

CHAPTER 4

Relatos de Usuario

4.1 Antecedentes

El proceso de levantamiento de requisitos, también conocido como requerimientos, marca el inicio de un proyecto de desarrollo de software. Su esencia radica en descubrir las verdaderas necesidades que se deben abordar. Aunque esta tarea puede parecer sencilla en teoría, a menudo resulta complicada, ya que aquellos que están interesados en el producto o los futuros usuarios a veces carecen de claridad sobre lo que realmente se requiere, o enfrentan dificultades al transmitir sus necesidades a quienes son responsables de la construcción.

Existen dos categorías principales de requisitos: los funcionales y los no funcionales. Los requisitos funcionales se centran en definir QUÉ capacidades debe poseer el sistema. Por ejemplo, podría ser necesario que el sistema permita verificar la cantidad de productos en inventario. Por otro lado, los requisitos no funcionales imponen restricciones o condiciones adicionales, como rendimiento y usabilidad. Siguiendo el ejemplo de la consulta de inventario, un requisito no funcional podría establecer que la respuesta debe ser generada en menos de un segundo.

En el pasado, en las metodologías de desarrollo tradicionales, la etapa de levantamiento de requisitos culminaba con la creación de un documento de requisitos, que servía como un contrato entre los usuarios y los desarrolladores. Sin embargo, esta aproximación presentaba dos desafíos significativos: (1) Era inviable congelar los requisitos en un momento temprano del proyecto, cuando aún no se había iniciado; (2) Los documentos escritos solían estar plagados de ambigüedades.

Las metodologías ágiles abordan el primer inconveniente permitiendo la adición, eliminación o modificación de requisitos en cada iteración. En cuanto a

la ambigüedad en las especificaciones escritas, se asume que lo plasmado en papel es incompleto y que requerirá ajustes a medida que avance el proceso.

Un cambio fundamental en la forma de abordar el levantamiento de requisitos es la orientación hacia el usuario en lugar de centrarse exclusivamente en el sistema en desarrollo. La premisa clave es capturar cómo el sistema ayudará a cada tipo de usuario a alcanzar sus objetivos y realizar sus tareas de manera efectiva.

En esta línea, dos enfoques principales destacan:

- **Relatos de Usuario (User Stories):** Descripciones breves y sencillas de las características desde la perspectiva del usuario, enfocadas en la entrega de valor y la satisfacción de necesidades concretas.
- **Casos de Uso (Use Cases):** Escenarios detallados que narran las interacciones entre los usuarios y el sistema, ilustrando cómo se logran los objetivos del usuario a través de la aplicación.

Estos cambios en el enfoque y las metodologías permiten abordar los desafíos del levantamiento de requisitos de manera más dinámica y centrada en el usuario, contribuyendo a una comprensión más profunda de las necesidades reales y a una mayor agilidad en el proceso de desarrollo de software.

4.2 Relatos de Usuario

Dos conceptos fundamentales emergen en este contexto: la orientación hacia el usuario en los requisitos y la noción de que lo que consignamos por escrito constituye solo el punto de partida para un diálogo más amplio.

Ya hemos discutido la inherente ambigüedad del lenguaje natural al expresar requerimientos. Ilustrando esto, consideremos el caso de alguien que solicita un pastel de despedida para su amiga Alicia. Lo que el pastelero recibe es la siguiente indicación:

“Por favor, escribe ‘Hasta pronto Alicia’ en morado y coloca estrellas alrededor”. La imagen muestra la creación resultante, que cumple perfectamente con los requerimientos.

El enfoque de los relatos de usuario surge de la observación de que tener un documento compartido no garantiza un entendimiento compartido. El propósito detrás de redactar los relatos radica en cultivar, con el tiempo, una comprensión colectiva de los requerimientos. La segunda premisa subyacente, arraigada en la filosofía ágil, sostiene que el objetivo no es simplemente maximizar la cantidad de código producido (*software*), sino más bien maximizar lo que se conoce como el *outcome* y el impacto.

La propuesta es emplear una “conversación” como medio para capturar los requerimientos del usuario. De hecho, un relato puede considerarse una promesa de diálogo con el usuario.



Figure 4-1 Resultado del requerimiento: Escribe ‘hasta pronto alicia’ en morado y coloa estrellas al rededor

En sus inicios, se emplearon tarjetas de 3x5 pulgadas (muy populares en los Estados Unidos) para plasmar los relatos.

Es fundamental entender que los relatos de usuario NO CONSTITUYEN un nuevo formato de requerimientos escritos, sino más bien un vehículo para construir un entendimiento colectivo. Representan deliberaciones sobre las soluciones a los problemas que enfrentan nuestros usuarios y clientes, lo que conduce a acuerdos sobre qué construir.

Los cimientos de esta noción se sustentan en los siguientes pilares:

- Los requerimientos deben surgir de una colaboración activa.
- Los requerimientos consignados en documentos concluyen con todas las decisiones tomadas.
- Dada la naturaleza de plazos de entrega breves, es irreal especificar todos los detalles de antemano.
- La responsabilidad de plasmar los requerimientos de manera precisa se distribuye entre los actores involucrados.

Nos enfrentamos, por tanto, no únicamente a una estructura diferente respecto a los convencionales documentos de requerimientos, sino a una filosofía enteramente divergente que armoniza con los valores y principios que definen el movimiento ágil. La figura resalta de manera explícita esta contrastante perspectiva.

4.3 Estructura de un relato de usuario

Una tarjeta para cada relato. En el anverso incluir el título y la descripción. En el reverso las condiciones de satisfacción.

El título debe comenzar con un verbo y debe describir en forma muy clara el objetivo. Por ejemplo: Encontrar Cines donde se esté dando una película.

La descripción tiene una estructura que se repite para cada relato que es la siguiente:

Yo, como _____ necesito _____ para _____

El primer espacio debe ser llenado con un rol específico de entre los diversos roles que existen para los usuarios. Por ejemplo, si una aplicación va a ser usada por alumnos, ayudantes y profesores, allí pondremos “yo como alumno” o “yo como profesor”.

El segundo espacio es la meta del usuario. Qué es lo que realmente necesita que este trozo de funcionalidad realice. Por ejemplo, “yo como profesor necesito revisar las notas de mis alumnos”.

Finalmente, la última parte es la justificación. Es muy importante por dos razones:

- evita agregar funcionalidades que no son necesarias
- ayuda a entender mejor la funcionalidad

Ejemplo 1

- **Título.** Encontrar reseñas de restaurantes cercanos.
- **Descripción.** Yo, como simple usuario de la aplicación necesito revisar reseñas de los restaurantes cercanos a donde me encuentro para seleccionar un lugar adecuado para cenar.
- **Condiciones de aceptación.**
 - Escenario 1:
 - * Dado un usuario que ingresa a la aplicación.
 - * Cuando el usuario comparte su ubicación a la aplicación.
 - * Entonces el sistema debe mostrar los restaurantes mas cercanos a su ubicación ordenados por orden de recomendación.

Ejemplo 2

- **Título.** Recuperar Contraseña
- **Descripción.** Como usuario quiero recuperar mi contraseña para que pueda ingresar al sistema en caso de olvidarla.
- **Condiciones de aceptación.**
 - Escenario 1:
 - * Dado un usuario que navega a la pagina de inicio.

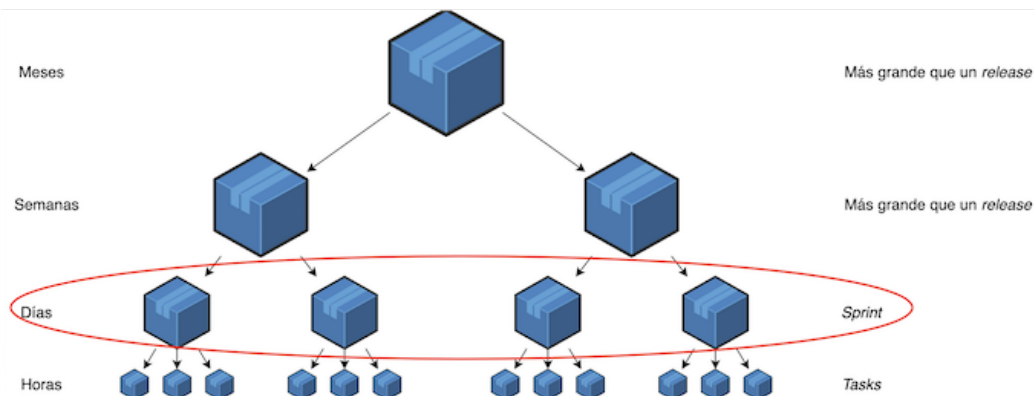
- * Cuando el usuario selecciona la opción “recuperar contraseña” e ingresa un email valido.
- * Entonces el sistema debe enviar un enlace al email ingresado.
- * Cuando el usuario entra al sistema a través del enlace.
- * Entonces el sistema le permite al usuario establecer su nueva contraseña.

Hay múltiples enfoques para redactar las condiciones de aceptación de historias de usuario; en este curso, nos centraremos en la estrategia popularmente denominada ‘Given, When, Then’ (Dado, Cuando, Entonces). En esencia, esta estructura nos capacita para plasmar las condiciones de aceptación de una historia de usuario de manera que una vez concluida la historia, sea posible verificar su funcionalidad, asegurando así que cumple con los resultados esperados a partir de un punto de partida específico.

4.4 Nivel de detalle

Una de las cosas más difíciles para quienes están comenzando a escribir relatos de usuario es encontrar el nivel de detalle que es apropiado para un relato. Por ejemplo, supongamos que mi meta es ir de paseo a San Francisco, eso parece demasiado complejo para un relato, ya que puede descomponerse en varias partes como comprar el ticket de avión, hacer la reserva de hotel, reservar un auto en el aeropuerto, etc. Si tomamos la compra de un ticket de avión puede incluso ser demasiado para un relato, ya que posiblemente tendré que evaluar posibilidades, seleccionar aerolíneas y horarios, etc.

Un criterio útil es considerar un relato cuando puede implementarse en unos pocos días y, por lo tanto, puede ser desarrollado fácilmente en un sprint. Tampoco nos interesa seguir dividiendo hasta llegar a subtarefas elementales porque el número de relatos se elevaría demasiado. Ver figura.



4.5 Directrices para escribir buenos relatos

Pautas para la redacción de relatos efectivos:

- Idealmente, deben ser independientes (permitiendo una reorganización flexible según la prioridad).
- Deben aportar valor **AL CLIENTE**.
- Deben permitir al equipo de desarrollo estimar con facilidad el esfuerzo de implementación.

Prácticas que suelen ser consideradas como “malos hábitos”:

- Evitar la expresión genérica “as a user” (ser más específico en la descripción).
- Evitar la concisión extrema en los relatos.
 - Ejemplo: “Los resultados se deben guardar en formato XML”.
 - Ejemplo: “Los resultados se deben almacenar en formato HTML”.
- Evitar relatos con interdependencias (dificultan la planificación).
- Evitar el *goldplating* (añadir características innecesarias).
 - Ejemplo: Además de las pestañas, permitir el arrastre de elementos.
- Evitar un exceso de detalle (ya que se utilizan tarjetas pequeñas para la descripción).
- Retrasar detalles específicos de la interfaz de usuario en el proceso.
- No transferir requisitos formales a los relatos de usuario.

4.6 Épicas, Temas y Relatos

La idea consiste en considerar los relatos de usuario dentro de un contexto más amplio en lugar de abordarlos de manera fragmentada. Para lograr esto, existen dos enfoques principales: las épicas y los temas.

El concepto de una épica busca describir un objetivo de mayor envergadura, por ejemplo: la compra de un boleto de avión a San Francisco o la inscripción en los cursos del semestre 1'2020 utilizando Banner. Esta meta abarcará una serie de objetivos más pequeños que se traducirán en relatos de usuario.

De esta manera, una épica se refiere a un objetivo principal que debe descomponerse en diversos relatos a ser realizados secuencialmente. Por lo general, los relatos que forman parte de una épica no tienen un valor significativo por sí mismos.

En contraposición a las épicas, un tema puede abarcar cualquier agrupación de relatos (tales como variaciones de un mismo objetivo, entre otros). En este caso, los relatos que conforman un tema poseen valor de manera independiente al tema en sí.

Para simplificarlo, podríamos considerar un tema como una recopilación de relatos relacionados de alguna manera, mientras que una épica se refiere a un relato amplio que puede dividirse en múltiples etapas.

Ejemplo de temas son por ejemplo:

- Todos los relatos que tienen que ver con análisis de datos de los clientes
- Todos los relatos que tiene que ver con acceso a la aplicación
- Todos los relatos que tienen que ver con manejo de productos

4.7 Ejercicios

Preguntas de falso y verdadero

- El levantamiento de requisitos es el inicio de un proyecto de desarrollo de software.
- Los requisitos funcionales se centran en las restricciones y condiciones adicionales de un sistema.
- Los relatos de usuario y los casos de uso son dos enfoques principales para capturar requerimientos.
- Los relatos de usuario se basan en una conversación constante para comprender los requerimientos.
- Las metodologías ágiles permiten la adición, eliminación o modificación de requisitos en cada iteración.
- Los relatos de usuario buscan maximizar la cantidad de código producido.
- El nivel de detalle apropiado para un relato de usuario es aquel que puede implementarse en un solo sprint.
- Las épicas son objetivos principales que se descomponen en varios relatos secuenciales.
- Los temas son agrupaciones de relatos que tienen valor independiente al tema en sí.
- Los relatos de usuario constituyen un nuevo formato de requerimientos escritos.

Preguntas abiertas de reflexión:

- ¿Por qué es importante orientarse hacia el usuario al definir requisitos de software?
- ¿Qué ventajas ofrecen los relatos de usuario sobre los documentos tradicionales de requerimientos?
- ¿Cómo puede un equipo asegurarse de que los relatos de usuario tengan el nivel de detalle adecuado?
- ¿Qué significa que los relatos de usuario son una “promesa de diálogo con el usuario”?