### **1. Introducción**

* En Ingeniería del Software, los **requisitos** son las **necesidades o condiciones** que debe cumplir un sistema para ser útil a los usuarios.
* Se recogen **desde el punto de vista del cliente/usuario** y sirven de base para el diseño, desarrollo y prueba del software.
* Hay **varios tipos de requisitos**: funcionales, no funcionales, de información, etc.

### **2. Historias de usuario**

* Son una forma **ágil y sencilla** de describir requisitos.
* Se centran en **lo que el usuario quiere lograr**.
* Formato típico:  
  **"Como [tipo de usuario], quiero [funcionalidad] para [beneficio]"**  
  Ejemplo: "Como cliente, quiero pagar con tarjeta para no usar efectivo."
* Sirven para comunicación rápida y entendible entre usuarios y desarrolladores.

### **3. Requisitos generales (objetivos)**

* También llamados **requisitos de alto nivel** o **metas del sistema**.
* Describen **lo que se espera lograr** con el software, pero **sin entrar en detalles técnicos**.
* Son declaraciones amplias que guían el resto del proyecto.  
  Ejemplo: “Reducir el tiempo de atención al cliente en un 30%”.

### **4. Requisitos de información**

* Definen **qué datos necesita el sistema**: qué debe capturar, almacenar, procesar y mostrar.
* Pueden incluir:
  + Tipos de datos (nombre, edad, correo, etc.)
  + Origen del dato (usuario, otro sistema, etc.)
  + Relación entre datos
* Muy importantes en sistemas basados en bases de datos.

### **5. Reglas de negocio**

* Son **normas o restricciones propias del negocio o empresa** que afectan al sistema.
* No siempre son funcionales, pero **influyen en cómo debe comportarse** el software.
  + Ejemplo: “Un cliente no puede tener más de 3 préstamos activos.”
* Deben ser recogidas claramente para no violar políticas internas.

### **6. Requisitos funcionales**

* Describen **lo que el sistema debe hacer**.
* Son las **funcionalidades específicas** del sistema, por ejemplo:
  + Registrar un usuario
  + Calcular un total de compra
  + Generar un informe
* Se expresan normalmente como “El sistema debe [acción]...”

### **7. Requisitos no funcionales**

* No describen funciones directas, sino **características de calidad** del sistema.
* Ejemplos:
  + Rendimiento (responder en menos de 2 segundos)
  + Seguridad (encriptar contraseñas)
  + Usabilidad (interfaz intuitiva)
  + Escalabilidad, disponibilidad, compatibilidad, etc.
* A menudo son más difíciles de medir y probar.

### **8. Pruebas de aceptación**

* Son pruebas diseñadas para **verificar que el sistema cumple con los requisitos del usuario**.
* Se hacen al final del desarrollo (o por entregas en metodologías ágiles).
* Basadas en **casos de uso o historias de usuario**.
* Si las pasa, el cliente acepta el sistema.

### **9. Matrices de trazabilidad**

* Herramienta que relaciona **requisitos con sus correspondientes artefactos**: casos de uso, pruebas, código, etc.
* Sirve para:
  + Ver si todos los requisitos están cubiertos
  + Detectar qué se ve afectado si un requisito cambia
* Se representa en forma de **tabla** (matriz).
  + Ejemplo: Requisito R1 está cubierto por los Casos de uso CU1 y CU3, y validado por la prueba T2.