

4.1	Definición de Roles
ROL	
Project Manager	La función de este rol es el cumplimiento de las expectativas de los resultados esperados por los servicios administrativos, definir las especificaciones.
Analyst	Este rol se encarga de hacer un análisis del proyecto que sea detallado de la realidad de las distintas características.
Designer	Diseñar las interfaces gráficas de datos Reserve System.
Development	Este rol debe llevar al término, decir, debe realizar la implementación del producto tanto en arquitectura.
Test	Este rol se encarga de validar los servicios entregados y validar el momento en que comienza.
User Education	Es la función que se encarga de la calidad de la documentación del proyecto.

# ADSI:

## ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

### FASE IDENTIFICACIÓN

4.2	Responsabilidades
CODIGO	
INF-1	El sistema tendrá...
Project Management	

#### 5.4 REQUISITOS NO FUNCIONALES

# INGENIERÍA DE REQUISITOS

## REQUISITOS - CLASIFICACIÓN - INGENIERÍA

### ACTIVIDAD DE PROYECTO

1. Determinar las especificaciones funcionales del Sistema de Información.

### ACTIVIDAD DE APRENDIZAJE

3. Elaborar el Documento Técnico de Identificación de necesidades del Sistema a desarrollar (Documento SRS)



De clase mundial

- 3.1 Resumen de Stakeholders
- 3.2 Resumen de Usuarios
- 3.3 Entorno de usuario
- 3.4 Perfil de los Stakeholders
  - 3.4.1 Coordinadora
- 3.5 Perfiles de Usuario
  - 3.5.1 Auxiliar administrativo
  - 3.5.2 Auxiliar administrativo
- Descripción Global de Producto
  - 5.1 Perspectiva del producto
  - 5.2 Resumen de características
  - 5.3 Suposiciones y dependencias
- Descripción Global del Producto
  - 6.1 Departamento de coordinación
  - 6.2 Departamento de apoyo
- Restricciones
- Riesgos
  - 8.1 Riesgo Tecnológico
  - 8.2 Riesgo Político
  - 8.3 Riesgo de recurso
  - 8.4 Riesgo Humano
- Requisitos de Documentación
  - 9.1 Manual de Usuario
  - 9.2 Ayuda en Línea
  - 9.3 Guía de instalación, Configuración, Ejecución





Creative  
Commos



Este material puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

<div><div>REQUISITOS</div><div>Concepto</div><div>¿Cómo deben ser?</div><div>¿Qué deben indicar?</div><div>¿Cómo obtener Requisitos?</div><div>Problemas comunes al obtener</div></div> <div>4</div>	<div><div>CLASIFICACIÓN</div><div>Requisitos No Funcionales</div><div>Requisitos Funcionales</div><div>Aspectos a tener en cuenta al describir requisitos</div><div>Forma de Presentación de los Requisitos</div></div> <div>8</div>	<div><div>INGENIERÍA</div><div>Participantes en el Proceso de Requisitos</div></div> <div>12</div>	<div><div>Glosario</div><div>Referencias</div></div> <div>14</div>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------	--------------------------------------------------------------------



# REQUISITOS

## REQUI- SITOS

El análisis de requisitos es una de las tareas más importantes en el ciclo de vida del desarrollo de software, puesto que en ella se determinan los “planos” de la nueva aplicación.

En cualquier proyecto software los requisitos son las necesidades del producto que se debe desarrollar.”  
(*Monferrer, 2001, p1*)

Es por esto que para realizar un buen análisis de los requisitos, se deben identificar claramente estas necesidades y documentarlas. Como artefacto se debe producir y entregar un documento de especificación de requisitos en el que se describa lo que el futuro sistema debe hacer.

El análisis de requisitos se puede definir como el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, hardware o software, así como el proceso de estudio y refinamiento de dichos requisitos, definición proporcionada por el IEEE [*Piattini, 1996*]. Asimismo, se define requisito como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado [*Piattini, 1996*]. Esta definición se extiende y se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación.

### *Monferrer (2001) comenta:*

En la determinación de los requisitos no sólo deben actuar los analistas, es muy importante la participación de los propios usuarios, porque son éstos los que mejor conocen el sistema que se va a automatizar. Analista y cliente se deben poner de acuerdo en las necesidades del nuevo sistema, ya que el cliente no suele entender el proceso de diseño y desarrollo del software como para redactar una especificación de requisitos software (ERS) y los analistas no suelen entender completamente el problema del cliente, debido a que no dominan su área de trabajo.

## CONCEPTO

- se suelen especificar en lenguaje natural,
- se expresan de forma individual (p.ej. esquemáticamente).
- se organizan de forma jerárquica (a distintos niveles de detalle).
- a menudo, se numeran (para facilitar su gestión).

## ¿CÓMO DEBEN SER?

Deben ser:

- **Claros y concretos:** (evitando imprecisiones y ambigüedades)  
p.ej. Uso de puntos suspensivos, etcétera...
- **Concisos:** (sin rodeos ni figuras retóricas).
- **Completo y consistentes.**

## ¿CÓMO OBTENER REQUISITOS?

- Revisar las necesidades de los clientes, usuarios y otros interesados.
- Revisar la situación actual.
- Revisar la organización actual.
- Conocer la versión actual del sistema.
- Entrevistar desarrolladores de versiones anteriores.
- Revisar documentos existentes (antecedentes).
- Revisar sistemas análogos (antecedentes).
- Se debe trabajar en conjunto con los usuarios y clientes.

## ¿QUÉ DEBEN INDICAR?

- Lo que se espera que haga el sistema (¿qué?).
- Su **justificación** (¿por qué ha de ser así? ¿quién lo propuso?).
- Los **criterios de aceptación** que sean aplicables (¿cómo se verifica su cumplimiento?).

## PROBLEMAS COMUNES AL OBTENER REQUISITOS

- Distintos usuarios tienen distintos requisitos, se deben encontrar todas las fuentes.
- No saben lo que quieren del sistema, sólo en términos generales, no conocen el costo de sus peticiones.
- Los requisitos están en sus términos y con conocimiento implícito de su propio trabajo.
- La prioridad que se da a los requisitos varía con el tiempo.
- Aparecen nuevos requisitos.
- Un requerimiento es, a veces, difícil de verificar (especialmente, si es un requisito no funcional).
- Además, si somos incapaces de especificarlo, ¿cómo sabemos que realmente es un requisito?
- La existencia de un requerimiento ha de estar debidamente justificada (debemos saber por qué es un requisito del sistema).

# REQUISITOS FUNCIONALES

- Expresan la naturaleza del sistema (como interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento).
- Servicios o funciones que proveerá el sistema
- Describen la interacción entre el sistema y el entorno.

“Definen el **qué debe** hacer un sistema”

## Ejemplos:

- Se debe solicitar la identificación, nombres, apellidos, genero, correo electrónico.
- Debe generar un listado de todas las personas de acuerdo al genero

- Deben estar redactados de tal forma que sean comprensibles para usuarios sin conocimientos técnicos avanzados (de informática, desarrollo de software).
- Deben especificar el comportamiento externo del sistema y evitar, en la medida de lo posible, establecer características de su diseño.
- Deben priorizarse (al menos, se ha de distinguir entre requisitos obligatorios y requisitos deseables).

# REQUISITOS NO FUNCIONALES

- Restricciones a los servicios o funciones ofrecidos por el sistema.
- Describen restricciones que limitan las elecciones para construir una solución.

“Definen el **cómo debe** hacer un sistema”

## Ejemplos:

- El lenguaje de programación debe ser java.
- El tiempo de respuesta en las consultas no debe superar los 5 segundos.

Deben especificarse cuantitativamente, siempre que sea posible (para que se pueda verificar su cumplimiento).

### • Rendimiento del sistema:

Fiabilidad, tiempo de respuesta, disponibilidad...

### • Interfaces:

Dispositivos de E/S, usabilidad, interoperabilidad...

### • Proceso de desarrollo:

Estándares, herramientas, plazo de entrega.

### • Del Producto:

Especifican restricciones al comportamiento del producto.

Ejemplos: desempeño, confiabilidad, portabilidad, usabilidad.

### • De la Organización:

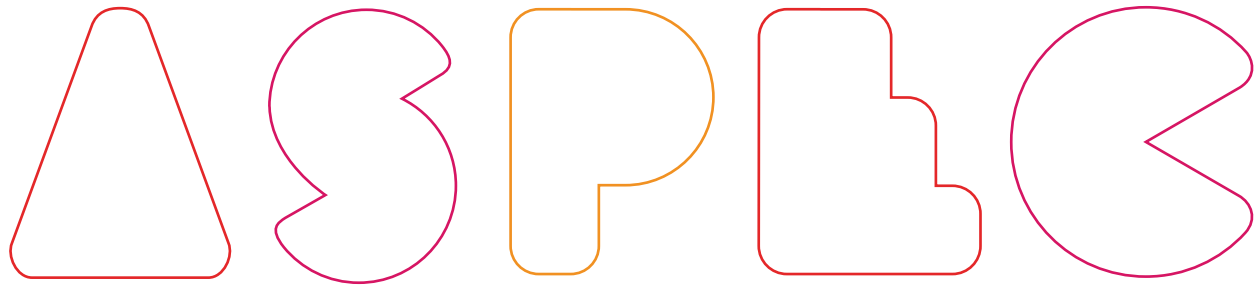
Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador.

Ejemplos: estándares, lenguajes de programación, método de diseño

### • Externos:

Se derivan de factores externos, como:

- Interoperabilidad: con otros sistemas.
- Legislativos: privacidad, seguridad.
- *Éticos*: dependen del contexto, las personas, etc

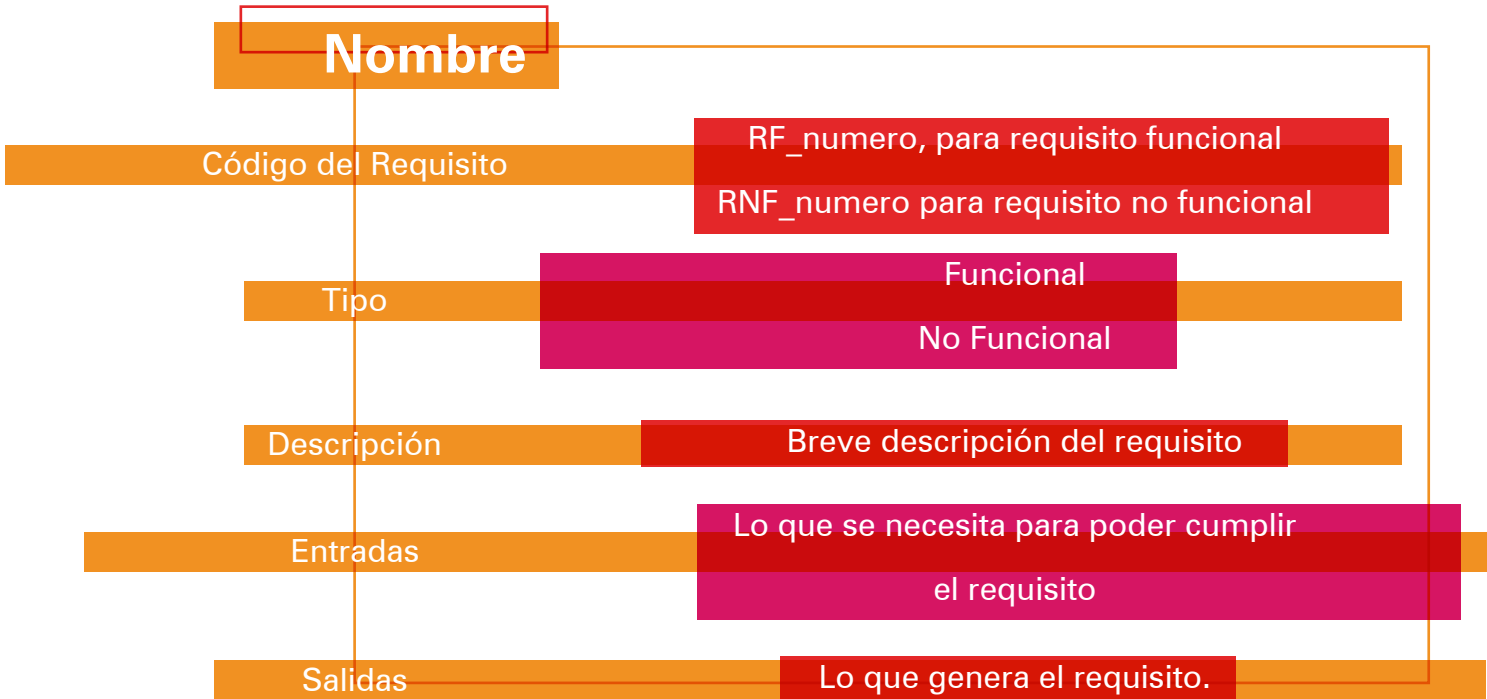


# ASPECTOS A TENER EN CUENTA AL DESCRIBIR REQUISITOS

Análisis y desarrollo de sistemas de información

- Ubicación y Entorno Físicos:** dónde, uno o varios, restricciones ambientales.
  - Funcionalidad y Restricciones asociadas:** qué debe hacer, cuándo, modos de operación, cómo y cuándo se puede modificar el sistema, restricciones de velocidad, tiempo de respuesta, capacidad de proceso.
  - Recursos:** materiales, personal y otros para construir, usar y mantener el sistema, habilidades de los desarrolladores, necesidades de espacio y ambientales, calendario prescrito, limitaciones en presupuesto.
  - Confiabilidad:** tiempo medio entre fallas, robustez, tolerancia a fallas.
- Interfaces:** Entrada de 1 o + sistemas, Salida a 1 o + sistemas, restricciones de formato, soporte.
  - Documentación:** cuánta, formato, para quién.
  - Seguridad:** control de acceso a las funciones/ datos, aislamiento de los programas, respaldos-frecuencia, disponibilidad-, seguridad física.
  - Disponibilidad:** tiempo para estar operativo luego de falla- mantenimiento estando activo- tiempo máximo de no disponibilidad.
- Usuarios y Factores Humanos:** capacidad de cada tipo de usuario, tipo de entrenamiento, facilidad de uso, posibilidad de mal uso.
  - Datos:** formatos E/S, frecuencia, fuentes, destinos, calidad requerida, precisión en cálculos, flujo en el sistema.
  - Aseguramiento de la Calidad**
  - Mantenibilidad.**
  - Seguridad.**
  - Portabilidad**

# FORMA DE PRESENTACIÓN DE LOS REQUISITOS



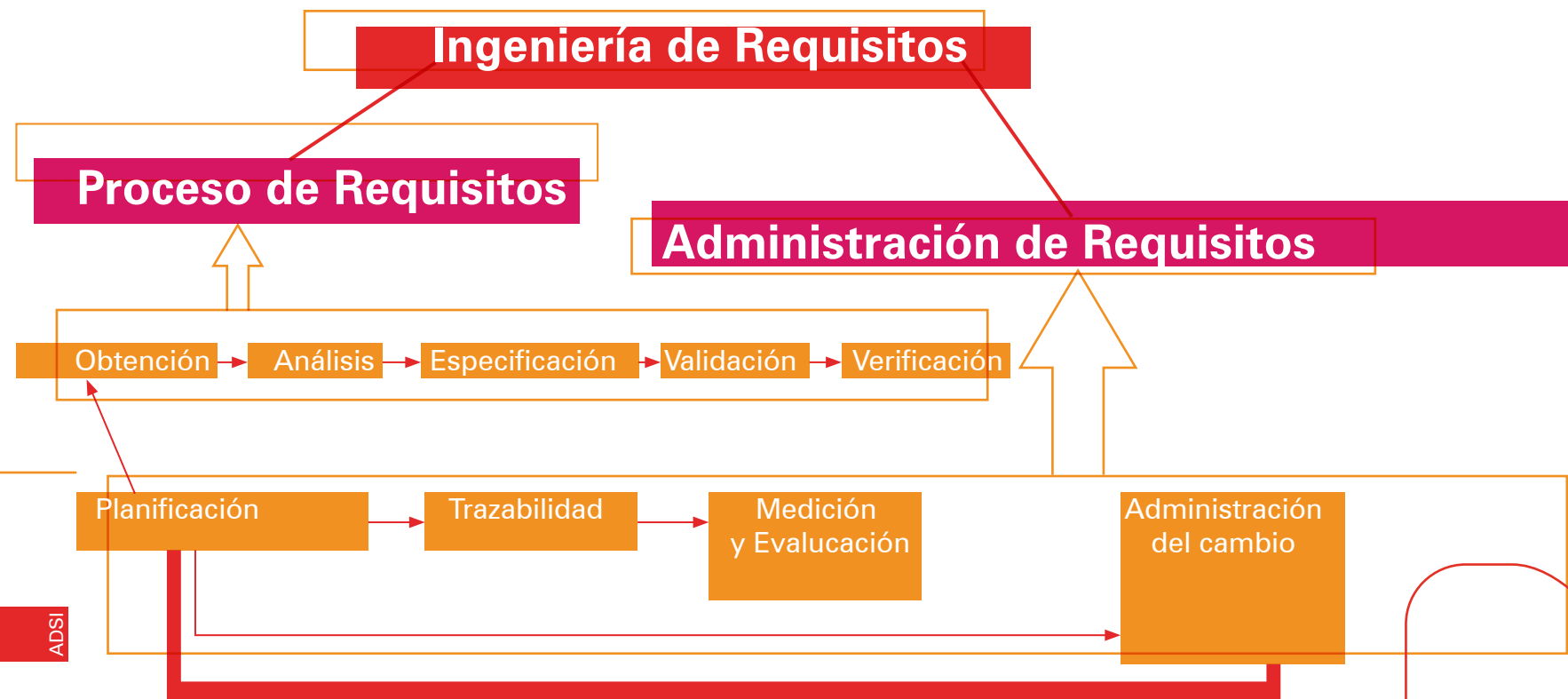
## Ejemplo Generación de reportes:

Código del Requisito	RF1
Tipo	Funcional
Descripción	El sistema deberá generar los reportes de todos los alumnos de una institución.
Entradas	Pinformación de alumnos por curso.
Salidas	El reporte de alumnos

SENA, DE CLASE MUNDIAL

ADSI

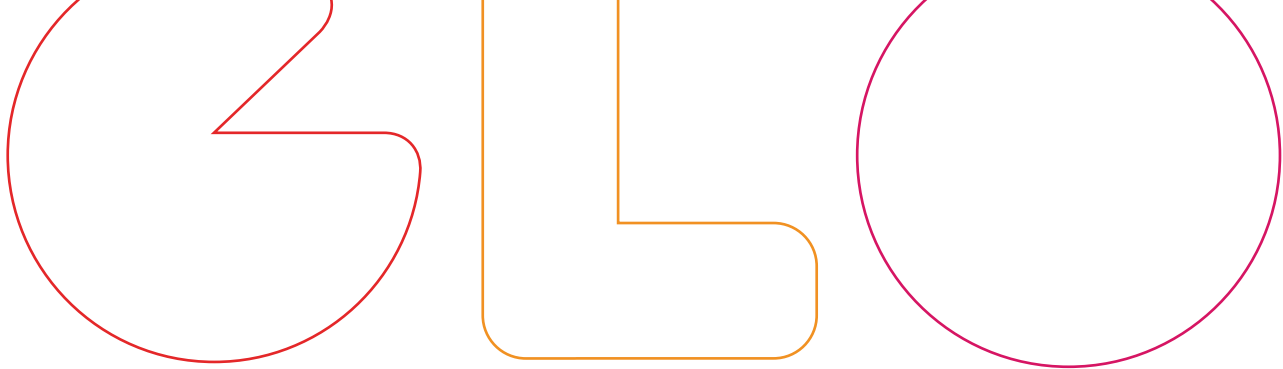
# INGENIERIA



## PARTICIPANTES EN EL PROCESO DE REQUISITOS

- **Cliente y Usuarios:** Requisitos adecuados a sus necesidades.
- **Diseñadores:** para lograr diseño que satisfaga las necesidades.
- **Supervisores del Contrato:** Hitos de Control, cronogramas.
- **Gerentes del Negocio:** Impacto en la Organización.
- **Verificadores:** para poder verificar si el sistema los satisface.





# GLOSARIO

**Artefacto de software:** (*software artefact*) Cualquier cosa que resulte del proceso de desarrollo de software; por ejemplo: documentos de requisitos, especificaciones, diseños, software, etc.

**Calidad:** (*quality*, ISO 8402, 1994) Conjunto de propiedades y de características de un producto o servicio, que le confieren su aptitud para satisfacer unas necesidades explícitas e implícitas. (*The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs*).

**Especificación:** [Piattini, 96]. Es un documento que define, de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento u otras características de un sistema o componente de un sistema.

**Fiabilidad:** (*reliability*,ISO 9126) Grado en que el sistema responde bajo las condiciones definidas durante un intervalo de tiempo dado. Se divide en las subcaracterísticas madurez, tolerancia a fallos, capacidad de recuperación.

**Funcionalidad:** (*functionality*, ISO 9126) Grado en que las necesidades asumidas o descritas se satisfacen. Se divide en las subcaracterísticas idoneidad, precisión, interoperabilidad, seguridad.

**IEEE:** (*Institute of Electrical and Electronics Engineers*). Asociación de profesionales norteamericanos que aporta criterios de estandarización de dispositivos eléctricos y electrónicos.

**Ingeniería de Requisitos:** Proceso de descubrir, analizar, documentar y verificar los servicios y sus restricciones.

**Interoperabilidad:** (*interoperability*,ISO 9126) Subcaracterística de funcionalidad, que indica el grado en que el sistema puede interactuar con otros sistema.

**Operabilidad:** (*operability*, ISO 9126) Subcaracterística de facilidad de uso, que indica las características del software que influyen en el esfuerzo del usuario para operar y control operacional.

**Portabilidad:** (*portability*, ISO 9126) Conjunto de características que determinan la capacidad del software para ser transferido de un entorno de operación a otro. Se divide en las subcaracterísticas adaptabilidad, facilidad de instalación, coexistencia, reemplazo.

**Precisión:** (*suitability*, ISO 9126) Subcaracterística de funcionalidad, que indica el grado de exactitud de los efectos del sistema (i.e. salida).

**Requerimientos:** son las necesidades que provienen del Negocio (Usuarios). Se plasman en el documento de requerimientos del negocio.

**Requisitos:** son las especificaciones puntuales sobre los servicios que debe ofrecer el sistema software y sus restricciones. Se plasman en el documento de especificación de requerimientos de software (SRS)

**Requisitos del sistema:** Son los requisitos para todo el sistema.

**Requisitos del software:** [SOMMERVILLE, 2002] Es la descripción de los servicios y restricciones de un sistema de software, es decir, lo que el software debe hacer y bajo qué circunstancias debe hacerlo.

**Seguridad:** (*security*, ISO 9126) Subcaracterística de funcionalidad, que indica el grado en que un acceso no autorizado (accidental o deliberado) se prevenga y se permita un acceso autorizado.

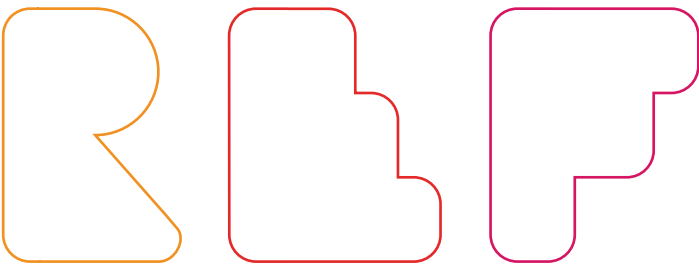
**Sistema:** Pensando en la solución, se puede definir como aquella que incluye *hardware, software, firmware, personas, información, técnicas, servicios, y otros elementos de soporte*.

**Stakeholder:** (*stakeholder*) Cualquier persona interesada en, afectada por y/o implicada con el funcionamiento del sistema software. Por ejemplo, el usuario, el cliente, nuestra empresa, etc.

•Monferrer, (2000-2001). E78. Ingeniería del Software. Universitat Jaume I, Departament d’Informàtica 5º Curso de Ingeniería Informática

# REFERENCIAS

•Pressman, R. (2006). Ingeniería de Software: Un enfoque Práctico. VI Edición. McGrawHill.



•Sommerville I., (2005). Ingeniería del Software. Séptima edición, México DF, Editorial Pearson.

•Ingeniería de Requisitos: <http://www.slideshare.net/ssharLudena/ingeniera-de-requisitos>.

•Términos de Calidad: <http://squac.iti.upv.es/glosario-calidad/>



Representante	Maria Elena Espinal.
Descripción	Auxiliar administrativo.
Pro	Usuario administrador, Apoyo a la administración.
Responsabilidades	Presta apoyo a la administración de la sala de conectividad, gestión de registro, gestión de reservas, gestión de servicios.
Criterio de Éxito	[A definir por el cliente]
Nivel de participación	Manejo total del sistema.
Comentarios	Ninguno.

# /ADSI:

## 1. SECCIÓN DE REQUISITOS

Representante	Cruz Helena Toro.	5.1 APLICACIONES		
Descripción	Auxiliar administrativo.	NOMBRE	DESCRIPCIÓN	
Pro	Usuario administrador, Apoyo a la administración.	Aplicación web que permitirá administrar la conectividad del complejo central del SENA regional (de computadores y tutorías), para los diferentes tipos de convenio (invitado).		
Responsabilidades	Presta apoyo a la administración de la sala de conectividad, gestión de reservas, gestión de servicios.	Reserve system		
Criterio de Éxito	[A definir por el cliente]			
Nivel de participación	Manejo total del sistema.			
Comentarios	Ninguno.			
		5.2 PROCESOS PRINCIPALES		
		PROCESOS	PRIORIDAD	DESCRIPCIÓN
		Procesos de reserva	Alta	Gestionar los diferentes tipos de

**LÍDER DEL PROGRAMA ADSI**  
Vanessa Cristina Miranda Cano  
[vanessa24@misena.edu.co](mailto:vanessa24@misena.edu.co)

**ASESORÍA PEDAGÓGICA**  
Claudia Herrera Cifuentes  
[pipelore@yahoo.com](mailto:pipelore@yahoo.com)

**ILUSTRACIÓN PORTADA**  
Saúl Suaza  
[ssuaza@gmail.com](mailto:ssuaza@gmail.com)

**COMPILACIÓN Y PREPARACIÓN**  
César Marino Cuéllar Chacón  
Vanessa Cristina Miranda Cano

**LÍDER LÍNEA DE PRODUCCIÓN**  
Iliana Eneth Molina Cuarta  
[ilmocu@sena.edu.co](mailto:ilmocu@sena.edu.co)

**DIAGRAMACIÓN**  
Coproducción  
Línea de Producción - Regional Santander

**DISEÑO EDITORIAL Y PORTADA**  
Ricardo Burbano Martínez  
[ribuma@gmail.com](mailto:ribuma@gmail.com)

Ricardo Burbano Martínez  
[ribuma@gmail.com](mailto:ribuma@gmail.com)



Gestión de clientes	A	5	Gestionar los clientes de la sala de conectividad, permitiendo organizar y almacenando el estado de la configuración de los clientes que utilizan los servicios del aplicativo (equipos y tutorías). <i>Hablen solo de clientes, su información básica y tipo.</i>	
Movimiento de clientes	Resumen de Stakeholders			
	A	6	Registrar la entrada y salida de los clientes que utilizan el aplicativo, para así tener un control de todos los que concurren a la sala de conectividad.	
Seguridad	<b>Nombre</b>			
	Lina Marcela Gutiérrez	M	1	<b>Descripción</b>
Configuración	<b>Responsabilidades</b>			
	M	2	Configuración, permite registrar y llevar almacenada en base de datos la configuración de los equipos y tutorías.	
Equipo de desarrollo				
		Andrés Camilo Cano	Project manager-desarrollador.	Conducción del proyecto.
		Wilmar Arturo Álvarez	Analista-documentador.	Analiza, diseña, desarrolla, documenta, prueba y capacita el sistema de información Reserve System.
		Sergio Taborda	Diseñador.	Analiza, diseña, desarrolla, documenta, prueba y capacita el sistema de información reserve system.