISPONIBILIDAD	17.1	CREAR NUEVO PRODUCTO	BF	17.1.1, 17.1.2	DEFINIDO	4	20	1
1 CONSTRUIR CASA	1.1	AGREGAR PRIMERA CASA	BF	1.1.1, 1.1.2, 1.1.3	DEFINIDO	5	5	1
2 DISEÑAR INTERIORES	2.1	DISEÑAR CUARTO	BF, A3		VERIFICADO	6	5	1
2 DISEÑAR INTERIORES	2.4	COMPRAR CONTENIDO	A6		DEFINIDO	7	3	1
7 REVISAR Y COMPRAR	7.5	MANEJAR PÉRDIDA DE SISTEMAS DE APOYO	A9, A11, A12		DEFINIDO	8	5	1
7 REVISAR Y COMPRAR	7.6	PRODUCTO SIN INVENTARIO	A7		DEFINIDO	9	13	1
17 MANTENER PRODUCTOS Y DISPONIBILIDAD	17.2	MANEJAR ERRORES EN DETALLE DE PRODUCTO	A2, A3		DEFINIDO	10	5	1
7 REVISAR Y COMPRAR	7.7	SALIR DE COMPRAR	A14		DEFINIDO	11	20	1
5 DISEÑAR INTERIORES	5.1	NAVEGAR EL DISEÑO	BF, A1		DEFINIDO	12	8	1
5 DISEÑAR INTERIORES	5.2	MANEJAR ERRORES DE NAVEGACIÓN	A2		DEFINIDO	13	2	1
1 CONSTRUIR CASA	1.2	ADICIONAR O REMOVER PISO	A2, A5	1.2.1, 1.2.2, 1.2.3	DEFINIDO	14	1	1

**Figura 13:** La Hoja de Cálculo de una Porción de Caso de Uso de un Rastreador Simple de Casos de Uso.

## Completar los Productos de Trabajo

A medida que rastree los casos de uso y las porciones de los casos de uso usted necesitará, al menos, bosquejar y compartir los productos de trabajo de soporte.

Todos los productos de trabajo se definen con varios niveles de detalle. El primer nivel de detalle define lo estrictamente necesario, la mínima cantidad de información que se requiere para que la práctica trabaje. Los niveles posteriores de detalle ayudan al equipo a lidiar con cualquier circunstancia que pudieran encontrar. Por ejemplo, esto le permite a los equipos pequeños y colaborativos tener narrativas de caso de uso muy livianas definidas en simples tarjetas indexadas y a grandes equipos distribuidos tener narrativas de caso de uso más detalladas presentadas como documentos. Así, los equipos pueden ampliar posteriormente las narrativas a medida que se necesite, para ayudar con la comunicación o definir completamente los requisitos importantes o críticos. Se deja al equipo que decida si necesita o no ir más allá de lo necesario, agregando detalles de una manera natural a medida que encuentren problemas que lo necesario no aborde.

La **Figura 14** muestra los niveles de detalle definidos para el conjunto de productos de trabajo de los Casos de Uso 2.0. El nivel más liviano de detalle se muestra en la parte superior de la tabla. La cantidad de detalle en el producto de trabajo se aumenta a medida que se baja en las columnas, ampliando y expandiendo el contenido.

	MODELO DE CASOS DE USO	NARRATIVA DE CASOS DE USO	REALIZACIÓN DE CASOS DE USO	CASOS DE PRUEBA	INFORMACIÓN DE SOPORTE
BOSQUEJO:		BREVEMENTE DESCRITA		CON IDEAS DE PRUEBA FORMULADAS	INICIADA
LO MÉRAMENTE ESENCIAL:	CON VALOR ESTABLECIDO	CON RESUMEN BÁSICO	CON ELEMENTOS DE IMPLEMENTACIÓN IDENTIFICADOS	CON ESCENARIO SELECCIONADO	SIMPLEMENTE DEFINIDA
AMPLIADO:	CON LÍMITES DEL SISTEMA ESTABLECIDOS	CON RESUMEN ESENCIAL	CON RESPONSABILIDADES ASIGNADAS	CON VARIABLES IDENTIFICADAS	MODELADA E ILUSTRADA
EXPANDIDO:	ESTRUCTURADO	COMPLETAMENTE DESCRITA	CON INTERACCIÓN DEFINIDA	CON VARIABLES FIJAS	DEFINIDA DETALLADAMENTE
EXPANDIDO POSTERIORMENTE:				PROGRAMADAS/AUTOMATIZADAS	

**Figura 14:** Niveles de Detalle de los Productos de Trabajo.

La buena noticia es que usted siempre empieza de la misma manera, con lo estrictamente necesario. Luego, el equipo puede adaptar continuamente el nivel de detalle en las narrativas de sus casos de uso para cumplir con las necesidades emergentes. El nivel de detalle también se puede ajustar para reducir desperdicio, cualquier cosa más allá de lo estrictamente necesario debería tener una razón clara para existir o se eliminará. Como Einstein decía (o se le atribuye) "Todo debería ser tan simple como sea posible, pero no más simple".

Para más información sobre los productos de trabajo y sus niveles de detalle vea el Apéndice 1: Productos de Trabajo.

## Cosas Por Hacer

Los Casos de Uso 2.0 dividen el trabajo en un número de actividades esenciales que se deben hacer si los casos de uso se hacen para proporcionar un valor real al equipo. Estas actividades se muestran en la **Figura 15**, donde se agrupan en actividades que se usan para descubrir, ordenar y verificar los requisitos, y actividades que se usan para modelar, implementar y probar el sistema. Los polígonos sólidos indican actividades que la práctica explícitamente define. Los polígonos punteados indican otras actividades de las cuales la práctica depende para ser exitosa. Los Casos de Uso 2.0 no se preocupan por cómo se ejecutan estas actividades, solo necesitan que se hagan.



**Figura 15:** Las Actividades en los Casos de Uso 2.0.

Lea la **Figura 15** de izquierda a derecha para hacerse una idea del orden en el cual se ejecutan las actividades. Estas siempre se ejecutarán muchas veces en el curso de su trabajo. Aun si una simple actividad tal como 'Encontrar Actores y Casos de Uso' se debe ejecutar muchas veces para encontrar todos los casos de uso y se pueda realizar en paralelo con, o después de, las otras actividades. Por ejemplo, mientras continua con 'Encontrar Actores y Casos de Uso' usted también puede implementar algunas de las porciones de casos de uso que encontró previamente.

El resto de este capítulo proporciona una introducción a cada una de las actividades, siguiendo el viaje de una porción de caso de uso, desde la identificación inicial de su caso de uso padre hasta su prueba e inspección final. El siguiente capítulo incluye una breve discusión de cómo organizar las actividades para soportar distintas estrategias de desarrollo tales como *Scrum*, *Kanban*, Iterativo y Cascada.

---

## Encontrar Actores y Casos de Uso

Primero, usted necesita encontrar algunos actores y casos de uso que le ayuden a:

- Acordar las metas del sistema
- Acordar el comportamiento del nuevo sistema
- Definir el alcance de las versiones del sistema
- Acordar el valor que el sistema proporciona
- Identificar las formas para emplear y probar el sistema

La mejor forma de hacer esto es realizar un taller de modelado de casos de uso con los interesados. No hay necesidad de encontrar todos los casos de uso del sistema, solamente enfocarse en aquellos que van a proporcionar el valor que los interesados buscan. A medida que usted inspeccione y adapte los casos de uso se encontrarán otros actores y casos de uso.

A medida que se descubren los casos de uso, se deberían ordenar para apoyar el plan de versiones del equipo. Uno de los aspectos más importantes de los casos de uso es que habilitan una gestión de alto nivel sin la necesidad de descubrir o escribir todas las historias. Si un caso de uso no se necesita, entonces no es necesario discutir o documentar sus historias. Si el caso de uso está en el alcance, se debería describir de tal forma que haya suficiente información para iniciar el proceso de dividirlo en porciones.

Repita esta actividad tantas veces como sea necesario para evolucionar su modelo y encontrar cualquier actor o caso de uso faltante.

### TIP: TORMENTA DE MODELOS PARA INICIAR EL MODELO DE CASOS DE USO

La forma natural del modelo de casos de uso y su uso en el Lenguaje de Modelado Unificado, puede ser una barrera para involucrar a los interesados en el esfuerzo de modelado.

Una muy buena manera de salir adelante con esto es, simplemente, reunir a los interesados para hacer una tormenta de ideas con diferentes usuarios y sus metas usando notas adhesivas (vertical para usuarios y horizontal para metas). Luego, se debe facilitar su agrupamiento en actores y casos de uso, los cuales se cuantificarán, describirán y ordenarán fácilmente para los interesados.



## Dividir los Casos de Uso

A continuación, usted necesita crear las primeras porciones de casos de uso. Esto se hace para:

- Crear elementos de tamaño adecuado con los cuales el equipo trabaje
- Estar acorde con el tiempo y el presupuesto disponible
- Entregar el más alto valor a los usuarios y a los demás interesados
- Demostrar el progreso crítico del proyecto o el entendimiento de las necesidades

Aún el caso de uso más simple cubre múltiples historias. Es necesario dividir los casos de uso para seleccionar las historias a implementar. La división se debería hacer con los interesados para asegurarnos de que todas las porciones creadas sean de valor y valga la pena implementarlas. No divida todos los casos de uso de una sola vez. Tan solo identifique suficientes porciones para cubrir las necesidades inmediatas del equipo.

No es necesario dividir completamente los casos de uso, solo defina las porciones que son necesarias para avanzar en el trabajo y deje el resto de las historias en los casos de uso para dividirlos siempre y cuando se necesiten. Incluso, es posible adoptar un modelo donde los desarrolladores soliciten nuevas porciones cuando ellos tengan la capacidad de implementarlas.

Las porciones creadas se deberían ordenar para asegurarnos de que el equipo de desarrollo las enfrente en el orden correcto. De nuevo, esto se debería hacer con los interesados y otros miembros del equipo, para asegurarnos de que el ordenamiento defina el sistema usable más pequeño posible. La mejor forma de hacerlo es considerando la combinación de prioridad, valor, riesgo y necesidad.

Repita esta actividad siempre que se requieran nuevas porciones.

### TIP: SOLO SE NECESITA EL FLUJO BÁSICO DEL CASO DE USO MÁS IMPORTANTE PARA CREAR LA PRIMERA PORCIÓN

Mucha gente piensa que es necesario bosquejar todos los casos de uso antes de empezar a crear sus porciones. Esto conduce a la adopción de un método en cascada que retrasa la creación de software de valor y funcional.

Una porción de un caso de uso es suficiente para que el equipo empiece el desarrollo y prueba del sistema.

La primera porción siempre debería ser del flujo básico. Para algunos sistemas complejos esta porción quizás no cubra todo el flujo. Es posible tomar un sub-conjunto del flujo básico, saltando el detalle de algunos pasos y dejar la solución a otros, para atacar los desafíos más grandes en la implementación del caso de uso y aprender si se puede implementar o no.

## Preparar una porción del Caso de Uso

Luego de seleccionar una porción para el desarrollo, se requiere más trabajo para:

- Preparar la porción para la implementación
- Definir claramente lo que significa implementar con éxito la porción
- Definir las características requeridas (por ejemplo, los requisitos no funcionales)
- Enfocar el desarrollo de software en las pruebas que se deben cumplir

Preparar una porción del caso de uso es una actividad de colaboración. Es necesario el aporte de los interesados, los desarrolladores y verificadores para aclarar la narrativa del caso de uso y definir los casos de prueba.

Al preparar una porción de un caso de uso, usted debería centrarse en las necesidades de los desarrolladores y verificadores, quienes implementarán y verificarán la porción. Piense acerca de cómo van a tener acceso a la información que necesitan. ¿Serán capaces de hablar con expertos en la materia para dar cuerpo a las historias o necesitarán que todo se documente para ellos?

También, es necesario balancear el trabajo entre el nivel de detalle de la narrativa de los casos de uso y el detalle de la narrativa de los casos de prueba.

Cuanto más detalles se pongan en el relato de los casos de uso, más fácil será crear los casos de prueba.

Por otra parte, entre más ligera la narrativa del caso de uso, menos duplicación y repetición habrá entre el caso de uso y los casos de prueba. Se debe crear la narrativa de los casos de uso y de los casos de prueba al mismo tiempo, de modo que los autores puedan balancear sus propias necesidades y las de sus interesados. Todavía puede haber trabajo por hacer si los casos de prueba se van a automatizar, pero no habrá duda acerca de lo que hay que lograr mediante una implementación exitosa.

Ejecute esta actividad al menos una vez para cada porción. Repita esta actividad siempre que se apliquen cambios a la porción.

### **TIP: SI LA PORCIÓN NO TIENE CASOS DE PRUEBA, NO TIENE LA PREPARACIÓN ADECUADA**

Quando prepare una porción de caso de uso, no olvide definir los casos de prueba que se usarán para verificarla. Al mirar los casos de prueba conocemos la meta que necesitamos conseguir.

Los casos de prueba proporcionan a los desarrolladores un conocimiento empírico de lo que el sistema necesita hacer. Ellos sabrán que el desarrollo de la porción no se completará hasta que pase todos los casos de prueba exitosamente.

---

## Analizar una porción de Caso de Uso

Antes de empezar a codificar debe analizar la porción para:

- Entender su impacto en los elementos del sistema que se emplearán para implementarla
- Definir las responsabilidades de los elementos que se afectan del sistema
- Definir cómo interactúan los elementos del sistema para ejecutar el caso de uso

Cuando a un equipo se le presenta una nueva porción a implementar, lo primero que debe hacer es averiguar cómo va a afectar el sistema. ¿Cuántos *bits* del sistema se va a necesitar cambiar y de qué manera? ¿Cuántas cosas nuevas se necesitan y dónde encajan?

Analizar las porciones objetivo es, a menudo, un precedente de la planificación de las tareas de desarrollo. Esto permite al equipo planear la implementación de la porción como un conjunto de cambios menores en el código en lugar de como una pieza grande e indivisible de trabajo.

Alternativamente, la misma porción se puede utilizar como unidad de trabajo y analizar la porción es solo la última cosa que realiza el desarrollador antes de iniciar la codificación.

A medida que el equipo entiende el sistema y su arquitectura, sus integrantes hallarán más y más fácil analizar las porciones y, a menudo, serán capaces de hacerlo en sus cabezas. Vale la pena hacer un esbozo del análisis con algunos colegas antes de iniciar la codificación.

Esto validará muchas de las decisiones de diseño, comprobará que nada se malentienda y proporcionará trazabilidad para su uso posterior cuando se investigue el impacto de los defectos y cambios. El resultado de este tipo de análisis se conoce como una realización del caso de uso, ya que muestra cómo los elementos del sistema en implementación ejecutan el caso de uso.

Realice esta actividad al menos una vez para cada porción. Repita esta actividad siempre que se apliquen cambios a la porción.

### TIP: MANTENGA EL ANÁLISIS COLABORATIVO Y LIVIANO

La forma más fácil de analizar la porción de un caso de uso es hacer que el equipo se reúna para discutir como esta afectará a los distintos elementos del sistema.

A medida que el equipo elabora el diseño, lo dibuja en un tablero, típicamente en forma de un diagrama de secuencia o comunicación, el cual se puede fotografiar o modelar en una herramienta que el equipo seleccione.

---

## Implementar el Software (para una porción)

Ahora usted está listo para diseñar, programar, hacer las pruebas unitarias e integrar los componentes de software necesarios para implementar una porción del caso de uso.

El software se desarrollará porción por porción, con diferentes miembros del equipo trabajando en paralelo en diferentes porciones. Cada porción requerirá cambios a una o más piezas del sistema. Para completar la implementación de una porción, las piezas resultantes de software se tendrán que someter a pruebas unitarias y luego integradas con el resto del sistema.

## Probar el sistema (para una porción)

A continuación, pruebe independientemente el software para verificar que la porción del caso de uso se implementó exitosamente. Cada porción del caso de uso se tiene que probar antes de que se pueda considerar completa y verificada. Esto se hace ejecutando con éxito los casos de prueba de la porción. La independencia de las porciones del caso de uso permite probarlas tan pronto como se implementen y proporciona retroalimentación inmediata a los desarrolladores.

Los Casos de Uso 2.0 trabajan con las prácticas de prueba más populares. Se puede considerar una forma de desarrollo dirigido por pruebas (TDD, por su nombre en inglés), ya que crea por adelantado los casos de prueba para cada porción, antes de que la porción se entregue a los desarrolladores para su implementación.

## Probar el sistema como un Todo

Cada incremento del sistema de software se tiene que probar para verificar que se implementaron correctamente todas las porciones nuevas del caso de uso, sin dañar ninguna otra parte del sistema. No es suficiente probar cada porción cuando se complete. El equipo, además, debe probar el sistema como un todo para asegurarse de que todas las porciones implementadas son compatibles y que los cambios en el sistema no se tradujeron en fallas de las porciones previamente verificadas.

Los casos de prueba producidos utilizando los Casos de Uso 2.0 son robustos y flexibles. Esto se debe a que la estructura de la narrativa de los casos de uso se convierte en casos de prueba basados en escenarios, ejecutables de manera independiente.

---

## Inspeccionar y adaptar los Casos de Uso

También, es necesario afinar y evaluar continuamente los casos de uso y las porciones de los casos de uso para:

- Controlar los cambios
- Hacer seguimiento del progreso
- Ajustar el trabajo dentro del tiempo y presupuesto disponible
- Mantener el modelo de casos de uso al día
- Ajustar el tamaño de las porciones para aumentar el rendimiento

A medida que el proyecto avanza, es esencial que usted trabaje continuamente con el modelo de casos de uso, los casos de uso y las porciones de casos de uso. Las prioridades cambian, se aprenden lecciones y se requieren cambios. Todo esto puede tener un impacto en los casos de uso y en las porciones de casos de uso que ya se implementaron, así como aquellas que todavía se espera implementar. A menudo, esta actividad conducirá al descubrimiento de nuevos casos de uso y a la refactorización de casos de uso y porciones de casos de uso existentes.

La demanda variable del proyecto puede necesitar que usted ajuste el empleo de los Casos de Uso 2.0 y que ajuste el tamaño de las porciones o el nivel de detalle en la narrativa de los casos de uso, apoyando las definiciones y los casos de prueba. Es importante que, continuamente, revise y adapte su forma de trabajo, así como los casos de uso y las porciones de casos de uso con las que está trabajando.

Lleve a cabo esta actividad según sea necesario para mantener sus casos de uso y gestionar los cambios.

### **TIP:** NO OLVIDE MANTENER SU LISTA DE PENDIENTES DE PORCIONES DE CASOS DE USO

Ordenando las porciones y rastreando su estado (alcance definido, preparada, analizada, implementada, verificada), usted crea una lista de pendientes de los requisitos que faltan por implementar. Esta lista se debería monitorear y ajustar continuamente para reflejar el progreso del equipo y los deseos de los interesados.

A medida que el proyecto avanza, usted debería monitorear y ajustar el tamaño de la porción para eliminar la basura y mejorar la efectividad del equipo.

---

# Emplear los Casos de Uso 2.0

Los Casos de Uso 2.0 se pueden utilizar en muchos contextos diferentes para ayudar a producir muchos tipos diversos de sistemas. En este Capítulo analizaremos como utilizar los casos de uso para diferentes tipos de sistemas, diferentes tipos de requisitos y diferentes enfoques de desarrollo.

## Casos de Uso 2.0: Aplicables para todos los tipos de sistema

Mucha gente piensa que los casos de uso se pueden aplicar solamente a los sistemas de uso intensivo en los que hay una gran cantidad de interacción entre usuarios humanos y el sistema. Esto es extraño porque la idea original de los casos de uso provino de los sistemas de conmutación de telecomunicaciones, que tienen tanto usuarios humanos (suscriptores, operadores) y usuarios máquinas, en forma de sistemas interconectados. Los casos de uso son, por supuesto, aplicables a todos los sistemas que se utilizan y eso significa, por supuesto, todos los sistemas.

### Casos de Uso 2.0: No son solo para aplicaciones de uso intensivo

De hecho, los casos de uso son tan útiles para los sistemas con poca o ninguna interacción humana como para los de uso intensivo. Hoy en día, la gente está utilizando casos de uso en el desarrollo de todo tipo de software embebido, en dominios tan diversos como la industria automotriz, la electrónica de consumo, la aeroespacial, la militar y la médica. Incluso, los sistemas en tiempo real de control de procesos usados para plantas químicas se pueden describir con casos de uso, donde cada caso de uso se centra en una parte específica del comportamiento de los procesos de la planta y en las necesidades de automatización.

Todo lo que se necesita para que los casos de uso sean apropiados es que el sistema colabore con el mundo exterior, independientemente de si los usuarios son humanos u otros sistemas. Su aplicabilidad es mucho más amplia de lo que la mayoría de las personas piensa.

### Casos de Uso 2.0: No son solo para desarrollo de software

La aplicación de los casos de uso no se limita al desarrollo de software. También pueden ayudar a entender los requisitos del negocio, analizar el negocio existente, diseñar nuevos y mejores procesos de negocio y explotar el poder de las tecnologías de la información y la comunicación (TIC) para transformar su negocio. Los casos de uso se emplean de forma recursiva para 1) modelar el negocio y sus interacciones con el mundo exterior y 2) modelar sistemas necesarios para apoyar y mejorar el negocio. Así, usted perfectamente puede identificar dónde impactarán los sistemas en el negocio y qué sistemas se requieren para apoyar el negocio.

Los casos de uso que se emplean para modelar el negocio se referencian, a menudo, como casos de uso del negocio. Ellos proporcionan el contexto para el trabajo de desarrollo de sus sistemas TIC, permitiendo que el desarrollo del negocio y el desarrollo de la TIC se lleven a cabo en perfecta sincronización. No solo se pueden desarrollar sistemas TIC porción por porción, sino que también se puede desarrollar el modelo de negocio porción por porción. Esto es muy poderoso, ya que le permite evolucionar su negocio y sus sistemas de apoyo en simultánea uno con el otro, lo que permite desarrollo incremental del negocio y el desarrollo incremental de los sistemas.

En el mundo moderno, los negocios y los sistemas TIC que los soportan se pueden y deberían desarrollar en sincronía (uno no funcionaría sin el otro). El empleo de casos de uso y de porciones de los casos de uso en los límites del negocio y de la TIC puede cerrar la brecha entre el negocio y la TIC, permitiéndoles trabajar como verdaderos socios colaborativos.

---

## Casos de Uso 2.0: Manejando todo tipo de requisitos

A pesar de ser una de las técnicas más populares para describir la funcionalidad de los sistemas, los casos de uso también se utilizan para explorar sus características no funcionales. La forma más sencilla de hacer esto es capturarlas como parte de los casos de uso mismos. Por ejemplo, se pueden relacionar los requisitos de rendimiento con el tiempo transcurrido entre pasos específicos de un caso de uso o listar los niveles de servicio esperados para un caso de uso, como parte del caso de uso en sí.

Algunas características no funcionales son más sutiles que esto y aplican a muchos, sino a todos, los casos de uso. Esto es particularmente cierto en la construcción de arquitecturas de capas, incluyendo componentes de infraestructura, tales como seguridad, gestión de transacciones, servicios de mensajería y gestión de datos. Los requisitos en estas áreas se pueden expresar como casos de uso (por lo general, casos de uso separados centrados en la utilización técnica del sistema). Llamamos a estos casos de uso adicionales casos de uso de infraestructura, ya que los requisitos que contienen conducirán la creación de la infraestructura sobre la cual se ejecutará la aplicación. Estos casos de uso y sus porciones se pueden considerar como intereses transversales que afectarán el comportamiento del sistema cuando los casos de uso funcionales más tradicionales se realicen. Por ejemplo, un caso de uso se podría crear para explorar cómo el sistema se encargará de las operaciones de base de datos, incluyendo todos los escenarios de uso diferentes, tales como los esquemas de bloqueo de datos, almacenamiento en memoria temporal, *commit* y *roll-back*. Este caso de uso se aplicaría cada vez que otro caso de uso recupera o almacena datos en el sistema.

La combinación de estos casos de uso de infraestructura con otras técnicas como la separación de intereses y la programación orientada a aspectos, permite que estos requisitos comunes se atiendan sin tener que cambiar la implementación de los casos de uso funcionales ya existentes.

## Casos de Uso 2.0: Aplicable para todos los enfoques de desarrollo

Los Casos de Uso 2.0 trabajan con todos los enfoques de desarrollo de software populares como:

- Enfoques iterativos dirigidos por listas de pendientes, como *Scrum*, *EssUP* y *OpenUP*
- Enfoques basados en flujo de una pieza, como *Kanban*
- Enfoques todo en uno, como la Cascada tradicional

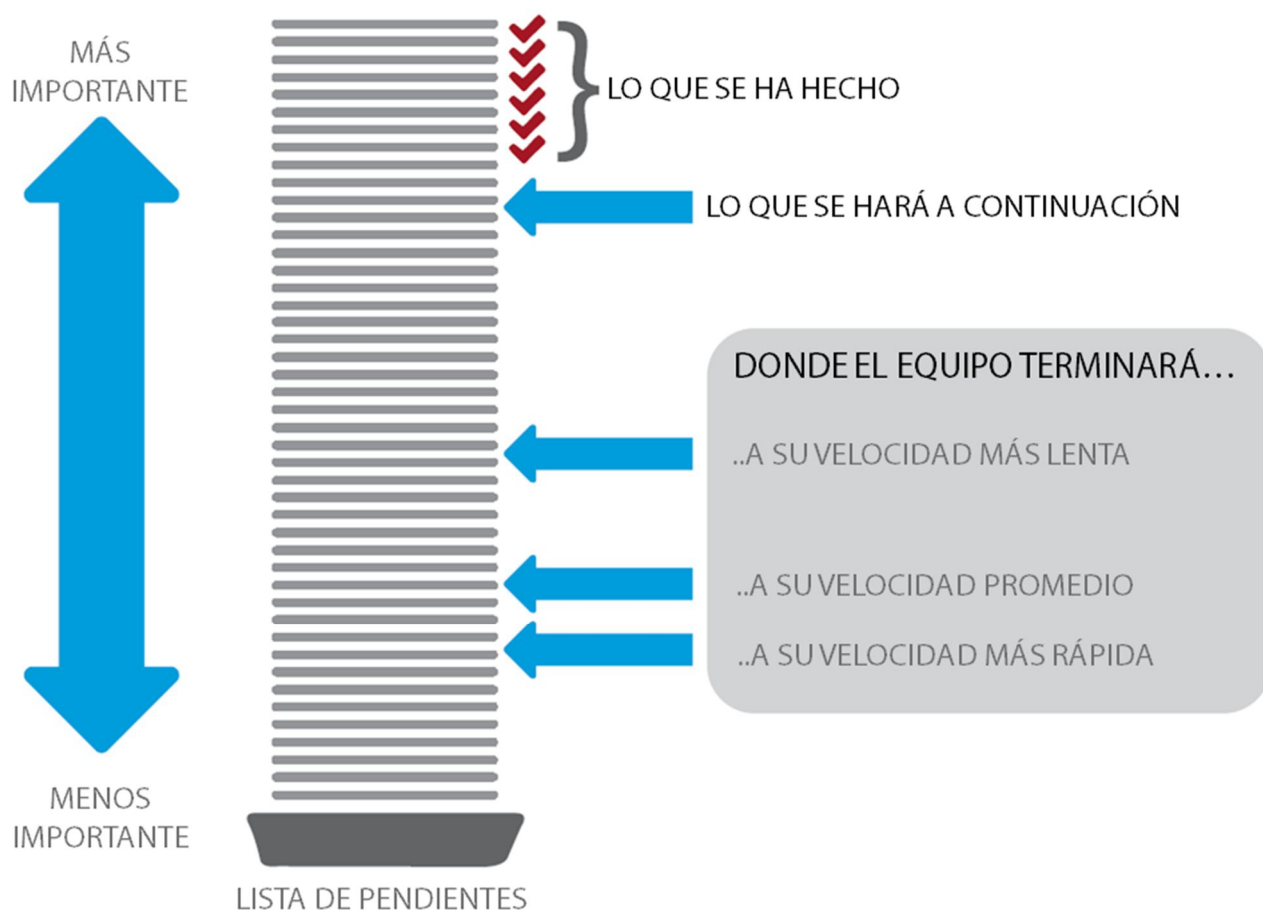
En las siguientes tres secciones cortas vamos a ilustrar como los Casos de Uso 2.0 y, en particular, las porciones de casos de uso pueden ayudar con cada uno de estos enfoques. Estas secciones no son tan independientes como el resto del documento y dependen del conocimiento básico que el lector tenga del enfoque que se discute. Le recomendamos que lea solo las secciones de los enfoques con los que tenga familiaridad.



## Casos de Uso 2.0 e iteraciones que se conducen con listas de pendientes

Antes de adoptar cualquier enfoque que se conduce con listas de pendientes, debemos entender cuáles elementos van en la lista de pendientes. Hay varias formas de listas de pendientes que los equipos usan para conducir su trabajo, incluyendo productos, liberación de versiones y proyectos.

Independientemente de la terminología usada, todas ellas siguen los mismos principios. La lista de pendientes misma es una lista ordenada de todo lo que se podría necesitar y es la única fuente de los requisitos para cualquier cambio que se haga. El concepto básico de una lista de pendientes se ilustra en la **Figura 16**.

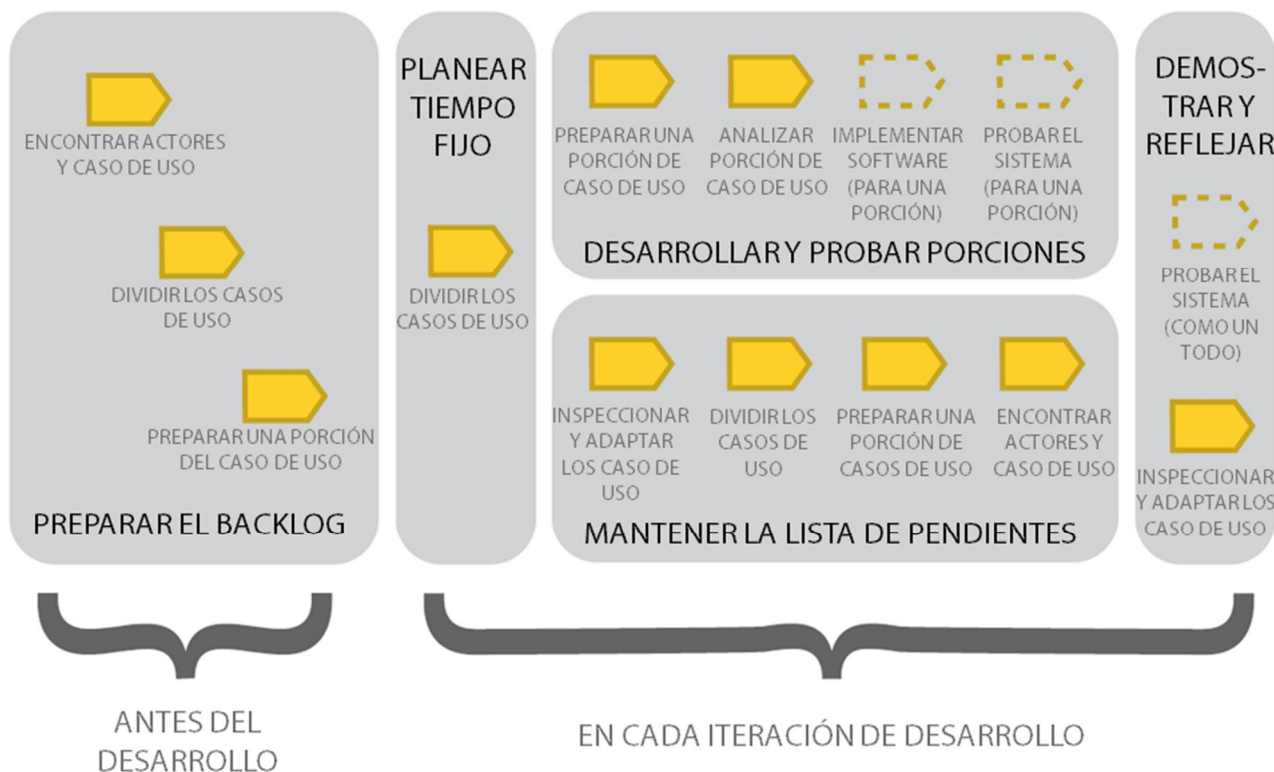


**Figura 16: Una Lista de Pendientes Básica.**

Cuando empleamos los Casos de Uso 2.0, las porciones de casos de uso son los elementos primarios de la lista de pendientes. El uso de porciones de casos de uso asegura que los elementos de la lista de pendientes se conformen bien, puesto que ellos son naturalmente independientes, de valor y se pueden probar. La estructuración de la narrativa de los casos de uso que las define, asegura que las porciones son estimables y negociables y el mecanismo de división de los casos de uso permite definir porciones tan pequeñas como se necesiten para apoyar al equipo de desarrollo.

Los casos de uso no se colocan en la lista ordenada, ya que no es claro lo que esto significaría. ¿Significa que es donde la primera porción del caso de uso aparecería o donde la última porción del caso de uso aparecería? Si queremos ubicar un caso de uso en la lista antes de definir sus porciones, simplemente creamos una porción ficticia para representar todo el caso de uso y la insertamos en la lista.

Cuando adoptamos un enfoque que se conduce con listas de pendientes, es importante tener en cuenta que la lista de pendientes no se construye y completa de antemano, sino que se revisa y se refina continuamente, algo que muchas veces se conoce como la preparación o el mantenimiento de la lista de pendientes. La secuencia típica de actividades para un enfoque iterativo que se conduce con listas de pendientes se muestra en la **Figura 17**.



**Figura 17: Actividades de los Casos de Uso 2.0 para Enfoques de Desarrollo Iterativos.**

Antes de iniciar el desarrollo se prepara la lista de pendientes; 'Encontrar Actores y Casos de Uso' se usa para construir el modelo inicial de casos de uso y el alcance del sistema, 'Dividir los Casos de Uso' se usa para crear el conjunto inicial de porciones más importantes de casos de uso para alimentar la lista de pendientes, y 'Preparar una Porción del Caso de Uso' se usa para hacer que una o más de estas estén listas para el desarrollo en la primera iteración.

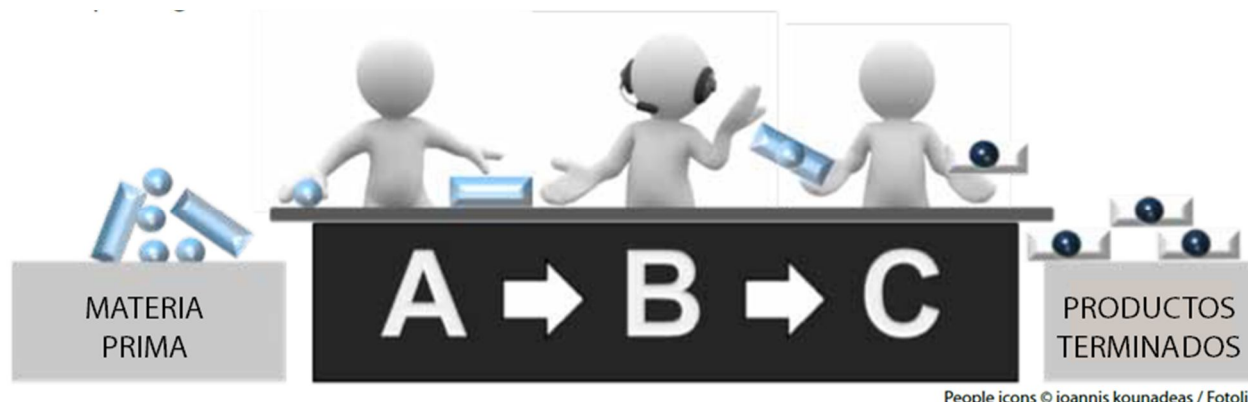
Una vez que la lista de pendientes está lista y corriendo, usted puede iniciar la primera iteración del desarrollo. Cada iteración comienza con algo de planeación. Durante esta planeación usted puede necesitar usar 'Dividir los Casos de Uso' para dividir aún más las porciones de casos de uso y asegurar que estas sean lo suficientemente pequeñas, de tal forma que se puedan completar en la iteración. El equipo de desarrollo emplea, entonces, 'Preparar una Porción de Caso de Uso', 'Analizar una Porción de Caso de Uso', 'Implementar Software (para una porción)' y 'Probar el Sistema (para una porción)' para desarrollar las porciones identificadas y adicionarlas al sistema.

Mientras el desarrollo avanza, el equipo también usa 'Inspeccionar y Adaptar los Casos de Uso', 'Dividir los Casos de Uso' y 'Preparar una Porción de Casos de Uso' para mantener la lista de pendientes, manejar los cambios y asegurar que haya suficientes elementos listos en la lista de pendientes para conducir la siguiente iteración. El equipo puede aún necesitar 'Encontrar Actores y Casos de Uso' para manejar cambios mayores o descubrir más casos de uso para el equipo. En *Scrum*, se recomienda que los equipos gasten del 5% al 10% de su tiempo mantenimiento su lista de pendientes. Este es no es un gasto considerable para el equipo y los Casos de Uso 2.0 proporcionan los productos de trabajo y las actividades necesarias para hacer esto fácil y eficientemente.

Por último, al final de la iteración el equipo necesita hacer una demostración del sistema y reflejar su desempeño durante la iteración. El equipo debería usar 'Probar el Sistema (como un todo)' para entender dónde está, e 'Inspeccionar y Adaptar Casos de Uso' para mostrar la calidad y efectividad de sus casos de uso y porciones de los casos de uso.

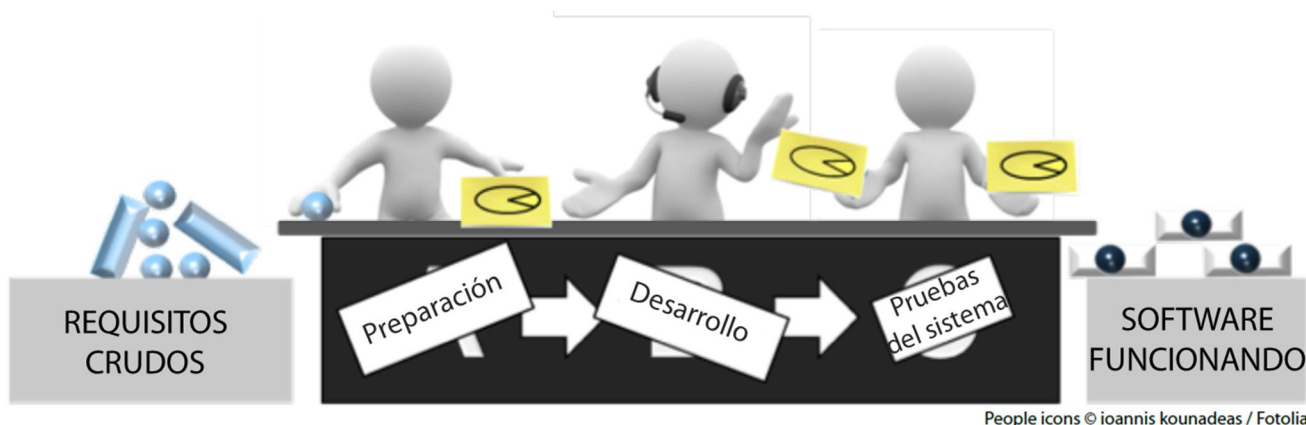
## Casos de Uso 2.0 y el flujo de una sola pieza

El flujo de una sola pieza es un enfoque que evita el represamiento de los requisitos que se ve en las estrategias iterativas y cascada. En el enfoque del flujo de una sola pieza, cada elemento de los requisitos fluye a lo largo del proceso de desarrollo. El flujo de una sola pieza es una técnica tomada del proceso de manufactura *Lean*. La **Figura 18** muestra un equipo pequeño ocupado en el flujo de una sola pieza, pasando cada elemento directamente de la estación A a la B y a la C.



**Figura 18: Flujo Básico de Una-Pieza.**

Para que esto trabaje efectivamente, se necesitan elementos pequeños más o menos iguales que fluyan rápidamente a lo largo del sistema. Para el desarrollo de software, los requisitos son la materia prima y el software funcionando es el producto terminado. Los casos de uso podrían ser de tamaños irregulares y demasiado grandes para fluir a lo largo del sistema. El tiempo en las estaciones A, B y C podría ser impredecible y las cosas se podrían atrancar. En cambio, las porciones de los casos de uso se pueden dimensionar apropiadamente y afinar para cubrir las necesidades del equipo. La **Figura 19** ilustra el flujo de una sola pieza para el desarrollo de software con porciones de casos de uso.



**Figura 19: Flujo de Una-Sola Pieza para Desarrollo de Software con Porciones de Casos de Uso.**

Así como los elementos deben fluir rápidamente a lo largo del sistema, también debe haber suficientes de ellos para mantener ocupado al equipo. El flujo de una sola pieza no significa que hay solo un elemento de requisitos con el que se está trabajando al tiempo o que hay solo una pieza de trabajo entre una estación de trabajo y la siguiente. Se usan límites de trabajo en progreso para nivelar el flujo y prevenir que se construya cualquier lista de pendientes dispendiosa.

El flujo de una sola pieza no significa que los individuos hacen una sola cosa ni que trabajan en una sola estación. Por ejemplo, puede haber más gente trabajando en Desarrollo de la que hay en Pruebas y, si las cosas se empiezan a atascar, entonces la gente se debería mover para ver cómo conseguir que las cosas se muevan nuevamente. Si no hay porciones de casos de uso esperando pruebas pero hay porciones atrancadas en análisis, entonces los verificadores pueden mostrar su iniciativa y ayudar a hacer el trabajo

de preparación. De la misma forma, no se limite a tener una persona en una estación de trabajo o solo una instancia de cada estación de trabajo.

Los tableros *Kanban* constituyen una técnica para visualizar el flujo a lo largo de una línea de producción. Un *Kanban* es un signo, bandera o señal dentro del proceso de producción para disparar la producción y suministro de productos como parte de la manufactura justo a tiempo y *lean*. En un tablero *Kanban* se usan tarjetas *Kanban* para representar los elementos en el sistema. La **Figura 20** muestra un tablero *Kanban* simple para un equipo de desarrollo, el cual primero analiza cada porción para entender su impacto, luego desarrolla y hace pruebas unitarias del software y, finalmente, prueba de manera independiente el software resultante antes de ponerlo en vivo.

Los límites del trabajo en progreso se muestran en rojo. Leyendo de izquierda a derecha, se puede ver que se identificaron las porciones y se delimitó su alcance antes de enviarlas como entrada al equipo. Aquí hay un límite del trabajo en progreso de 5 y los clientes, dueños del producto, o el equipo de requisitos del negocio que son la fuente de los requisitos, intentan mantener 5 porciones de casos de uso listas para implementación en cualquier instante de tiempo.



**Figura 20:** Porciones de Casos de Uso en un Tablero *Kanban*.

Las porciones se empujan desde la cola de entrada al área de preparación donde se lleva a cabo el análisis de impacto, se clarifican las historias y se finalizan los casos de prueba. Aquí hay un límite del trabajo en progreso de tres elementos. Sobre los elementos en la columna en proceso se trabaja actualmente. Los elementos en la columna “hecho” finalizan su preparación y esperan a que los recoja un desarrollador. De esta forma, las porciones avanzan a lo largo de todo el equipo de desarrollo y, después de pasar exitosamente las pruebas independientes del sistema, se despliegan en vivo. Un límite del trabajo en progreso cubre todo el trabajo en la estación, incluyendo tanto los elementos en proceso como hechos. No existe límite del trabajo en progreso en la salida ni número de elementos que se puedan desplegar en vivo.

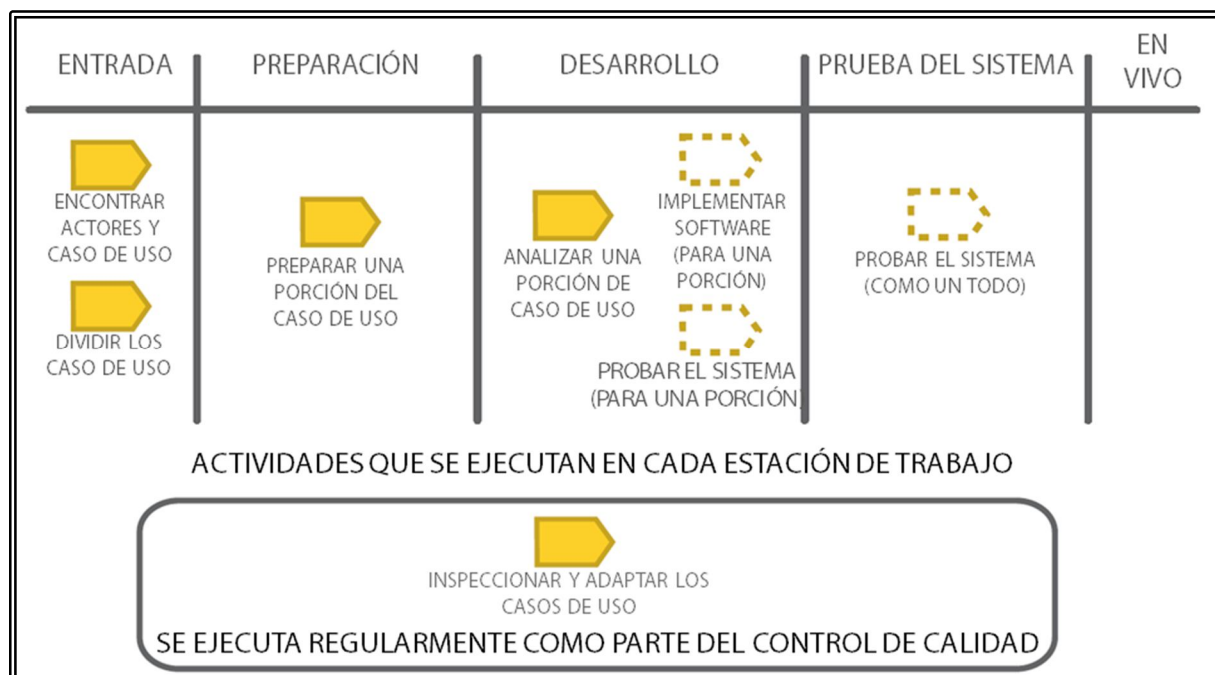
Una cosa importante de notar acerca de *Kanban* es que no hay un tablero *Kanban* definitivo ni un conjunto de límites del trabajo en progreso; la estructura del tablero es dependiente de la estructura del equipo y las prácticas trabajadas. Usted debería afinar el tablero y los límites del trabajo en progreso a medida que afina sus prácticas. Los estados de las porciones de casos de uso son de gran ayuda para esta

clase de diseño de trabajo. La **Figura 21** muestra la alineación entre los estados y el tablero *Kanban* que se muestran en la **Figura 20**. Los estados son muy poderosos porque definen claramente en qué estado debería estar la porción cuando se vaya a pasar a la siguiente sección de la cadena.

La **Figura 22** muestra dónde se aplican las distintas actividades de los Casos de Uso 2.0. El punto interesante aquí es que “Inspeccionar y Adaptar los Casos de Uso” no es responsabilidad de alguna estación de trabajo en particular sino que se conduce como parte del normal control de calidad que hace el equipo. Esta actividad ayuda al equipo a afinar el número y tipo de estaciones de trabajo que deben tener, así como sus límites del trabajo en progreso.



**Figura 21:** Alineación de los Estados de una Porción de Caso de Uso en un Tablero *Kanban*.



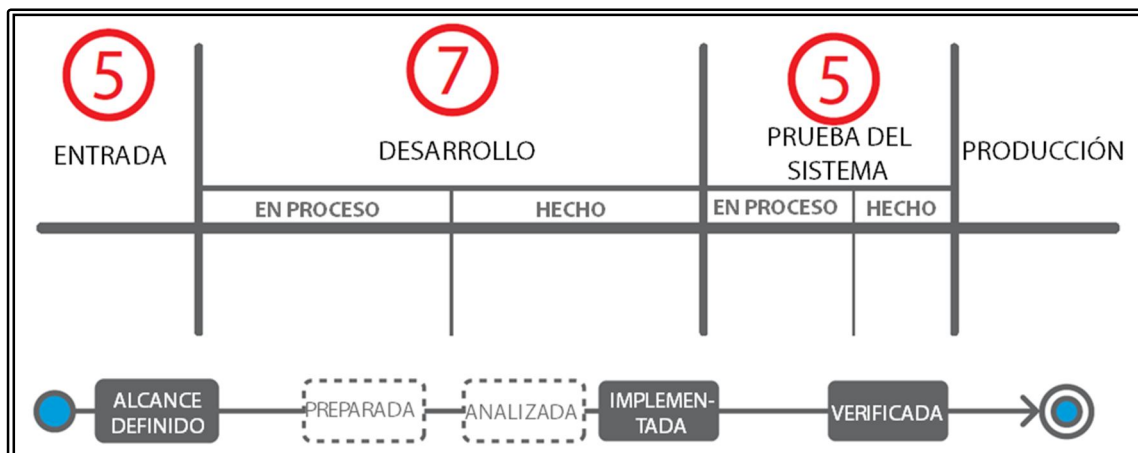
**Figura 22:** Actividades de los Casos de Uso 2.0 para Enfoques de Cascada.

Por ejemplo, como resultado de revisar la efectividad del equipo, usted podría decidir eliminar la estación de trabajo de preparación y aumentar los límites del trabajo en progreso para el desarrollo y las pruebas del sistema. Nuevamente, usted explota los estados de la porción del caso de uso para definir lo que eso significa para que cada estación de trabajo finalice su trabajo, dando como resultado el tablero *Kanban* que se muestra en la **Figura 23**.



## Casos de Uso 2.0 y la cascada

Por varias razones, usted puede encontrar que necesita desarrollar su aplicación de software dentro de las restricciones de alguna forma de modelo de gobernabilidad en cascada. Típicamente, esto significa que se hará algún intento por capturar todos los requisitos antes de que los envíen a un tercero para su desarrollo.



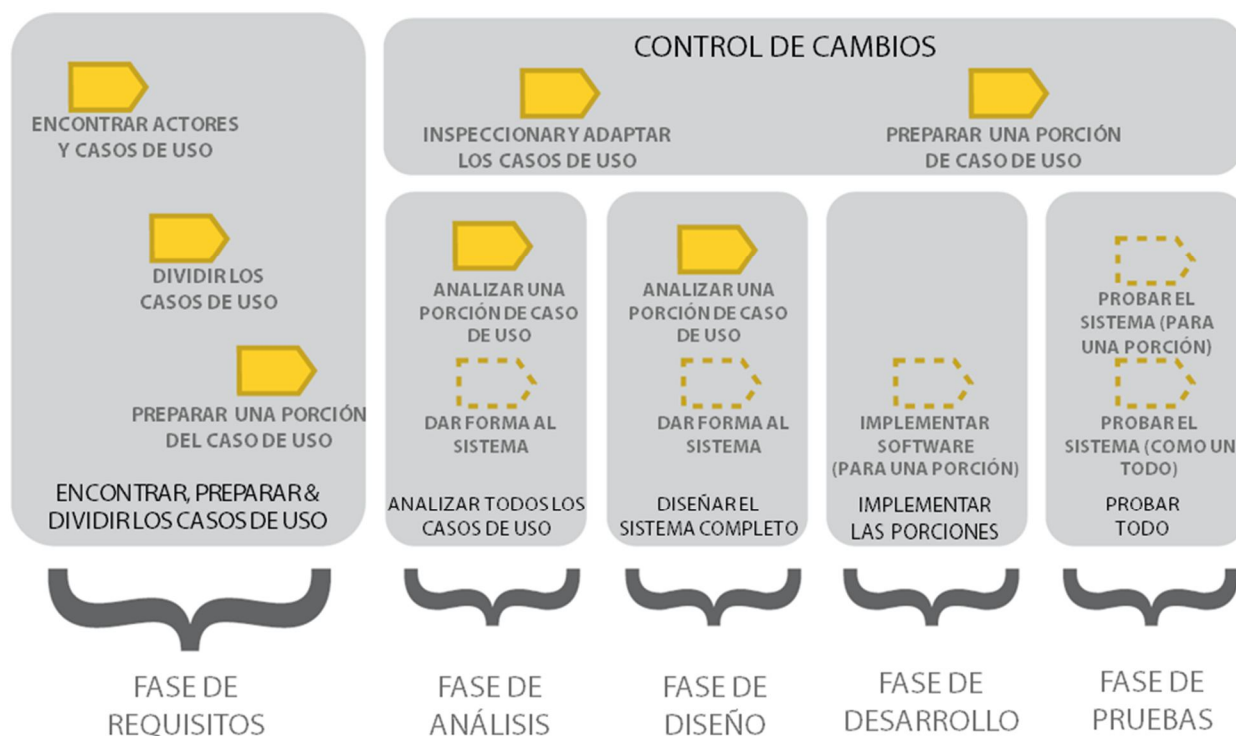
**Figura 23:** El Tablero Kanban Revisado de un Equipo Mostrando Estados de Terminación.

Cuando usted adopta un enfoque en cascada, los casos de uso no se trabajan ni afinan continuamente para permitir que el sistema final emerja, sino que todos se definen de una sola vez al inicio del trabajo. Luego, se dirigen en sincronización perfecta a lo largo de las otras fases del desarrollo, cada una de las cuales se enfoca en un tipo de actividad al tiempo. La secuencia típica de actividades para un enfoque en cascada se muestra en la **Figura 24**.

Aun dentro del entorno más estricto de cascada, hay cambios que ocurren durante el desarrollo del software mismo. En vez de abarcar y fomentar el cambio, los proyectos en cascada intentan controlar el cambio. Ocasionalmente, estos harán 'Inspeccionar y Adaptar los Casos de Uso' cuando una solicitud de cambio no se pueda aplazar y prepararán porciones de casos de uso adicionales para manejar cualquier cambio que se acepte. Es poco probable que se encuentren casos de uso nuevos después de la fase de requisitos, puesto que esto se consideraría un cambio muy grande en el alcance.

La naturaleza de 'una cosa a la vez' del enfoque en cascada quiere decir que la composición del equipo está cambiando continuamente en el tiempo y la habilidad de usar comunicación cara a cara para compartir historias se limita mucho. Para lidiar con esto, usted necesita subir el nivel de detalle de los productos de trabajo, yendo más allá de lo estrictamente necesario. La **Figura 25** muestra el nivel de detalle típicamente usado en los proyectos en cascada.

Dentro de cada fase de desarrollo, uno o más de los productos de trabajo se llevan hasta un nivel de detalle muy alto para asegurar que ellos 1) están completos y 2) responden todas y cada una de las preguntas que puedan surgir durante fases posteriores. En la fase de requisitos se elabora y reelabora el modelo de casos de uso para asegurar que se descubran todos los casos de uso, que todas las narrativas de los casos de uso se describen completamente y que la información de soporte se define en forma detallada. En esta etapa, se comienza a pensar en las pruebas y se formulan algunas ideas de prueba. Los casos de prueba se hacen a un lado hasta llegar a la fase de pruebas.



**Figura 24:** Actividades de Casos de Uso 2.0 para Enfoques en Cascada.

	MODELO DE CASOS DE USO	NARRATIVA DE CASOS DE USO	REALIZACIÓN DE CASOS DE USO	CASOS DE PRUEBA	INFORMACIÓN DE SOPORTE
BOSQUEJO:		BREVEMENTE DESCRITA		CON IDEAS DE PRUEBA FORMULADAS	INICIADA
LO MERAMENTE ESENCIAL:	CON VALOR ESTABLECIDO	CON RESUMEN BÁSICO	CON ELEMENTOS DE IMPLEMENTACIÓN IDENTIFICADOS	CON ESCENARIO SELECCIONADO	SIMPLEMENTE DEFINIDA
AMPLIADO:	CON LÍMITES DEL SISTEMA ESTABLECIDOS	CON RESUMEN ESENCIAL	CON RESPONSABILIDADES ASIGNADAS	CON VARIABLES IDENTIFICADAS	MODELADA E ILUSTRADA
EXPANDIDO:	ESTRUCTURADO	COMPLETAMENTE DESCRITA	CON INTERACCIÓN DEFINIDA	CON VARIABLES FIJAS	DEFINIDA DETALLADAMENTE
EXPANDIDO POSTERIORMENTE:				PROGRAMADAS/AUTOMATIZADAS	

**Figura 25:** Niveles de Detalle para los Productos de Trabajo Usando un Enfoque de Cascada.

Los casos de uso y su información de soporte se entregan al equipo de análisis y diseño, que elaborará las realizaciones de los casos de uso, primero para asignar responsabilidades a los elementos del sistema y luego para definir todas las interacciones. Eventualmente, la codificación iniciará y se implementarán todos los casos de uso y las porciones de los casos de uso. Finalmente, los verificadores se involucrarán, todos los casos de prueba se definirán en detalle y las pruebas iniciarán.

La naturaleza secuencial de esta forma de trabajo puede llevar a pensar que no hay que tener un rol para las porciones de casos de uso y que manejar los casos de uso completos podría ser suficiente. Esto no es cierto porque el control más detallado que entregan las porciones de casos de uso permite que el equipo de requisitos sea mucho más específico sobre el alcance actual del sistema que se va a construir. Aun en los proyectos en cascada, es improbable que usted vaya a necesitar todas las historias de todos los casos de uso. Las porciones ayudarán también a manejar los cambios de último minuto debidos a problemas de agenda o de calidad.



---

## Casos de Uso 2.0: No son solo para un tipo de equipo

Otro aspecto importante de los Casos de Uso 2.0 es la habilidad de adaptarlos a estructuras de equipos existentes y funciones existentes en el trabajo, al mismo tiempo que elimina el desperdicio y aumenta la eficiencia. En este sentido, los Casos de Uso 2.0 no predefinen ningún rol en particular ni estructuras de equipo, sino que definen un conjunto de estados para cada uno de los elementos centrales (el caso de uso y la porción de casos de uso).

Como se ilustró en la discusión sobre los Casos de Uso 2.0 y el flujo de una sola pieza, los estados indican cuando los elementos están en descanso y se pueden transferir de una persona o equipo al siguiente. Esto permite que la práctica se emplee con equipos de todas las formas y tamaños, desde equipos multifuncionales con poca o ninguna transferencia hasta grandes redes de equipos de especialistas, donde cada estado es responsabilidad de un especialista diferente. Rastrear los estados y las transferencias de estos elementos permite que se monitoree el flujo de trabajo a lo largo del equipo (o equipos) y que los equipos adapten su forma de trabajo para aumentar continuamente su desempeño.

## Casos de Uso 2.0: Escalando para cubrir todas tus necesidades: escalando hacia adentro, escalando horizontal y verticalmente

Ningún enfoque predefinido cubre las necesidades de todo el mundo. Siendo así las cosas, necesitamos ser capaces de escalar nuestro empleo de los Casos de Uso 2.0 en diferentes dimensiones:

6. Los casos de uso se escalan hacia adentro para proporcionar más guía a los practicantes menos experimentados (desarrolladores, analistas, verificadores, etc.) o para los practicantes que quieren o necesitan más guía.
7. Se escalan horizontalmente para cubrir todo el ciclo de vida, cubriendo no solo análisis, diseño, codificación y pruebas, sino también el uso operacional y el mantenimiento.
8. Se escalan verticalmente para soportar sistemas grandes y muy grandes, así como sistemas de sistemas: sistemas corporativos, líneas de productos y sistemas por capas. Tales sistemas son complejos y típicamente los desarrollan muchos equipos trabajando en paralelo, en sitios distintos, posiblemente de diferentes compañías y reutilizando muchos sistemas legados o soluciones empaquetadas.

Independientemente de la complejidad del sistema que usted desarrolle, siempre se inicia de la misma forma: identificando los casos de uso más importantes y creando un panorama general que resuma lo que se necesita construir. Luego, se pueden adaptar los Casos de Uso 2.0 para cubrir las necesidades emergentes del equipo. De hecho, los Casos de Uso 2.0 insisten en que, continuamente, se inspeccione y se adapte su uso para eliminar desperdicio, aumentar el rendimiento y mantener el ritmo con las demandas cambiantes del equipo.

---

# Conclusión

Los Casos de Uso 2.0 existen como una práctica probada y bien definida, compatible con muchas otras prácticas de desarrollo de software tales como Integración Continua, Arquitectura Intencional y Desarrollo Dirigido por Pruebas. También, trabaja con todas las prácticas populares de administración. En particular, es tan liviano y flexible como para apoyar a equipos que trabajan de una manera ágil. También, tiene la plenitud y el rigor requerido para soportar los equipos que se requieren para trabajar de una manera más formal o en un entorno en cascada.

Los Casos de Uso 2.0 son:

- **Livianos:** tanto en su definición como en su aplicación
- **Escalables:** y ajustables para equipos y sistemas de todos los tamaños
- **Versátiles:** y ajustables para todos los tipos de sistemas y enfoques de desarrollo
- **Fáciles de usar:** los modelos de casos de uso pueden ponerse rápidamente en su lugar y se pueden crear porciones para cubrir las necesidades de los equipos.

Los Casos de Uso 2.0 son libres y se ofrecen al público en esta guía. Las secciones 'cosas con las cuales trabajar' y 'cosas por hacer' de los Casos de Uso 2.0 no son negociables y, aunque es posible implementar los Casos de Uso 2.0 en parte, los resultados serían indeterminados y la práctica que se emplee no sería la de los Casos de Uso 2.0.

Esta guía se mantiene deliberadamente liviana y puede no ser suficiente para que usted adopte la práctica. Puede disponer de información adicional en forma de una práctica documentada completamente, publicada en la Web de [www.ivarjacobson.com](http://www.ivarjacobson.com). Esta Web se ofrece como un sitio independiente o como una extensión que se puede conectar con EssWork.

Esta es la primera de muchas publicaciones sobre los Casos de Uso 2.0 y podemos esperar ver muchos otros artículos, documentos y blogs en la materia publicados en [www.ivarjacobson.com](http://www.ivarjacobson.com).

---

# Apéndice 1: Productos de Trabajo

Este apéndice proporciona definiciones e información adicional de los productos de trabajo que emplean los Casos de Uso 2.0.

Los productos de trabajo cubiertos son:

- Información de soporte
- Casos de Prueba
- Modelo de Casos de Uso
- Narrativa de Casos de Uso
- Realización de Casos de Uso

# Información de Soporte

El propósito de la información de soporte es capturar términos importantes que se emplean para describir el sistema y todos los demás requisitos que no caben dentro del modelo de casos de uso.

La información de soporte:

- Ayuda a asegurar un entendimiento común de la solución especificada.
- Se enfoca en los conceptos y términos necesarios que todos los interesados en el trabajo deben entender y, en particular, en aquellos términos que referencian los casos de uso.
- Captura aquellos requisitos globales importantes y los atributos de calidad que no se relacionan con ningún caso de uso en particular, tales como las plataformas de apoyo y la disponibilidad del sistema.
- Detalla cualquier estándar que sea necesario seguir. Por ejemplo, codificación, presentación, lenguaje, seguridad y otros estándares de la industria que apliquen al sistema.
- Ayuda a identificar historias adicionales que no se identifican directamente de los casos de uso, tales como aquellas que se emplearán para demostrar las distintas plataformas soportadas o los niveles deseados de disponibilidad.

El rol de la información de soporte es apoyar la evolución de los casos de uso y la implementación de las porciones de casos de uso. Captúrela para complementar su modelo de casos de uso y evitar problemas de comunicación entre los miembros del equipo. La información puede venir desde muchas fuentes, tales como otros documentos de requisitos, especificaciones y discusiones con los interesados y expertos del dominio. También, es posible incluir modelos del dominio, de procesos y otros modelos del negocio si estos son útiles para entender el modelo de casos de uso y el sistema que describe.

La información de soporte se puede documentar a varios niveles de detalle, desde un conjunto simple de definiciones básicas, hasta un conjunto detallado y completamente descrito de definiciones, estándares y atributos de calidad.

La información de soporte se puede presentar en los siguientes niveles de detalle:



**Iniciada:** un nivel intermedio de detalle que indica que lo que se incluye es solo un resumen de los términos más obvios y de las áreas a tratar.

Se añadirá mayor detalle si la información apoya la identificación exitosa y la preparación de las porciones de casos de uso correctas.



**Simplemente Definida:** Todos los términos que referencian las narrativas de los casos de uso se deben definir y los atributos de calidad globales del sistema se deben especificar claramente. En este nivel de detalle, estas características se capturan como simples listas de sentencias declarativas, como las que se emplean en el glosario que acompaña este libro electrónico.

Este es el nivel más liviano de detalle que proporciona soporte para el desarrollo de las porciones de los casos de uso. También clarifica los requisitos globales del sistema, permitiendo que el equipo diga si el sistema que implementa las porciones es verdaderamente usable y no es solo demostrable. Es adecuado para la mayoría de los equipos, particularmente para aquellos que colaboran estrechamente con sus usuarios y son capaces de cubrir cualquier detalle faltante hablando con ellos.



**Modelada e Ilustrada:** Se puede adicionar más detalle a la información de soporte transformando las definiciones básicas en modelos que, precisamente, capturen las definiciones, sus atributos y sus relaciones y proporcionando ejemplos del mundo real para clarificar las cosas.

---

En este nivel de detalle vamos más allá de las definiciones simples y empezamos a usar técnicas complementarias, tales como catálogos de reglas del negocio, modelado de información y modelado del dominio. Es particularmente útil para soportar aquellos modelos de casos de uso, donde un malentendido en los requisitos podría tener consecuencias graves en seguridad, finanzas o asuntos legales.



**Definida Detalladamente:** Algunas veces, es necesario clarificar la información proporcionando explicaciones más detalladas y material de apoyo como ejemplos detallados, fuentes y referencias cruzadas.

En este nivel de detalle, la información de soporte es más complicada, con más precisión, con referencias cruzadas y usa técnicas de especificación formal.

La información de soporte proporciona una ubicación central para buscar términos y abreviaturas que pueden ser nuevas para el equipo y para encontrar los atributos de calidad globales que todos en el equipo deberían entender. Es un complemento esencial para el modelo mismo de casos de uso. Sin la información de soporte será imposible entender lo que significa para el sistema que sea usable y esté listo para su uso.

La información de soporte se representa, usualmente, como una lista simple de términos y sus definiciones. Frecuentemente, la lista se divide en secciones tales como: definiciones de términos, reglas del negocio, restricciones operacionales, restricciones de diseño, estándares y requisitos generales del sistema. La lista se puede publicar como parte de un sitio Wiki para simplificar su acceso y mantenimiento.

# Casos de Prueba

El propósito de un caso de prueba es proporcionar una definición clara de los que significa completar una porción de los requisitos. Un caso de prueba define un conjunto de entradas de prueba y los resultados esperados, con el propósito de evaluar si un sistema trabaja correctamente o no.

Los casos de prueba:

- Proporcionan los bloques de construcción para el diseño y la implementación de las pruebas
- Proporcionan un mecanismo para completar y verificar los requisitos
- Permiten que se especifiquen las pruebas antes de iniciar la implementación
- Proporcionan formas de evaluar la calidad de un sistema

Los casos de prueba son partes importantes, aunque a muchas veces descuidadas, de los casos de uso. Los casos de prueba proporcionan una definición verdadera de lo que se supone que haga un sistema para soportar un caso de uso. Los casos de prueba son particularmente importantes cuando comenzamos a dividir los casos de uso, porque dan a los desarrolladores una definición clara de lo que significa implementar exitosamente una porción de un caso de uso.

Los casos de prueba se pueden usar con muchas formas de prácticas de requisitos, incluyendo casos de uso, historias de usuario y requisitos declarativos. En todos los casos, se le debe presentar al verificador una porción de los requisitos a probar, con un claro inicio y fin que posibiliten la derivación de un escenario de pruebas ejecutable.

Los casos de prueba se pueden presentar en los siguientes niveles de detalle:



**Con Ideas de Prueba Formuladas:** El nivel más liviano de detalle; solo captura la idea inicial que informará el caso de prueba. Cuando se define un caso de prueba, se debe aclarar cuál es la idea detrás del caso de prueba y a cuál porción de los requisitos aplica.

Se añadirá mayor detalle si el caso de prueba se puede ejecutar.



**Con Escenario Seleccionado:** Para ser capaz de correr un caso de prueba, se le debe presentar un escenario que el verificador pueda ejecutar. La estructura de la narrativa del caso de uso asegura que cada porción del caso de uso presentará al verificador uno o más escenarios de prueba candidatos. El arte de crear casos de prueba efectivos es seleccionar el subconjunto correcto de escenarios de prueba potenciales para cumplir con la idea de prueba y definir claramente cuando la porción está hecha.

Este es el nivel más liviano que proporciona un caso de prueba ejecutable. Una vez que se selecciona el escenario, se define el caso de prueba lo suficiente como para apoyar pruebas exploratorias e investigativas. Esto puede ser muy útil al principio del ciclo de vida, cuando la visión que obtenemos al probar el sistema es invaluable pero la especificación (y la solución) puede no ser lo suficientemente estable para soportar una prueba formal con un guión.



**Con Variables Identificadas:** Un caso de prueba toma algunas entradas, manipula los estados del sistema y produce algunos resultados. Estas variables aparecen como entradas, estados internos y salidas en los requisitos. En este nivel de detalle, se definen explícitamente los rangos aceptables para las variables clave involucradas en el escenario.

Este nivel de detalle es apropiado para aquellos casos de prueba donde solicitar la opinión del verificador es una parte esencial de prueba, por ejemplo cuando nos comprometemos con pruebas de usabilidad. También se puede usar cuando se necesita más estructura para pruebas exploratorias e investigativas.



**Con Variables Fijas:** El caso de prueba se puede elaborar posteriormente, proporcionando explícitamente valores específicos para todas las variables involucradas en el caso de prueba.



---

Este nivel de detalle es apropiado para los casos de prueba manuales, ya que se cuenta con toda la información que un verificador inteligente necesita para ejecutar repetida y consistentemente el caso de prueba.



**Programada/Automatizada:** Si un caso de prueba se va a emplear muchas veces o apoyará muchas pruebas distintas, vale la pena hacer el esfuerzo de programarlo o automatizarlo completamente.

En este nivel de detalle, el caso de prueba se puede ejecutar sin ninguna intervención o sin tomar ninguna decisión adicional.

Los casos de prueba constituyen el producto de trabajo más importante asociado con un caso de uso. Recuerde que son los casos de prueba los que definen lo que significa completar el desarrollo de un caso de uso, no la narrativa del caso de uso. En cierta forma, los casos de prueba son la mejor forma de requisitos que usted puede tener.

Los casos de prueba se usarán durante el tiempo de vida del sistema: se usan no solo durante la implementación de los casos de uso, sino también como la base de las pruebas de regresión y otros chequeos de calidad. La buena noticia es que la estructura de los casos de uso y las narrativas de los casos de uso naturalmente conducen a casos de prueba bien formados, robustos y resistentes; unos que duren mientras el sistema continúe soportando los casos de uso.

Las narrativas de los casos de uso son colecciones de historias que el sistema debe soportar y, para cada historia descrita en la narrativa del caso de uso, habrá al menos un caso de prueba. Usted crea los casos de prueba al mismo tiempo que las narrativas de los casos de uso, como parte de la preparación de una porción de caso de uso para desarrollo.

---

# Modelo de Casos de Uso

Un modelo de casos de uso es un modelo de todas las formas útiles de emplear un sistema y el valor que ellas proporcionan. El propósito de un modelo de casos de uso es capturar todas las formas útiles de emplear un sistema, en un formato accesible que captura los requisitos de un sistema y se puede usar para conducir su desarrollo y pruebas.

Un modelo de casos de uso:

- Permite a los equipos acordar la funcionalidad y las características que requiere un sistema
- Establece claramente los límites y el alcance del sistema, proporcionando un panorama general de sus actores (que están fuera del sistema) y de los casos de uso (que están dentro del sistema)
- Habilita el manejo ágil de requisitos

Un modelo de casos de uso está, principalmente, hecho de un conjunto de actores y casos de uso y diagramas que ilustran sus relaciones. Los modelos de casos de uso se pueden capturar de muchas formas distintas que incluyen una Wiki, un tablero, diapositivas de *PowerPoint*, un documento de MS *Word* o una herramienta de modelado.

El modelo de casos de uso se puede preparar en diferentes niveles de detalles:



**Con Valor Establecido:** El primer paso hacia un modelo completo de casos de uso es identificar los actores y casos de uso más importantes (los primarios), que son los que dan valor al sistema.

Este es el nivel más liviano de detalle. Es ajustable para la mayoría de los proyectos, particularmente aquellos que adicionan nueva funcionalidad de sistemas existentes donde hay poco o ningún valor para modelar todas las cosas que el sistema ya hace.



**Con Límites del Sistema Establecidos:** Los actores y casos de uso primarios capturan los motivos principales por los cuales se construye el sistema y cómo los usuarios obtendrán valor del sistema. Quizás no proporcionan suficiente valor para configurar y ejecutar el sistema. En estos casos, los actores y casos de uso secundarios son necesarios para habilitar y soportar la operación efectiva del sistema.

Este nivel de detalle es útil cuando se modelan sistemas completamente nuevos o nuevas generaciones de sistemas existentes. En este nivel de detalle todos los actores y casos de uso del sistema se identifican y se modelan.



**Estructurado:** El modelo de casos de uso, frecuentemente, contiene información redundante, como secuencias o patrones de interacción. Estructurar el modelo de casos de uso es la manera de lidiar con estas redundancias.

Para sistemas grandes y complejos, especialmente aquellos que se emplean para proporcionar funcionalidad similar dentro de muchos contextos diferentes, estructurar el modelo de casos de uso puede ayudar a entender y eliminar desperdicio y ayuda a encontrar elementos reusables.

A medida que el modelo de casos de uso muestra claramente el valor que los interesados recibirán del sistema nuevo o actualizado, entonces está haciendo su trabajo. Se debe tener cuidado al adicionar detalles al modelo. Únicamente avance a los niveles “Con Límites del Sistema Establecido” o “Estructurado” si es claro que estos niveles de detalle van a adicionar valor y a ayudar a entregar el nuevo sistema de manera más eficiente.

---

# Narrativa del Caso de Uso

El propósito de la narrativa de un caso de uso es contar la historia de como el sistema y sus actores trabajan en conjunto para alcanzar una meta específica.

Las narrativas de los casos de uso:

- Resumen las historias que se emplean para explorar los requisitos e identifican las porciones de los casos de uso
- Describen una secuencia de acciones, incluyendo variantes que un sistema y sus actores pueden ejecutar para alcanzar una meta
- Se representan como un conjunto de flujos que describen como un actor emplea un sistema para alcanzar una meta y lo que el sistema hace para ayudar al actor a alcanzar esa meta
- Capturan la información de los requisitos necesarios para apoyar las otras actividades del desarrollo

Las narrativas de los casos de uso se pueden capturar de muchas maneras, incluyendo una Wiki, tarjetas indexadas, documentos de MS *Word* o dentro de una de las muchas herramientas de administración, de manejo de requisitos o de modelado de requisitos comercialmente disponibles.

Las narrativas de los casos de uso se pueden desarrollar a distintos niveles de detalle, desde un simple resumen, identificando el flujo básico y las variantes más importantes, hasta una especificación completa, altamente detallada, que defina las acciones, entradas y salidas involucradas en la ejecución del caso de uso. Las narrativas de los casos de uso se pueden preparar en los siguientes niveles de detalle:



**Brevemente Descrita:** El nivel más liviano de detalle que solo captura la meta del caso de uso y el actor que lo inicia.

Este nivel de detalle es apropiado para aquellos casos de uso que usted decida no implementar. Se necesitará más detalle si el caso de uso se va a dividir para su implementación.



**Con Resumen Básico:** El caso de uso se debe resumir para entender su tamaño y complejidad. Este nivel de detalle también posibilita un manejo efectivo del alcance, puesto que el resumen permite que las distintas partes del caso de uso se prioricen unas con otras y, si es necesario, se destinen a distintas versiones.

Este es el nivel más liviano de detalle que posibilita que el caso de uso se divida y desarrolle. Es ajustable para aquellos equipos que trabajan en estrecha colaboración con sus usuarios y que son capaces de completar cualquier detalle faltante vía conversaciones y la terminación de los casos de prueba acompañantes.



**Con Resumen Esencial:** Algunas veces es necesario aclarar las responsabilidades del sistema y sus actores mientras se trabaja en el caso de uso. Un resumen básico captura sus responsabilidades, pero no define claramente qué partes del caso de uso son responsabilidad del caso de uso y qué partes son responsabilidad de los actores.

En este nivel de detalle, la narrativa se convierte en una descripción del diálogo entre el sistema y sus actores. Es particularmente útil cuando se establece la arquitectura de un nuevo sistema o se intenta establecer una nueva experiencia de usuario.



**Completamente Descrita:** Las narrativas de los casos de uso se pueden usar para proporcionar una especificación bastante detallada de los requisitos, haciéndolas evolucionar hasta su nivel de detalle más completo: completamente descritas. El detalle extra puede ser necesario para cubrir la falta de experiencia dentro del equipo, la falta de acceso a los interesados o para comunicar efectivamente requisitos complejos.

---

Este nivel de detalle es particularmente útil para aquellos casos de uso donde un malentendido del contenido podría tener consecuencias severas de seguridad, finanzas o asuntos legales. También puede ser útil cuando hacemos desarrollo de software externo o por subcontratación.

La narrativa del caso de uso es un producto de trabajo bastante flexible que se puede expandir para capturar la cantidad de detalle necesario para cualquier circunstancia. Si usted es parte de un equipo pequeño que trabaja de manera colaborativa con el cliente o de un proyecto exploratorio, entonces los resúmenes básicos proporcionarán una forma liviana de descubrir los requisitos. Si usted está trabajando en un entorno más rígido donde hay poco acceso a los expertos, entonces las narrativas con resúmenes esenciales o las completamente descritas se pueden usar para cubrir los vacíos en el conocimiento del equipo.

No todos los casos de uso se deben especificar al mismo nivel de detalle; no es extraño que los casos de uso más importantes y riesgosos tengan más detalle que los otros. Lo mismo va para las secciones de la narrativa del caso de uso: las partes más importantes, complejas o riesgosas de un caso de uso, frecuentemente, se describen con más detalle que las otras.

---

# Realización del Caso de Uso

El propósito de la realización de un caso de uso es mostrar como los elementos del sistema, tales como componentes, programas, procedimientos almacenados, archivos de configuración y tablas de la base de datos, colaboran en conjunto para ejecutar un caso de uso.

Las realizaciones de los casos de uso:

- Identifican los elementos del sistema involucrados en los casos de uso
- Capturan las responsabilidades de los elementos del sistema cuando ejecuta un caso de uso
- Describen como los elementos del sistema interactúan para ejecutar el caso de uso
- Traducen el lenguaje del negocio que se emplea en las narrativas del caso de uso en lenguaje del desarrollador usado para describir la implementación del sistema.

Las realizaciones de casos de uso son increíblemente útiles y se pueden usar para conducir la creación y validación de muchas de las distintas vistas que los equipos usan para diseñar y construir sus sistemas. Por ejemplo, los diseñadores de la interfaz de usuario usan las realizaciones de los casos de uso (en forma de tableros de historias) para explorar el impacto de los casos de uso en la interfaz de usuario. Los arquitectos usan las realizaciones de los casos de uso para analizar los casos de uso significativos para la arquitectura y evaluar si la arquitectura cumple su propósito o no. Las realizaciones de los casos de uso se presentan en muchos formatos distintos: el formato de la realización es completamente dependiente de las prácticas de desarrollo del equipo. Formas comunes de expresar realizaciones de casos de uso incluyen tablas simples, tableros de historias, diagramas de secuencias, diagramas de colaboración y diagramas de flujos de datos. El aspecto importante es que el equipo cree una realización para identificar cuáles elementos del sistema se involucran en la implementación del caso de uso y cómo cambiarán.

Haga la realización de un caso de uso para cada caso de uso, con el fin de identificar los elementos del sistema involucrados en su ejecución y, más importante aún, para evaluar cuanto se tendrán que modificar. Usted puede pensar en las realizaciones de casos de uso como las que proporcionan el 'como' para complementar las narrativas del caso de uso, que constituyen el 'que'.

Las realizaciones de los casos de uso se pueden presentar en los siguientes niveles de detalle:



**Con Elementos de Implementación Identificados:** El nivel más bajo de detalle que solo captura los elementos del sistema, tanto nuevos como existentes, que participarán en el caso de uso.

Este nivel de detalle es apropiado para equipos pequeños, trabajando en estrecha colaboración y desarrollando sistemas simples con una arquitectura conocida. Usted puede adicionar más detalle si su sistema es complejo o si su equipo es grande o distribuido.



**Con Responsabilidades Asignadas:** Para permitir que el equipo sea capaz de actualizar en paralelo los elementos del sistema afectados o para soportar múltiples porciones, los desarrolladores necesitan entender las responsabilidades de los elementos individuales. Las responsabilidades proporcionan una definición de alto nivel de lo que cada elemento necesita hacer, almacenar y rastrear.

Este nivel de detalle es apropiado para situaciones donde cada porción del caso de uso toca uno o múltiples elementos del sistema, o donde las porciones las desarrollarán múltiples desarrolladores trabajando en paralelo. También, se debería usar cuando la arquitectura del sistema es inmadura y las responsabilidades en general de los elementos del sistema todavía se tienen que entender.



**Con Interacción Definida:** Para proporcionar una definición completa y sin ambigüedad de los cambios que requiere cada elemento del sistema involucrado en el caso de uso, la

---

realización del caso de uso debe incluir detalles de todas las interfaces e interacciones involucradas en la ejecución del caso de uso.

Este nivel de detalle es particularmente útil para aquellos casos de uso donde el diseño del sistema es complejo o desafiante. También, puede ser útil cuando desarrolladores con poco o ningún conocimiento del diseño del sistema van a desarrollar los elementos del sistema y cuando no hay acceso a diseñadores experimentados a quienes se les pueda enviar el diseño para refactorización o modificación. También es útil cuando tratamos con desarrolladores sin experiencia que todavía están aprendiendo su oficio.

La realización del caso de uso es un producto de trabajo muy flexible, pues los equipos pueden expandir sus realizaciones para adicionar más detalle a medida que lo necesiten. Si un equipo pequeño está haciendo todo su análisis y diseño de manera colaborativa, entonces unas realizaciones simples y livianas serán suficientes. Si un equipo grande, incapaz de tener muchas sesiones colaborativas, está desarrollando un sistema complejo, entonces unas realizaciones más detalladas ayudarán en la comunicación y, con esto, asegurarán que todos los desarrolladores tengan toda la información que necesitan para entregar los elementos del sistema e implementar sus porciones de casos de uso de manera exitosa.



---

# Glosario de Términos

**Actor:** un actor define un rol que un usuario puede desempeñar cuando interactúa con el sistema. Un usuario puede ser un individuo u otro sistema. Los actores tienen un nombre y una breve descripción y se asocian con los casos de uso, con los cuales interactúan.

**Aplicación:** software de computador diseñado para ayudar a los actores en la ejecución de tareas específicas.

**Caso de Prueba:** define un conjunto de entradas de prueba y los resultados esperados, con el propósito de evaluar si un sistema trabaja correctamente o no.

**Caso de Uso:** define todas las formas de emplear un sistema para lograr una meta en particular para un usuario en particular.

**Casos de Uso 2.0:** una práctica escalable y ágil que usa casos de uso para capturar un conjunto de requisitos y conduce el desarrollo incremental de un sistema que cumple con esos requisitos.

**Cliente:** el interesado que está pagando por el desarrollo del sistema o quien esperamos que compre el sistema una vez se termine.

**Diagrama de Casos de Uso:** un diagrama que muestra un número de actores y casos de uso y las relaciones entre unos y otros.

**Elemento del Sistema:** miembro de un conjunto de elementos que constituyen un sistema (ISO/IEC 15288:2008).

**Flujo:** una descripción de alguna narrativa completa o parcial de un caso de uso. Siempre hay al menos un flujo básico y puede haber flujos alternativos.

**Flujo Alternativo:** descripción de un comportamiento variante u opcional como parte de la narrativa de un caso de uso. Los flujos alternativos se definen relativos al flujo básico del caso de uso.

**Flujo Básico:** la descripción del camino normal y esperado del caso de uso. Este es el camino que toma la mayoría de los usuarios la mayor parte del tiempo; es la parte más importante de la narrativa del caso de uso.

**Historia:** una forma de emplear un sistema que es de valor para un usuario u otro interesado. En Casos de Uso 2.0, una historia se describe como parte de la narrativa del caso de uso, uno o más flujos y requisitos especiales, y uno o más casos de prueba.

**Interesado:** una persona, grupo u organización que afecta o se afecta con el sistema de software.

**Modelo de Casos de Uso:** un modelo de todas las formas útiles de emplear un sistema y del valor que este proporcionará. Un modelo de casos de uso se conforma, principalmente, con un conjunto de actores y un conjunto de casos de uso, y con diagramas que ilustran las relaciones entre unos y otros.

**Narrativa del Caso de Uso:** una descripción de un caso de uso que cuenta la historia de cómo el sistema y sus actores trabajan en conjunto para alcanzar una meta en particular. Incluye una secuencia de acciones (incluyendo variantes) que el sistema y sus actores pueden ejecutar para lograr la meta.

**Porción de Caso de Uso:** una porción de caso de uso es una o más historias seleccionadas de un caso de uso para formar un elemento de trabajo que es de valor claro para el cliente.

**Programación Orientada a Aspectos:** una técnica de programación que apunta a aumentar la modularidad, permitiendo la separación de intereses transversales (ver [http://en.wikipedia.org/wiki/Aspect-oriented\\_programming](http://en.wikipedia.org/wiki/Aspect-oriented_programming)).

---

**Requisitos:** lo que el sistema de software debe hacer para satisfacer a los interesados.

**Separación de Asuntos:** el proceso para dividir un sistema, de forma que se minimice el solapamiento en funcionalidad (ver [http://en.wikipedia.org/wiki/Separation\\_of\\_concerns](http://en.wikipedia.org/wiki/Separation_of_concerns)).

**Sistema:** un grupo de cosas o partes que trabajan en conjunto o conectadas de alguna manera para formar un todo. Típicamente, se emplea para referirse al sujeto del modelo de casos de uso: el producto a construir.

**Sistema de Software:** un sistema hecho de software, hardware e información digital y que proporciona su valor primario mediante la ejecución del software.

Un sistema de software puede ser parte de una solución más grande de software, hardware, de negocios o social.

**Usuario:** un interesado que interactúa con el sistema para alcanzar sus metas.

---

# Agradecimientos

## General

Los Casos de Uso 2.0 se basan en mejores prácticas que la industria aceptó, empleó o probó por décadas. Nos gustaría agradecer a decenas de cientos de personas que usan casos de uso cada día en sus proyectos y, en particular, a aquellos que comparten sus experiencias dentro y fuera de sus propias organizaciones. Sin todo su trabajo duro y entusiasmo no tendríamos la motivación o conocimiento para intentar esta evolución de la técnica. Esperamos que usted encuentre útil este corto libro electrónico y continúe inspeccionando y adaptando la forma en que aplica casos de uso.

## Personas

Nos gustaría agradecer también a todos los que directamente contribuyeron con la creación de este libro electrónico, incluyendo, sin ningún orden particular, a Paul McMahon, Richard Schaff, Eric Lopes Cardozo, Svante Lidman, Craig Lucia, Tony Ludwig, Ron Garton, Burkhard Perens-Golomb, Arran Hartgroves, James Gamble, Brian Hooper, Stefan Bylund y a Pan-Wei Ng

---

# Bibliografía

## Object Oriented Software Engineering: A Use Case Driven Approach

**Ivar Jacobson, Magnus Christerson, Patrik Jonsson, Gunnar Overgaard**

El libro original que introdujo los casos de uso al mundo.

- Editor: Addison-Wesley Professional; Revised edition (julio 10, 1992)
- ISBN-10: 0201544350
- ISBN-13: 978-0201544350

## The Object Advantage: Business Process Reengineering With Object Technology

**Ivar Jacobson, Maria Ericsson, Agneta Jacobson**

La guía definitiva para usar casos de uso para reingeniería de procesos del negocio.

- Editor: Addison-Wesley Professional (septiembre 30, 1994)
- ISBN-10: 0201422891
- ISBN-13: 978-0201422894

## Software Reuse: Architecture, Process and Organization for Business Success

**Ivar Jacobson, Martin Griss, Patrik Jonsson**

Una guía detallada para la reutilización del software, incluyendo una guía profunda sobre el uso de casos de uso para el desarrollo de líneas de producto y sistemas de sistemas interconectados.

- Editor: Addison-Wesley Professional (junio 1, 1997)
- Idioma: English
- ISBN-10: 0201924765
- ISBN-13: 978-0201924763

## Use-Case Modeling

**Kurt Bittner y Ian Spence**

La guía definitiva para crear modelos de casos de uso y escribir buenos casos de uso.

- Editor: Addison-Wesley Professional; 1 edición (agosto 30, 2002)
- ISBN-10: 0201709139
- ISBN-13: 978-0201709131

## Aspect-Oriented Software Development with Use Cases

**Ivar Jacobson y Pan-Wei Ng**

El libro que introdujo al mundo las porciones de casos de uso en su apariencia previa como módulos de casos de uso.

- Editor: Addison-Wesley Professional; 1 edición (enero 9, 2005)
- ISBN-10: 0321268881
- ISBN-13: 978-0321268884

---

## Acerca de los Autores

### Ivar Jacobson

El Dr. Ivar Jacobson es el padre de los componentes y de la arquitectura de componentes, los casos de uso, el desarrollo de software orientado a aspectos, la ingeniería de negocios moderna, el Lenguaje de Modelado Unificado y el Proceso Unificado Racional. Su última contribución a la industria es un concepto de una práctica formal que promueve prácticas como los 'ciudadanos de primera clase' del desarrollo de software y ve a los procesos simplemente como una composición de prácticas. Él es el autor principal de seis libros influyentes y *best-sellers*. Es conferencista principal en muchas conferencias importantes alrededor del mundo y ha entrenado a varios consultores en mejoramiento de procesos.

### Ian Spence

Ian es Oficial de Tecnología en Jefe (CTO por sus siglas en inglés) en Ivar Jacobson International, donde se especializa en la aplicación ágil del Proceso Unificado. Es un practicante RUP certificado, *Scrum Master* y un entrenador experimentado que trabajó en cientos de proyectos para introducir técnicas iterativas y ágiles. Tiene más de 20 años de experiencia en la industria del software y cubre todo el ciclo de vida del software, incluyendo captura de requisitos, arquitectura, análisis, diseño, implementación y gerencia de proyectos. Sus áreas de especialidad son manejo iterativo de proyectos, trabajo ágil en equipo y manejo de requisitos con casos de uso. En su rol como CTO, Ian contribuye con la dirección de Ivar Jacobson International y trabaja con la Oficina de Tecnología de la compañía para definir la próxima generación de prácticas inteligentes y activas de desarrollo de software. Es el líder del proyecto y arquitecto del proceso para el desarrollo del Proceso Unificado Esencial y las prácticas que contiene. Cuando no está trabajando en investigación, capturando y definiendo prácticas, él pasa su tiempo asistiendo a compañías en la creación y ejecución de programas de cambio para mejorar su capacidad de desarrollo de software. Es coautor de los libros de Addison Wesley "*Use Case Modeling*" y "*Managing Iterative Software Development Projects*".

### Kurt Bittner

Kurt es Oficial de Tecnología en Jefe en Ivar Jacobson International, Américas. Viene trabajando en la industria del software por más de 25 años en una variedad de roles incluyendo desarrollador, líder de equipo, arquitecto, gerente de proyectos y líder de negocio. Lideró proyectos ágiles, manejó una gran división de una compañía de desarrollo de software, sobrevivió y prosperó varios emprendimientos, manejó una adquisición y trabajó con clientes en una variedad de industrias, incluyendo la aeroespacial, financiera, energía y electrónica. Fue un colaborador clave en el desarrollo inicial del Proceso Unificado Racional, así como del más reciente proyecto *Jazz* de IBM. Su experiencia incluye trabajos significativos en banca y finanzas, diseño y arquitectura de sistemas de bases de datos relacionales, y consultor y mentor de una amplia variedad de clientes en estrategias y enfoques de mejoramiento de desarrollo de software. Es coautor de "*Use Case Modeling*", "*Managing Iterative Software Development Projects*" y de "*The Economics of Iterative Software Development*".

---

## Acerca de los Traductores

### Luis Antonio Salazar

Luis tiene 25 años de experiencia en la industria de software. Actualmente es Consultor de Ingeniería de Software en Intergrupo Medellín (Colombia), donde ha trabajado los últimos 13 años. Es *IBM Certified Specialist for Rational Unified Process* e *IBM Certified Specialist for Requirements Engineering With Use Cases*. Luis conduce evaluaciones de la situación actual de los procesos de desarrollo de software y propone e implanta soluciones para el mejoramiento de los procesos de TI, incluyendo estrategias para gerencia de proyectos, administración de riesgos e ingeniería de requisitos, arquitectura del software y análisis y diseño del sistema con RUP, CMMI y métodos ágiles de software. Es miembro de IEEE. Luis es el autor del libro *Asuntos de la Ingeniería de Software*. Su blog se puede leer en <http://www.gazafatonarioit.com>.

### Carlos Mario Zapata-Jaramillo

Carlos es Profesor Asociado de la Universidad Nacional de Colombia, sede Medellín, donde dicta diferentes asignaturas de la línea de Ingeniería de Software, como Ingeniería de Requisitos, Calidad de Software y Gestión de Proyectos de Software. Es, también, Presidente del Comité Ejecutivo del Capítulo Latinoamericano de Semat (*Software Engineering Method and Teaching*) e integrante del Comité Editorial de varias revistas a nivel nacional e internacional. Tiene más de 150 publicaciones en revistas y congresos a nivel nacional e internacional y es autor de siete libros del área de Ingeniería de Software.





### Acerca de Ivar Jacobson International

IJI es una compañía global de servicios que se especializa en mejorar el desempeño de equipos de desarrollo de software, eliminando las barreras hacia la adopción de nuevas prácticas. Por medio del suministro de personal de alto calibre, prácticas innovadoras y soluciones probadas, aseguramos que nuestros clientes logren alinear fuertemente el negocio con la TIC, equipos de alto desempeño y proyectos que entregan valor.

**[www.ivarjacobson.com](http://www.ivarjacobson.com)**

#### **Suecia**

+46 8 515 10 174  
[info-se@ivarjacobson.com](mailto:info-se@ivarjacobson.com)

#### **Los Países Bajos**

+31(0) 20 654 1878  
[info-nl@ivarjacobson.com](mailto:info-nl@ivarjacobson.com)

#### **Reino Unido**

+44 (0)207 953 9784  
[info-uk@ivarjacobson.com](mailto:info-uk@ivarjacobson.com)

#### **Asia**

+8610 824 86030  
[info-asia@ivarjacobson.com](mailto:info-asia@ivarjacobson.com)

#### **Américas**

+1 703 338 5421  
[info-usa@ivarjacobson.com](mailto:info-usa@ivarjacobson.com)