# Especificación de Requisitos de Software (ERS)

Es una descripción de un sistema de software a desarrollar. Dicha descripción abarca, en condiciones ideales, absolutamente todos los elementos necesarios para que un proyecto sea desarrollado. Esos elementos pueden ser entendidos como:

* **Requisitos funcionales**: (a veces llamados “casos de uso”) Están formados por todo el conjunto de entradas y salidas de información; así como el comportamiento que esa información tendrá dentro del sistema y la manera en que se comunicará con el entorno en que se encuentra.

Cuando se comprenden como casos de uso, indican la interacción que el sistema tendrá con partes externas. Estas terceras partes serán los *Actores* del sistema. Pueden ser usuarios humanos que generen o que accedan a la información o bien, otros sistemas que se comuniquen por medio de interfaces conjuntas.

* **Requisitos no-funcionales**: (a veces llamados “Cualidades del sistema”) Son los criterios que se utilizan para monitorear o medir la operación correcta del sistema, pero no su comportamiento como tal. Al contrario de los requisitos funcionales, que detallan el diseño del sistema; los no-funcionales detallan su arquitectura.

La arquitectura hace referencia a lo que el sistema debe ser, no a lo que debe hacer. Y, para especificar eso, se detalla el entorno bajo el que se trabajará y la amplitud que el sistema debe abarcar.

Una ERS será utilizada tanto por la parte que solicita la creación del sistema como por la parte que pondrá en marcha la creación del proyecto. Por esta razón, es muy importante que el lenguaje y formato que se utilicen en ella sea fácil de comprender para ambas partes.

Hay muchas recomendaciones y descripciones estandarizadas sobre las partes que componen una ERS. Sin embargo, el Instituto de Ingeniería Eléctrica y Electrónica (IEEE, por sus siglas en inglés), pese a ser la mayor entidad sobre estándares de sistemas, reconoce que las partes que componen una ERS varían en base a todos los aspectos relativos al sistema.

Se suelen recomendar que una ERS cubra los siguientes aspectos:

* **Funcionalidad**: Lo que el sistema debe hacer.
* **Interfaces externas**: Cómo interactuará con los usuarios, con el hardware del sistema y con hardware y software de terceros.
* **Rendimiento**: La velocidad, disponibilidad, tiempo de respuesta, tiempo de recuperación y tiempo requerido para ejecutar las funciones que se solicitan.
* **Atributos**: Indica la forma en que el sistema podrá ser mantenido, corregido y portado. También se incluyen atributos relativos a la seguridad para saber cómo puede ser protegido.
* **Restricciones de diseño en base a la implementación**: Sobre los estándares que necesitan ser respetados, las políticas que serán aplicadas, el lenguaje necesario para desarrollar, las limitaciones de recursos y el ambiente operativo. Son las condiciones “en contra” para desarrollar el sistema.

# Características de una ERS

* **Es correcta**: Además de estar absolutamente bien escrita y detallada desde el inicio, debe poder ser corregida en caso de haber errores o malos entendidos. Nunca puede seguirse una ERS si no es correcta en su descripción. De igual forma, el desarrollo no debe basarse en ella si no se cumple esta característica.
* **No es ambigua:** Esta característica indica que cada requerimiento que es enunciado en la ERS tiene una única interpretación.
* **Es completa**: Debe contener toda la información necesaria para que los desarrolladores del sistema inicien el proyecto.
* **Es consistente**: Toda la terminología, modos de lenguaje y tecnicismos deben ser exactamente iguales en la ERS y en los documentos a los que hace referencia.
* **Considera las prioridades**: Se deben poder desestimar los requerimientos que no sean absolutamente necesarios para favorecer a aquellos que son esenciales en el sistema. También se debe proveer información sobre porqué algunos requerimientos se consideran más importantes que otros.
* **Se puede medir**: Debe existir un método realista y objetivo para comprobar que los requerimientos se cumplen y que los enunciados que los proponen se acomodan a dicha medición.
* **Se puede modificar**: Los requerimientos pueden ser modificados o añadidos según se considere necesario por cualquiera de las partes que participan en ella.
* **Está identificado**: Todos los requerimientos y definiciones tienen una identificación única a través de la ERS. Dicha identificación debe seguir un patrón o nomenclatura que permita establecer la forma en la que se relaciona con el resto de ERS.

# Ventajas de elabora una ERS

* Establece las bases para llegar a un buen acuerdo entre los clientes y el analista respecto a lo que el sistema es y lo que se supone que haga.
* Reduce los costes de desarrollo porque permite desestimar los requerimientos que no son necesarios. También se evitan malos entendidos dentro del equipo del desarrollo y, al igual que la individualidad de los requerimientos, el trabajo de los desarrolladores resulta único.
* Proporciona bases para estimar costos y tiempos de entrega realistas. Incluso es posible que los requerimientos cambien si se encuentra que el sistema especificado excede los límites establecidos.
* Sirve como base para mejoras futuras al sistema. Algunos requerimientos pueden ser desestimados en el inicio de la creación y quedar en la “lista de deseos”. La ERS brinda acceso fácil a esa lista y permite, en base a los requerimientos ya cumplidos, establecer nuevos que puedan ser implementados con mayor facilidad.
* Funciona como línea base para validación y verificación del producto. Si el sistema implementado cumple con todo lo especificado en la ERS, significa que el sistema es y hace exactamente lo que el cliente necesita que haga. Y ello se puede comprobar comparando uno a uno todos los requerimientos establecidos.