



***Técnico en***  
**< DESARROLLO DE SOFTWARE >**

***Análisis de Requisitos del  
Sistema y del Software***

(CC BY-NC-ND 4.0)  
International

Attribution-NonCommercial-NoDerivatives 4.0



## **Atribución**

Usted debe reconocer el crédito de una obra de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace.



## **No Comercial**

Usted no puede hacer uso del material con fines comerciales.



## **Sin obra derivada**

Si usted mezcla, transforma o crea un nuevo material a partir de esta obra, no puede distribuir el material modificado.

No hay restricciones adicionales - Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



# *Análisis de Requisitos del Sistema y del Software*

## *Unidad II*

### *Especificación de requerimientos*

#### 1. Especificación de requerimientos

**La Especificación** es un documento que define, de forma completa, precisa y verificable, los requisitos, el diseño y el comportamiento u otras características, de un sistema o componente de un sistema.

Para realizar bien el desarrollo de software es esencial tener una especificación completa de los requerimientos. Independientemente de lo bien diseñado o codificado que esté, un sistema pobremente especificado decepcionará al usuario y hará fracasar el desarrollo. La forma de especificar tiene mucho que ver con la calidad de la solución. Los ingenieros de software que se han esforzado en trabajar con especificaciones incompletas, inconsistentes o mal establecidas han experimentado la frustración y confusión que invariablemente se produce. Como se ilustra en la *figura 3.1*, las consecuencias se padecen en la calidad, oportunidad y completitud del software resultante.

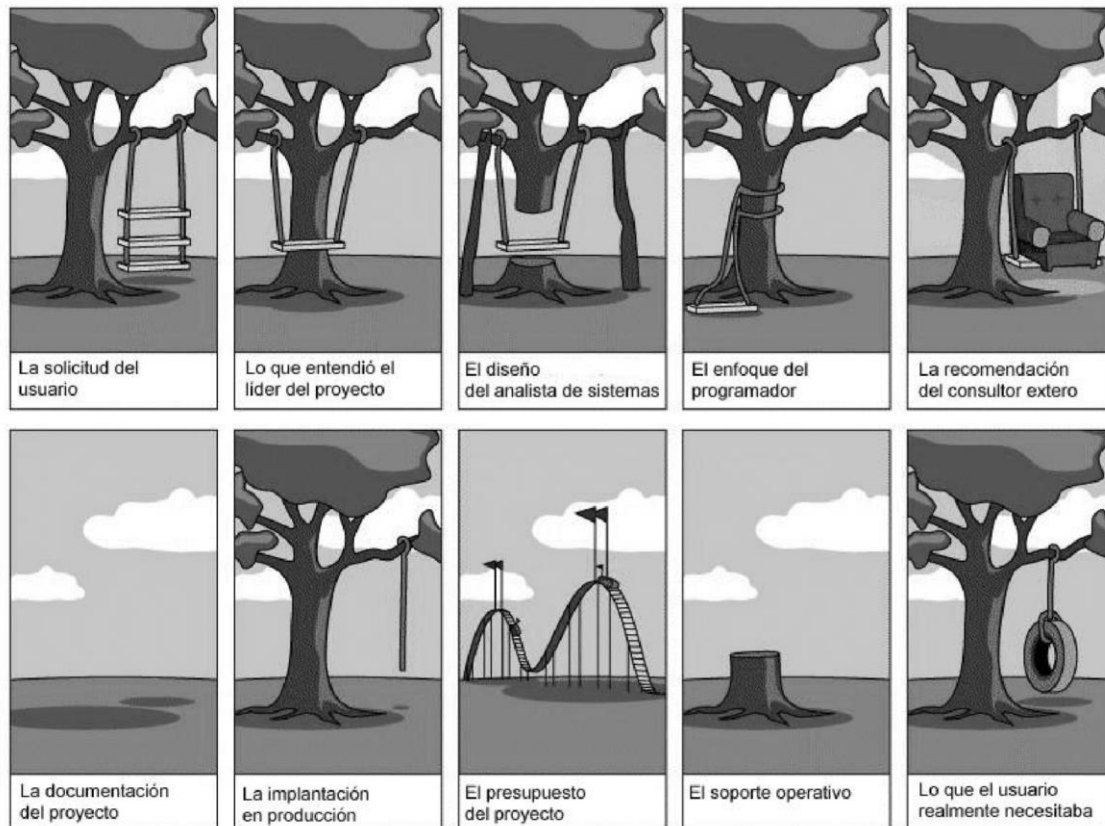


Figura 1: Problemas en el desarrollo de sistemas (ilustración anónima).

**Tipos de especificaciones.-** Los requerimientos de software pueden ser analizados de varias formas diferentes. Las técnicas de análisis pueden conducir a una especificación en papel que contenga las descripciones gráficas y el lenguaje natural de los requerimientos del software. La construcción de prototipos conduce a una especificación ejecutable, esto es, el prototipo sirve como una representación de los requerimientos. Los lenguajes de especificación formal conducen a representaciones formales de los requerimientos que pueden ser verificados o analizados.

## 2. Principios de Especificación

La especificación, independientemente del modo en que se realice, puede ser vista como un proceso de representación. Los requerimientos se representan de forma que conduzcan finalmente a una correcta implementación del software. Se plantea que una buena especificación debe procurar:

- *Separar funcionalidad de implementación.* Una especificación es una descripción de lo que se desea, en vez de como se realiza. Esto en la práctica puede llegar a no suceder del todo, sin embargo es un buen lineamiento a seguir.
- *Una especificación debe abarcar el entorno en el que el sistema opera.* Similarmente, el entorno en el que opera el sistema y con el que interactúa debe ser especificado.
- *Debe ser modificable.* Ninguna especificación puede ser siempre totalmente completa. El entorno en el que existe es demasiado complejo para ello. Una especificación es un modelo, una abstracción, de alguna situación real (o imaginada). Por tanto, será incompleta. Además, al ser formulada existirán muchos niveles de detalle.

## 3. Requerimientos funcionales y no funcionales.

La especificación debe contener los requerimientos del sistema, la IEEE-830, 1998 divide los requerimientos en funcionales y no funcionales, a continuación se describe en que consiste cada uno de estos grupos de requerimientos.

Los *requerimientos funcionales* describen una interacción entre el sistema y su ambiente, describen cómo debe comportarse el sistema ante determinado estímulo. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que

éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, también pueden declarar explícitamente lo que el sistema no debe hacer. *Los requerimientos **funcionales** de un sistema describen lo que el sistema debe hacer.*

Los *requerimientos no funcionales*: describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema. Restringen los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, el tipo de proceso de desarrollo a utilizar, fiabilidad, tiempo de respuesta, capacidad de almacenamiento. *Los requerimientos **no funcionales** ponen límites y restricciones al sistema.*

## **Tipos de Requerimientos No Funcionales:**

Se desglosa en la *figura 2.2* los tipos de requerimientos no funcionales. Los tres grupos generales son: requerimientos del producto, organizacionales y externos, de cada grupo se derivan los particulares.

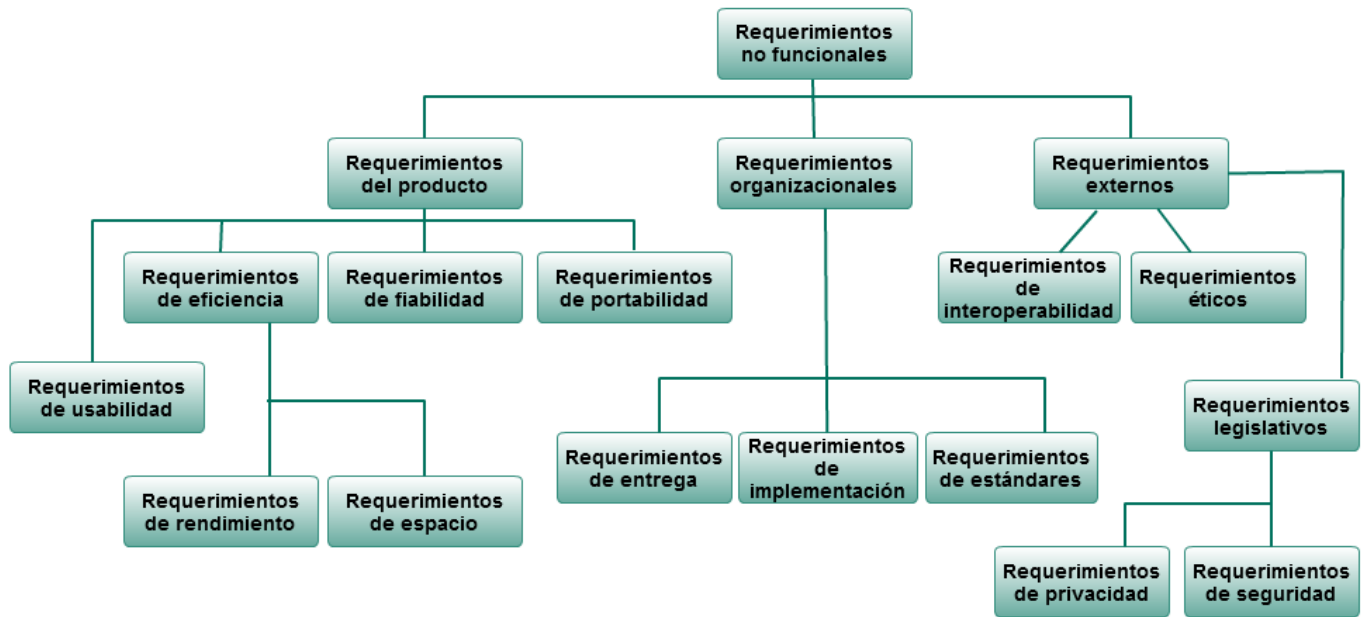


Figura 2. Tipos de requerimientos no funcionales (Sommerville, 2005)

## Requerimientos del Producto:

Especifican el comportamiento del producto. *Ejemplos: rapidez de la ejecución, capacidad de memoria, fiabilidad, etc.*

## Requerimientos Organizacionales:

Derivan de políticas y procedimientos existentes en la organización del cliente y del desarrollador. *Ejemplos: Estándares de procesos, métodos de diseño, lenguajes de programación, métodos de entrega, etc.*

## Requerimientos Externos:

Se derivan de factores externos al sistema y de sus procesos de desarrollo. *Ejemplos: Requisitos de interoperabilidad, legislativos, éticos, etc.*



**Requerimiento del producto**

8.1 La interfaz de usuario del LIBSYS se implementará como HTML simple sin marcos o applets Java.

**Requerimiento organizacional**

9.3.2 El proceso de desarrollo del sistema y los documentos a entregar deberán ajustarse al proceso y a los productos a entregar definidos en XYZCo-SP-STAN-95.

**Requerimiento externo**

10.6 El sistema no deberá revelar al personal de la biblioteca que lo utilice ninguna información personal de los usuarios del sistema aparte de su nombre y número de referencia de la biblioteca.

Figura 3: Ejemplos de requisitos no funcionales (Sommerville, 2005).

## 4. El dominio de la información.

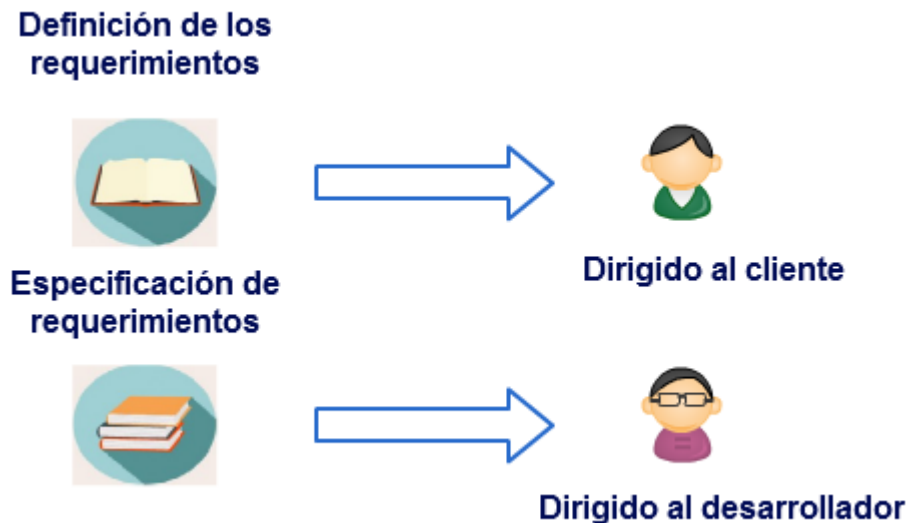
El software se construye para procesar datos; para transformar datos de una forma a otra; es decir, para aceptar una entrada, manipularla de alguna forma y producir una salida. El dominio de la información debe contener:

- *El flujo de información.*- representa la manera en la que los datos cambian conforme pasan a través de un sistema, esto se muestra, como se verá en la semana 4.
- *El contenido de la información.*- depende de la aplicación específica, por ejemplo, si se trata de un sistema de nómina, la información consistirá en los nombres de los empleados, sueldos, etc., la información es diferente para un sistema que administra un hospital, para una biblioteca, un aeropuerto, etc.
- *La estructura de la información.*- muestra la forma en la que se construye la información, es decir, sus partes y como se subdivide cada una de las partes.



## 5. La documentación

Existen dos tipos de documentos de requerimientos, el documento de *especificación de requerimientos* cubre exactamente lo mismo que el documento de *definición de los requerimientos*, la diferencia entre estos dos tipos de documentos la marca el lector, es decir, a quien están dirigidos, el nivel en el que están escritos depende si se trata del cliente o de las personas que van a desarrollar el sistema, esto se muestra con claridad en la *figura 4*:



*Figura 4:* Tipos de documentos de requerimientos.

Normalmente, la *definición de los requerimientos* está redactada en lenguaje natural, mientras que la *especificación de los requerimientos* se redacta de una forma más técnica, por ejemplo, puede definir un requerimiento que se hizo en lenguaje natural, como una serie de ecuaciones, un diagrama de flujo de datos, casos de uso, etc.

A continuación se muestra un ejemplo de un requerimiento redactado en una *definición de los requerimientos*:

INICIAR EL RASTREO DE UNA IMAGEN. Se debe hacer un procesamiento lógico para iniciar un RASTREO DE IMAGEN.

La entrada debe ser una TRANSFERENCIA DE DATOS. La salida puede ser un a IDENTIDAD DE IMAGEN, el ESTADO DE LOS DATOS o HOIQ. Cuando sea apropiado, el procesamiento lógico debe poder identificar una nueva instancia de la IMAGEN, el estado de esta nueva instancia de imagen debe ser IMAGEN EN RASTREO. Cuando se requiera el envío de pulsos, se debe ingresar un registro dentro de HOIQ el cual debe alimentar a los procedimientos de envío de pulsos.

En la *especificación*, este mismo requerimiento puede ser redactado en algún lenguaje formal de especificación y ligarse al documento de *definición de los requerimientos*, a continuación se muestra un ejemplo del estándar RSL. Aquí cabe señalar que organizaciones como el IEEE y otras organizaciones internacionales tienen estándares para el formato y contenido de los documentos de requerimientos, como por ejemplo el RSL (RAISE Specification Language, RAISE significa: Rigorous Approach to Industrial Software Engineering).

ALFA: INICIAR\_RASTREO\_IMAGEN.

ENTRADAS: TRANSFERENCIA\_DE\_DATOS.

SALIDAS: HOIQ, ESTADO\_DE\_DATOS, IDENTIDAD\_IMAGEN.

SE CREA: IMAGEN.

EL ESTADO ES: IMAGEN\_EN\_RASTREO.

DESCRIPCIÓN: la solicitud de un pulso se hace por medio de un registro formal dentro del HOIQ el cual alimenta a los procedimientos de envío de pulsos.

En el ejemplo anterior puede observarse que para el cliente es más fácil comprender la descripción dada en el documento de definición, sin embargo tendría más dificultad con la especificación escrita en RSL.

Otro ejemplo de modelo para hacer especificaciones es el SDL (Specification and Description Language). SDL es un método formal que incluye conceptos orientados a objetos. Hay herramientas comerciales disponibles para dar soporte al diseño, la depuración y el mantenimiento de los enunciados SDL. A continuación se muestra un ejemplo de una especificación formal usando SDL.

STATE SEL\_WORKING;

INPUT Clear(line);

DECISION ((cond[1line] == OOS)

|| (cond[WORK] == OOS)); (TRUE):

TASK 'badinput =1;

NEXTSTATE-

ENDDECISION;

TASK ' cond [line] = NORMAL';      DECISION ((line == PROT) &&

(cond[WORK] > cond[PROT]));

(TRUE):

NEXTSTATE SEL\_PROTECTION;

ENDDECISION;

NEXTSTATE-; ENDSTATE

SEL\_WORKING;

Existen diferentes lenguajes de especificación además de SDL y RSL, podemos mencionar: Modechart, VFSM, Esterel, Lotos, Z y C. Cuando se opta por usar un lenguaje formal de especificación, hay que estudiar que tanto se adopta el lenguaje a los criterios que se consideren importantes en los requerimientos, por ejemplo, para sistemas de lanzamiento de un cohete se toma en cuenta que con el lenguaje de especificación exista capacidad de comprobación, simulación y seguridad de ejecución.

## **Recomendación para la Especificación de Requerimientos de Software de la IEEE.**

Existe una organización muy importante a nivel internacional llamada IEEE (Institute of Electrical and Electronics Engineers, en español le llaman comunmente la I triple E). Esta organización, produce estándares que se aplican en muchas industrias del mundo. La

IEEE, edita revistas de divulgación científica con un prestigio muy alto, y también organiza congresos muy importantes en el ámbito internacional. Por lo general, los libros de texto que hablan acerca de los requerimientos de software, incluyendo estas notas, se basan en un estándar emitido por la IEEE que se aprobó en 1998, llamado: **IEEE Std 830-1998**

Std es la forma de abreviar “Standard” en inglés y el número de la especificación es 830, fue aprobada en 1998 y es una revisión de un estándar previo aceptado en 1993, Por las siglas en inglés, SRS que significan: Software Requirements Specifications, se acostumbra llamar SRS al documento de especificación.

En el **IEEE Std 830-1998** se habla sobre las características que deben tener los requerimientos (correctos, consistentes, completos, realistas, rastreables y verificables), los tipos de requerimientos (funcionales y no funcionales), así como lo que se debe tomar en cuenta al elaborarlos (ambiente físico, interfaces, usuarios y factores humanos, funcionalidad, documentación, datos, recursos, seguridad y aseguramiento de la calidad). En resumen, este estándar recomienda lo que hemos visto hasta ahora a lo largo del curso. Lo más importante del **IEEE Std 830-1998** es que define la estructura que debe tener una especificación de requerimientos, esta estructura se explica en la siguiente sección.

Normalmente, es necesario contar con un permiso para poder tener acceso al estándar **IEEE Std 830-1998** vía Internet, sin embargo, existe una traducción al español de este estándar cuyo acceso es libre, y se puede consultar buscando en google:

**[IEEE-STD-830-1998: ESPECIFICACIONES DE LOS REQUISITOS](#)**

## Estructura de una Especificación de requerimientos.

La **IEEE Std 830-1998** define la siguiente *estructura para una especificación de requerimientos*, cada una de las secciones mencionadas a continuación se detalla y se explica en el documento 830:

### 1. Introducción

#### 1.1. Objetivo

#### 1.2. Ámbito

#### 1.3. Definiciones, Siglas y Abreviaturas

#### 1.4. Referencias

#### 1.5. Visión Global

### 2. Descripción general

#### 2.1. Perspectiva del producto

#### 2.2. Funciones del producto

#### 2.3. Características del usuario

#### 2.4. Limitaciones generales

#### 2.5. Supuestos y dependencias

### 3. Requerimientos específicos

### 4. Apéndices

### 5. Índice

La **IEEE Std 830-1998** es parte de los estándares que es necesario cubrir cuando se pretende cumplir con las normas de calidad, por lo tanto, esta estructura se respeta en

la mayoría de las especificaciones de requerimientos en cualquier parte del mundo cuando se elaboran sistemas de software a nivel industrial.

En la *figura 5* se muestra gráficamente la estructura recomendada por la IEEE para una especificación de requerimientos.



*Figura 5:* Estructura de la especificación de requerimientos según IEEE std 830-1998.

En cuanto a la sección 3 *requerimientos específicos*, la **IEEE Std 830-1998** propone ocho plantillas diferentes a elegir, y son las siguientes:

A.1.- Plantilla de SRS organizada por *el modo* (versión 1).

A.2.- Plantilla de SRS organizada por *el modo* (versión 2).

A.3.- Plantilla de SRS organizada por *la clase del usuario*.

A.4.- Plantilla de SRS organizada por *el objeto*.



A.5.- Plantilla de SRS organizada por *el rasgo o característica*.

A.6.- Plantilla de SRS organizada por *el estímulo*.

A.7.- Plantilla de SRS organizada por *la jerarquía funcional*.

A.8.- Plantilla de SRS con *organización múltiple*.

A manera de ejemplo, se muestra la estructura de la sección 3 utilizando la plantilla de SRS organizada por *el modo* (versión 1).

### 3. Los requerimientos específicos

#### 3.1 requerimientos de las interfaces externas

##### 3.1.1 interfaz con el usuario

##### 3.1.2 interfaz con el hardware

##### 3.1.3 interfaz con el software

##### 3.1.4 interfaces de comunicaciones

#### 3.2 requerimientos funcionales

##### 3.2.1 modo 1

##### 3.2.1.1 requerimiento 1.1 funcional

##### 3.2.1.n requerimiento 1.n Funcional

##### 3.2.2 modo 2

##### 3.2.m Modo m

##### 3.2.2.1 requerimiento Funcional m.1

## 3.2.m.n requerimiento Funcional m.n

### 3.3 Requerimientos del desarrollo

### 3.4 Restricciones del diseño

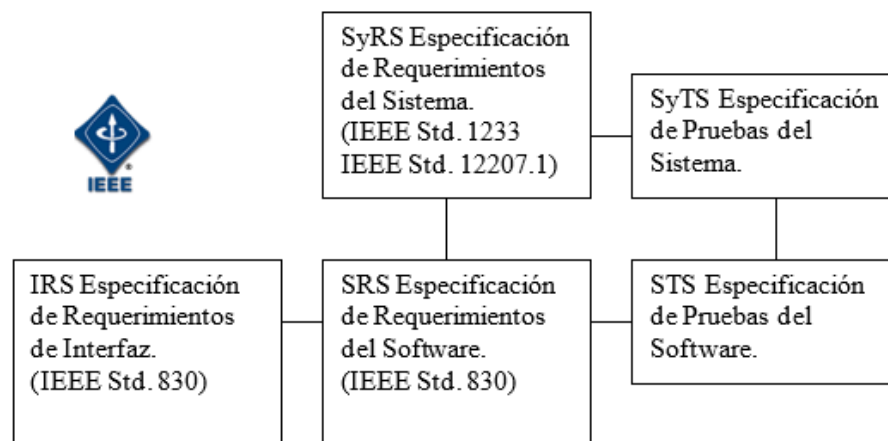
#### 3.4.1 Acatamiento de estándares

#### 3.4.2 Limitaciones hardware

### 3.5 Atributos de sistema de software

### 3.6 Otros requerimientos

Una vez obtenidos y analizados los requerimientos, se crean el **SyRS (System Requirements Specification)** y el **SRS (Software Requirements Specification)**, los estándares que se utilizan para escribir estos documentos se muestran en la *figura 2.6*.



*Figura 6: Visión global de los estándares de la IEEE.*

---

## ***Descargo de responsabilidad***

La información contenida en este documento descargable en formato PDF o PPT es un reflejo del material virtual presentado en la versión online del curso. Por lo tanto, su contenido, gráficos, links de consulta, acotaciones y comentarios son responsabilidad exclusiva de su(s) respectivo(s) autor(es) por lo que su contenido no compromete al área de e-Learning del Departamento GES o al programa académico al que pertenece.

El área de e-Learning no asume ninguna responsabilidad por la actualidad, exactitud, obligaciones de derechos de autor, integridad o calidad de los contenidos proporcionados y se aclara que la utilización de este descargable se encuentra limitada de manera expresa para los propósitos educativos del curso.

