



# INGENIERIA DE SOFTWARE II

# Ingeniería de Software II

---

- » Planes 2015, 2011, 2007 y 2003 Lic. en Sistemas
  - » Planes 2015, 2011, 2007 y 2003 Lic. en Informática
  - » Planes 2015, 2007 Analista Programador Universitario
- 
- » Correlativas:
    - Ingeniería de Software I
    - Prueba de Lecto-Comprensión y Traducción de Inglés (final)

# Ingeniería de Software II

---

## »Reglamento de cursada

La cursada constará de un proyecto que será desarrollado durante todo el semestre.

El proyecto será realizado por grupos, y cada grupo tendrá asignado un ayudante que será el tutor durante todo el desarrollo.

Las consultas del proyecto serán respondidas por el ayudante asignado, en los horarios establecidos.

Dicho proyecto contará con entregas parciales de documentos establecidos, presentados en fechas pautadas.

Una vez corregido el trabajo, cada alumno deberá presentarse a un coloquio para aprobar la cursada.

# Aprobación de la Materia

---

» El final de la materia se aprobará optando entre:

- Rendir dos exámenes teóricos durante la cursada y sacando 6 (seis) o mas, en cada uno (con un recuperatorio por examen) y presentándose a un coloquio en una mesa de final, o
- Rendir examen escrito en las mesas de final.

» Los alumnos podrán optar por la primer opción con las siguientes condiciones:

El alumno deberá contar con 80% asistencia a las teorías.

El alumno deberá presentarse a rendir los 2 parciales teóricos.

El alumno que apruebe la parte teórica deberá inscribirse y presentarse para un coloquio en una mesa de final en el término de NO más de 1 año de finalizada la cursada según el calendario académico, transcurrido el cual la aprobación NO tendrá más validez.

# Ingeniería de Software II

---

## »Horarios

### Teoría

*Miércoles de 8 hs a 11 hs (Aula 9)*

*Martes 14 hs a 17 hs (Aula 5)*

### Práctica

*Jueves de 11:00 a 13:40 (AULA 4)*

*Martes de 17:30 a 20:10 (AULA 7)*

*Viernes de 16:30 a 19:50 (AULA 11)*

# Entornos de comunicación

---

» Blog de la cátedra  
<http://blogs.unlp.edu.ar/ingenieria2/>

» Curso virtual de la cátedra

Ingeniería de Software II 2017

# Ingeniería de Software II

---

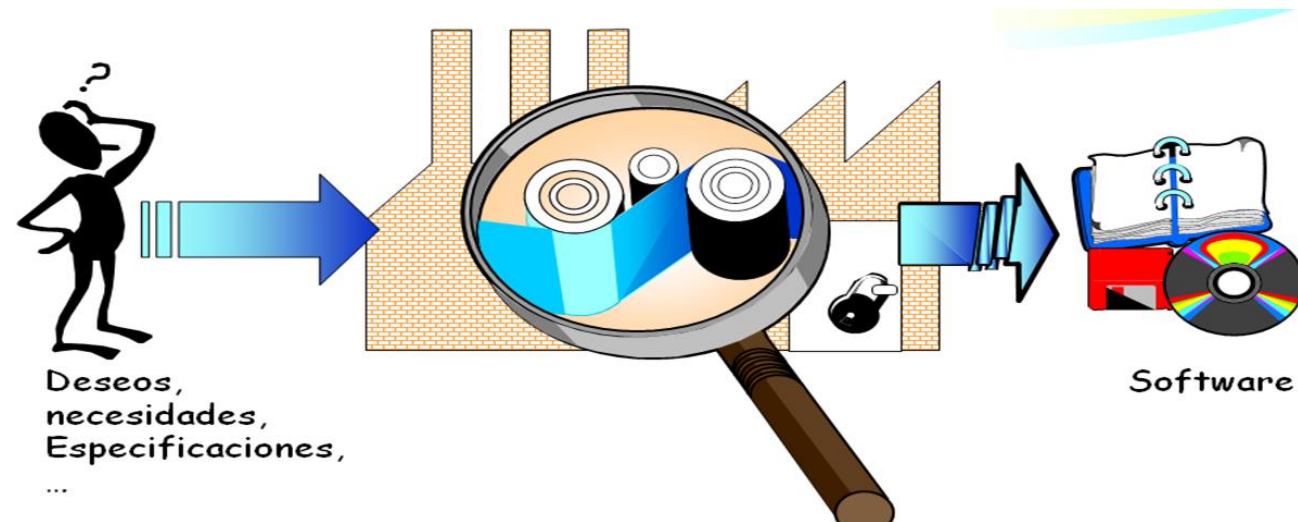
»Temas a desarrollar en la materia:

- 1- Gestión o Administración de Proyectos.
- 2- Diseño e Implementación de Software.
- 3- Verificación y Validación.
- 4- Mantenimiento de Software.
- 5- Gestión de Configuración.
- 6- Conceptos de Auditoría y Peritaje.



# ¿Qué es un proceso de software?

» Es un conjunto de actividades y resultados asociados que producen un producto de software.





# Ingeniería de Software II

---

Es una representación abstracta de un proceso del software.

## » Actividades fundamentales de los modelos de Proceso

Especificación del software

*Técnicas de elicitación*

*Especificación de requerimientos*

Desarrollo del software

Validación del software

Evolución del software

Ingeniería de Software I

# El problema de la comunicación

---

- » Les proponemos ver el video y analizar los inconvenientes que se producen y qué sería necesario para poder resolverlo:
- » Requerimientos
- » <https://www.youtube.com/watch?v=93SgXeu-SeY>

# El problema de la comunicación y los requisitos



# Elicitación de Requisitos

---

# ¿Qué es la elicitación de requisitos?

---

- » Les proponemos ver el siguiente video, y anotar los puntos principales que definen el proceso de elicitación y agregar lo que hayamos visto en Ingeniería de software 1.
- » El video es una entrevista que le hacen en Madrid en 2014 a Jordi Borja Sanz, en un seminario de requisitos de MTP que es el director de soluciones tecnológicas de MTP.
- » MTP es una consultora especializada en servicios y
- » calidad del software
  
- » <https://www.youtube.com/watch?v=wOmGTPBAJrM>

# Elicitación de Requisitos

---

- » Es el proceso de adquirir (“eliciting”) [sonsacar] todo el conocimiento relevante necesario para producir un modelo de los requerimientos de un dominio de problema
- » Objetivos:
  - Conocer el dominio del problema para poder comunicarse con clientes y usuarios y entender sus necesidades.
  - Conocer el sistema actual (manual o informatizado).
  - Identificar las necesidades, tanto explícitas como implícitas, de clientes y usuarios y sus expectativas sobre el sistema a desarrollar.

# Técnicas de elicitación

---

- » Muestreo de la documentación, los formularios y los datos existentes
- » Investigación y visitas al lugar
- » Observación del ambiente de trabajo
- » Cuestionarios
- » Entrevistas
- » Planeación conjunta de requerimientos (JRP o JAD)
- » Lluvia de ideas (Brainstorming)



# Muestreo de la documentación, los formularios y los datos existentes

---

- »Recolección de hechos a partir de la documentación existente
  - ¿Qué tipo de documentos pueden enseñar algo acerca del sistema?
    - Organigrama (identificar el propietario, usuarios claves)*
    - Memos, notas internas, minutas, registros contables*
    - Solicitudes de proyectos de sistemas de información anteriores*
  - Permiten conocer el historial que origina el proyecto

# Muestreo de la documentación, los formularios y los datos existentes

---

Documentos que describen la funcionalidad del negocio que está siendo analizada

*Declaración de la misión y plan estratégico de la organización*

*Objetivos formales del departamento en cuestión*

*Políticas, restricciones, procedimientos operativos*

*Formularios de operaciones realizadas*

*Bases de Datos*

*Sistemas en funcionamiento*

# Muestreo de la documentación, los formularios y los datos existentes

---

Documentación de sistemas anteriores

*Diagramas*

*Diccionario o Repositorios de proyecto*

*Documentos de diseño*

*Manuales de operación y/o entrenamiento*

» Técnicas de muestreo de documentos y archivos

# Investigación y visitas al sitio

---

- » Investigar el dominio
- » Patrones de soluciones (mismo problema en otra organización)
- » Revistas especializadas
- » Buscar problemas similares en internet
- » Consultar otras organizaciones

# Observación del ambiente de trabajo

---

- » El analista se convierte en observador de las personas y actividades con el objeto de aprender acerca del sistema.
- » Lineamientos de la observación:
  - Determinar quién y cuándo será observado
  - Obtener el permiso de la persona y explicar el porqué será observado
  - Mantener bajo perfil
  - Tomar nota de lo observado
  - Revisar las notas con la persona apropiada
  - No interrumpir a la persona en su trabajo

# Cuestionarios

---

» Documento que permite al analista recabar información y opiniones de los encuestados

- Recolectar hechos de un gran número de personas

- Detectar un seguimiento generalizado

- Detectar problemas entre usuarios

- Cuantificar respuestas

# Cuestionarios

---

- » ¿Permiten una respuesta rápida? ¿Por qué?
- » ¿Por qué son fáciles de analizar?
- » ¿Qué estrategia se puede utilizar para acelerar el proceso de creación de los cuestionarios?
- » ¿Qué inconvenientes encuentra en la puesta en práctica del cuestionario?
- » ¿Qué tipos de cuestionarios conoce y cuál es el objetivo de cada uno?



# Entrevistas

---

- » Técnica de exploración mediante la cual el analista de sistemas recolecta información de las personas a través de la interacción cara a cara
- » Es una conversación con un propósito específico, que se basa en un formato de preguntas y respuestas en general
- » Conocer opiniones y sentimientos del entrevistado

# Entrevistas

---

## »Debe

- Vestirse adecuadamente
- Ser cortés
- Escuchar cuidadosamente
- Mantener el control
- Observar los gestos
- Ser paciente
- Mantener al entrevistado en calma
- Mantener el autocontrol
- Terminar a tiempo

## »Evite

- Suponer que una respuesta no lleva a ningún lado
- Revelar pistas
- Usar jerga
- Revelar sesgos personales
- Hablar en lugar de escuchar
- Suponer cualquier cosa acerca del tema o del entrevistado
- Uso de grabadores (señal de debilidad de escuchar)

# Entrevistas

| Tipos de Entrevistas |                                 |          |
|----------------------|---------------------------------|----------|
| Abierta              |                                 | Cerrada  |
| Difícil              | Evaluación                      | Fácil    |
| Mucha                | Cantidad requerida de tiempo    | Poco     |
| Muy Necesario        | Entrenamiento requerido         | Limitado |
| Mucho                | Permite la espontaneidad        | Poco     |
| Muchas Oportunidades | Permite conocer al entrevistado | Muy Poco |
| Gran                 | Flexibilidad                    | Reducida |
| Bajo                 | Control de la Entrevista        | Alto     |
| Baja                 | Precisión                       | Alta     |
| Baja                 | Confiabilidad                   | Alta     |
| Mucha                | Amplitud y Profundidad          | Poca     |

# Planeación Conjunta de Requerimiento (JRP)

---

- »Proceso mediante el cual se conducen reuniones de grupo altamente estructurados con el propósito de analizar problemas y definir requerimientos
- »Requiere de extenso entrenamiento
- »Reduce el tiempo de exploración de requisitos
- »Amplia participación de los integrantes
- »Se trabaja sobre lo que se va generando
- »Alguna bibliografía la menciona como JAD (Joint Application Design)

# Lluvia De Ideas (Brainstorming)

---

- » Técnica para generar ideas al alentar a los participantes para que ofrezcan tantas ideas como sea posible en un corto tiempo sin ningún análisis hasta que se hayan agotado las ideas.
- » Se promueve el desarrollo de ideas creativas para obtener soluciones.
- » Se realizan reuniones del equipo involucrado en la resolución del problema, conducidas por un director.

# Lluvia De Ideas (Brainstorming)

---

- » Incluye una serie de fases de aplicación:  
Descubrir hechos, Producir ideas, Descubrir soluciones
- » Clave para resolver la falta de consenso entre usuarios
- » Es útil combinarlo con la toma de decisiones
- » Ayuda a entender el dominio del problema
- » Encara la dificultad del usuario para transmitir
- » Ayuda a entender: al usuario y al analista

# Requerimientos

---



# Requerimientos

Solicitud



Definición



Especificación



Análisis



Facultad de  
Informática



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Requerimientos

---

## » Características

Necesario: Su omisión provoca una deficiencia.

Conciso: Fácil de leer y entender

Completo: No necesita ampliarse

Consistente: No contradictorio con otro

No ambiguo: Tiene una sola implementación

Verificable: Puede testearse a través de inspecciones, pruebas, etc.

## » Dificultades para definir los requerimientos

No son obvios

Proviene de muchas fuentes

Están interrelacionados

Pueden ser muchos

Pueden cambiar a lo largo del desarrollo

Son particulares para cada proyecto

## » Participantes

Los clientes, usuarios, gerentes de negocio, supervisores de contrato, analistas, diseñadores, verificadores

# Defectos en los requerimientos

---

## » Estimación

Es muy difícil estimar los costos y recursos necesarios para desarrollar algo que no se conoce.

## » Planificación

No se puede confiar en la planificación para el desarrollo de algo que no se sabe bien como es.

## » Diseño

Los errores en requerimientos, las modificaciones frecuentes, las deficiencias en restricciones o futuras evoluciones, producen arquitecturas que más tarde se confirmarán como erróneas y serán modificadas

**Requerimientos**

Estimación

Planificación

Diseño

Construcción

V & V

Facultad de  
Informática



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Defectos en los requerimientos

---

## » Construcción

Las deficiencias en los requerimientos obligan a programar en ciclos de prueba y error que derrochan horas y paciencia de programación sobre patrones de “recodificación continua” y “programación heroica”.

## » Validación

Terminado el desarrollo del sistema, si las especificaciones tienen errores grandes o no están reflejadas en una especificación de requerimientos, no será posible validar el producto con el cliente.

**Requerimientos**

Estimación

Planificación

Diseño

Construcción

V & V

# Buenos requerimientos

---

» *Acuerdo entre desarrolladores, clientes y usuarios sobre el trabajo que debe realizarse.*

Requerimientos bien elaborados y validados con el cliente evitan descubrir al terminar el proyecto que el sistema no era lo que se pedía.

» *Acuerdo entre desarrolladores, clientes y usuarios sobre los criterios que se emplearán para su validación.*

Resulta muy difícil demostrar al cliente que el producto desarrollado hace lo que el pidió si su petición no está documentada y validada por él.

» *Base objetiva para la estimación de recursos (costo, personal en número y competencias, equipos y tiempo)*

Las estimaciones en el fondo son cálculos de probabilidad que siempre implican un margen de error; por esta razón disponer de la mayor información posible reduce el error.

# Buenos requerimientos

---

» *Concreción de los atributos de calidad (ergonomía, mantenibilidad, etc.)*

Más allá de funcionalidades precisas, los requerimientos recogen atributos de calidad necesarios que en ocasiones no son tenidos en cuenta por los desarrolladores, produciendo sistemas con serias deficiencias de rendimiento.

» *Eficiencia en el consumo de recursos: reducción de la re-codificación, reducción de omisiones y malentendidos.*

Tener un conocimiento preciso de lo que hay que hacer evita la prueba y error, repetición de partes ya desarrolladas, etc.

# Tipos de requerimientos

---

## »Requerimiento Funcionales

Definen el comportamiento del sistema.

Describen las tareas que el sistema debe realizar.

Al definir un requisito funcional es importante mantener el equilibrio entre la excesiva generalidad, y el exceso de detalle con descripciones innecesarias o redundantes.

## »Requerimiento No Funcionales

Definen aspectos, que sin ser funcionalidades, (tareas que el sistema debe realizar) resultan deseables desde el punto de vista del usuario. También se pueden ver como restricciones.

Tiempos de respuesta.

Características de usabilidad.

Facilidad de mantenimiento.



# Especificación de requerimientos

ConOps IEEE Std1362

## » Descripción del sistema

Documento, también denominado ConOps y normalizado en el estándar IEEE Std. 1362-1998.

Documento dirigido a los usuarios, que describe las características de un sistema propuesto, desde el punto de vista del usuario. La Descripción del Sistema es el medio de comunicación que recoge la visión general, cualitativa y cuantitativa de las características del sistema; compartido por la parte cliente y desarrolladora.

## » Requerimientos del Software

Documento, también denominado SRS (ERS) y normalizado en el estándar IEEE Std. 830-1998.

Un documento SRS es la especificación de las funciones que realiza un determinado producto de software, programa o conjunto de programas en un determinado entorno. El documento de especificación de requisitos

# Descripción del sistema - IEEE 1362

---

- » Ofrece un formato y contenidos para la confección de las descripciones de sistema en los desarrollos y modificaciones de sistemas.
- » El estándar no especifica técnicas exactas, sino que proporciona las líneas generales que deben respetarse. Es una guía de referencia.
- » Las partes esenciales de un ConOps son:
  - Punto 3: Descripción del sistema existente.
  - Punto 4: Descripción del sistema propuesto.
- » El estándar identifica los elementos que al menos debe incluir una Descripción del sistema. El usuario puede incorporar otros elementos, agregando cláusulas y sub-cláusulas.

# IEEE 1362

---

**IEEE Std 1362™-1998 (R2007)**

(Incorporates  
IEEE Std 1362a-1998)

## **IEEE Guide for Information Technology—System Definition— Concept of Operations (ConOps) Document**

# IEEE 1362

## Contents

|      |  |    |
|------|--|----|
| 1.   | Scope.....   | 1  |
| 2.   | References.....  | 1  |
| 3.   | Definitions.....   | 2  |
| 4.   | Elements of a ConOps document.....   | 4  |
| 4.1  | Scope (Clause 1 of the ConOps document).....                                   | 5  |
| 4.2  | Referenced documents (Clause 2 of the ConOps document).....                    | 6  |
| 4.3  | Current system or situation (Clause 3 of the ConOps document).....             | 7  |
| 4.4  | Justification for and nature of changes (Clause 4 of the ConOps document)..... | 9  |
| 4.5  | Concepts for the proposed system (Clause 5 of the ConOps document).....        | 11 |
| 4.6  | Operational scenarios (Clause 6 of the ConOps document).....                   | 14 |
| 4.7  | Summary of impacts (Clause 7 of the ConOps document).....                      | 15 |
| 4.8  | Analysis of the proposed system (Clause 8 of the ConOps document).....         | 16 |
| 4.9  | Notes (Clause 9 of the ConOps document).....                                   | 16 |
| 4.10 | Appendices (Appendices of the ConOps document).....                            | 16 |
| 4.11 | Glossary (Glossary of the ConOps document).....                                | 16 |
|      | Annex A (Informative) IEEE/EIA 12207.1-1997 Compliance Statement.....          | 17 |

Facultad de  
Informática



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Especificación de requerimientos

---

## » Descripción del sistema

Documento, también denominado ConOps y normalizado en el estándar IEEE Std. 1362-1998.

Documento dirigido a los usuarios, que describe las características de un sistema propuesto, desde el punto de vista del usuario. La Descripción del Sistema es el medio de comunicación que recoge la visión general, cualitativa y cuantitativa de las características del sistema; compartido por la parte cliente y desarrolladora.

## » Requerimientos del Software

Documento, también denominado SRS (ERS) y normalizado en el estándar IEEE Std. 830-1998.

Un documento SRS es la especificación de las funciones que realiza un determinado producto de software, programa o conjunto de programas en un determinado entorno. El documento de especificación de requisitos

SRS IEEE Std 830

Facultad de  
Informática



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Requerimientos del Software IEEE-830

---

## »SRS – IEEE 830

En las Especificaciones de Requerimientos de software, se debe evitar incluir requerimientos de diseño o de proyecto.

## »Los aspectos básicos que una descripción de requerimientos debe cubrir son:

Funcionalidad. Descripción de lo que el software debe hacer.

Interfaces externas. Cómo debe interactuar el software con las personas, el hardware, o con otros sistemas.

Rendimiento. Indicación de la velocidad, disponibilidad, tiempos de respuesta, tiempos de recuperación, tiempos de determinadas funciones.

Atributos. Consideraciones de portabilidad, corrección, mantenibilidad, seguridad, etc.

Restricciones de diseño en la implementación. Indicación de las restricciones que puedan afectar por la necesidad de sometimiento a estándares, lenguajes, políticas de integridad de bases de datos, límites de

# IEEE 830

---

**IEEE Std 830-1998**  
(Revision of  
IEEE Std 830-1993)

## **IEEE Recommended Practice for Software Requirements Specifications**

# IEEE 830- SRS

---

## » Resumen - Alcance

Brindar una colección de buenas prácticas para escribir especificaciones de requerimientos de software (SRS). Se describen los contenidos y las cualidades de una buena especificación de requerimientos.



# Consideraciones para un buen SRS

---

## » Naturaleza del SRS

El SRS es una especificación para un producto de software particular. El SRS es escrito por uno o mas representantes del equipo de desarrollo y uno o mas representantes de la parte cliente o ambos.

## » Ambiente del SRS

El software puede contener toda la funcionalidad del proyecto o puede ser parte de un sistema más grande. En el último caso habrá un SRS que declarará las interfaces entre el sistema y su software desarrollado, y pondrá qué función externa y requerimientos de funcionalidad tiene con el software desarrollado.

## » Características de un buen SRS

Correcto, no ambiguo, completo , consistente, priorizable, comprobable, modificable, trazable

# Consideraciones para un buen SRS

## Características de un buen SRS

---

### »Correcto

Un SRS es correcto si, y sólo si, cada requisito declarado se encuentra en el software.

### »No ambiguo

Un SRS es inequívoco si, y sólo si, cada requisito declarado tiene sólo una interpretación.

### »Completo

Un SRS está completo si, y sólo si, se reconoce cualquier requisito externo impuesto por una especificación del sistema.

### »Consistente

La consistencia se refiere a la consistencia interior. Si un SRS no está de acuerdo con algún documento del nivel superior, como una especificación de requerimientos de sistema, entonces no es consistente.

# Consideraciones para un buen SRS

## Características de un buen SRS

---

### »Priorizado

Un SRS es priorizado por la importancia de sus requerimientos particulares

### »Comprobable

Un SRS es comprobable si, y sólo si, cada requisito declarado es comprobable. Un requisito es comprobable si, y sólo si, existe algún proceso con que una persona o máquina puede verificar que el producto del software reúne el requisito. En general cualquier requisito ambiguo no es comprobable

### »Modificable

Un SRS es modificable si, y sólo si, su estructura y estilo son tales que puede hacerse cualquier cambio a los requerimientos fácilmente, completamente y de forma consistente mientras conserva la estructura y estilo

### »Trazabilidad

Claridad del origen de cada requerimiento y su trazabilidad hacia los

# Consideraciones para un buen SRS

---

## » Preparación conjunta del SRS

El SRS se debe preparar en conjunto con las partes intervinientes para lograr un buen acuerdo entre las partes

## » Evolución de SRS

El SRS debe evolucionar conjuntamente con el software, registrando los cambios, los responsables y aceptación de los mismos.

## » Prototipos

El uso de prototipos se utiliza frecuentemente para la definición de requerimientos

## » Diseño incorporado en el SRS

El SRS puede incorporar los atributos o funciones externos al sistema, en particular las que describen el diseño para interactuar entre los subsistemas.

# Partes de un SRS

|   |
|---|
| <b>Table of Contents</b>  |
| 1. Introduction   |
| 1.1 Purpose   |
| 1.2 Scope   |
| 1.3 Definitions, acronyms, and abbreviations  |
| 1.4 References  |
| 1.5 Overview  |
| 2. Overall description  |
| 2.1 Product perspective   |
| 2.2 Product functions   |
| 2.3 User characteristics  |
| 2.4 Constraints   |
| 2.5 Assumptions and dependencies  |
| 3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.) |
| Appendixes  |
| Index   |

**Figure 1 – Prototype SRS outline**

# Sección 1 del SRS

## 1 Introducción

---

### »1.1 Propósito

Se define el propósito del documento y se especifica a quién va dirigido el documento

### »1.2 Alcance o ámbito del sistema

Se da un nombre al futuro sistema

Se explica lo que el sistema hará y lo que no hará.

Se describen los beneficios, objetivos y metas que se espera alcanzar con el futuro sistema

### »1.3 Definiciones, siglas y abreviaciones

Glosario

### »1.4 Referencias

Esta subdivisión debe proporcionar una lista completa de todas las referencias de los documentos mencionados o utilizados para escribir el SRS. Identificar cada documento por el título, número de reporte, fecha y

# Sección 2 del SRS

## 2 Descripción General

---

»Esta sección del SRS debe describir los factores generales que afectan el producto y sus requerimientos. Esta sección no declara los requerimientos específicos. En cambio, mantiene una mención general de esos requerimientos que se definen en detalle en Sección 3 del SRS y los hacen más fáciles de entender.

»Esta sección normalmente consiste en seis subdivisiones, como sigue:

- Perspectiva del producto

- Funcionalidades del producto

- Características de los usuarios

- Restricciones

- Suposiciones y dependencias

- Evoluciones previsibles del sistema

# Sección 2 del SRS

## 2 Descripción General

---

### »2.1. Perspectiva del producto

Si el producto es independiente y totalmente autónomo, debe declararse que así es.

Si el SRS define un producto que es un componente de un sistema más grande entonces esta subdivisión debe relacionar los requerimientos de ese sistema más grande a la funcionalidad del software y debe identificar las interfaces entre ese sistema y el software.

### »2.2. Funciones del sistema

Se debe presentar un resumen, a grandes rasgos, de las funciones del futuro sistema.

Las funciones deberán mostrarse de forma organizada, y pueden utilizarse gráficos, siempre y cuando dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema.



# Sección 2 del SRS

## 2 Descripción General

---

### »2.4. Restricciones

a) Las interfaces: del Sistema, del Usuario, del Hardware; de las de Comunicaciones; b) Acceso y uso de la Memoria; c) Los requerimientos de adaptación del Sitio d) Políticas de la empresa e) Limitaciones del hardware f) Interfaces con otras aplicaciones g) Operaciones paralelas h) Funciones de auditoría i) Lenguaje(s) de programación. Bases de Datos. j) Protocolos de comunicación k) Req. de fiabilidad l) Consideraciones acerca de la seguridad

### »2.5. Suposiciones y dependencias

Se describen aquellos factores que, si cambian, pueden afectar a los requerimientos.

### »2.6 Evoluciones previsibles del sistema

Se identifican requerimientos que serán implementados en futuras

# Sección 3 del SRS

## 3 Requerimientos específicos

---

- » Debe contener todos los requerimientos del software a un nivel de detalle suficiente para permitirles a los diseñadores diseñar un sistema para satisfacer esos requerimientos, y a los auditores probar que el sistema satisface esos requerimientos.
- » A lo largo de esta sección, cada requisito declarado debe ser externamente perceptible por los usuarios, operadores u otros sistemas externos.

Requerimientos comunes de interfaces

*Descripción detallada de todas las entradas y salidas del sistema de software.*

Requerimientos funcionales

*Descripción de las funcionalidades de sistema*

Requerimientos no funcionales

*Descripción de los requerimientos no funcionales*

Otros requerimientos

# Sección 3 del SRS

## 3.1 Requerimientos comunes de las interfaces

---

### » 3.1.1 Interfaces de usuario

Describir los requerimientos del interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas del interfaz.

### » 3.1.2 Interfaces de hardware

Especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

### » 3.1.3 Interfaces de software

Indicar si hay que integrar el producto con otros productos de software.

Para cada producto de software debe especificarse lo siguiente: Descripción del producto software utilizado, Propósito del interfaz, Definición del interfaz.

### » 3.1.4 Interfaces de comunicación

Describir los requerimientos de interfaces de comunicación e inter-

# Sección 3 del SRS

## 3.2 Requerimientos funcionales

---

### »3.2.1 Requisito funcional 1

Objetivo, descripción, secuencia exacta de operaciones, respuesta a situaciones anormales, etc.

### »3.2.n Requisito funcional n

Objetivo, descripción, secuencia exacta de operaciones, respuesta a situaciones anormales, etc.

»Serán desarrollados utilizando Historias de Usuarios

# Sección 3 del SRS

## 3.3 Requerimientos no funcionales

---

### »3.3.1 Requerimientos de rendimiento

Especificación de los requerimientos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, etc.

Todos estos requerimientos deben ser mensurables. Por ejemplo, indicando “el 95% de las transacciones deben realizarse en menos de 1 segundo”, en lugar de “los operadores no deben esperar a que se complete la transacción”.

### »3.3.2 Seguridad

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requerimientos pueden especificar:

Ejemplos de técnicas criptográficas. Registro de ficheros con “logs” de

# Sección 3 del SRS

## 3.3 Requerimientos no funcionales

---

### »3.3.4 Disponibilidad

Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad.

### »3.3.5 Mantenibilidad

Identificación del tipo de mantenimiento necesario del sistema. Especificación de quién debe realizar las tareas de mantenimiento, por ejemplo usuarios, o un desarrollador. Especificación de cuándo deben realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanales y mensuales.

### »3.3.6 Portabilidad

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

Facultad de  
Informática



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Sección 3 del SRS

## 3.4 Otros Requerimientos y 4 Apéndice

---

»3.4 Cualquier otro requisito que no encaje en ninguna de las secciones anteriores.

Por ejemplo:

*requerimientos culturales y políticos*

*requerimientos legales*

»4 Apéndices

Pueden contener todo tipo de información relevante para la SRS pero que, propiamente, no forme parte de la SRS.

Por ejemplo:

*Casos de Uso*