



**El futuro  
es de todos**

**DNP**  
Departamento  
Nacional de Planeación

## **MANUAL DE ELABORACIÓN DE MODELOS CONCEPTUALES**

**Departamento Nacional de Planeación  
Bogotá, 2015**

## Tabla de contenido

1.	INTRODUCCIÓN .....	3
2.	OBJETIVO.....	4
3.	ALCANCE .....	4
4.	DEFINICIONES.....	4
4.1	Modelos y diagramas .....	4
4.2	Notación y lenguaje .....	5
4.3	Herramienta y ubicación .....	5
5.	Pasos para construir el modelo.....	5
5.1.	Definir y delimitar el alcance del dominio .....	6
5.2.	Elaborar un glosario.....	6
5.3.	Identificar Conceptos dentro del glosario .....	6
6.	Definir relaciones entre conceptos .....	6
6.1	Relaciones binarias .....	6
6.2	Conceptos universales, instancias individuales .....	6
7.	Relaciones jerárquicas o taxonómicas.....	7
8.	Tipo especialización - generalización .....	7
9.	Tipo todo - parte .....	8
10.	Relaciones no jerárquicas .....	8
10.1	Elaborar el modelo como diagrama(s) de clases.....	9
11.	Bibliografía.....	9

## 1. INTRODUCCIÓN

Niklaus Wirth, uno de los grandes pioneros y científicos de la computación, ganador del premio Turing y autor, entre otras cosas, del lenguaje de programación Pascal, escribía en 1995 un notable ensayo sobre cómo construir software de calidad, su artículo se titulaba “*Una súplica por software liviano*” (A Plea for Lean Software) (Wirth, 1995), si bien el documento se centraba en el buen uso de la orientación a objetos, hacía una crítica a la complejidad innecesaria en el diseño de software y aconsejaba centrarse en lo esencial, al final del documento concluía nueve lecciones aprendidas en las actividades de diseñar un nuevo software, hacemos énfasis en la 5ª y la 6ª que respectivamente dicen:

- *La creencia que los sistemas complejos requieren ejércitos de diseñadores y de programadores es errada. Un sistema que no es **comprendido** en su totalidad, o por lo menos a un grado de detalle suficiente **por parte de cada individuo**, probablemente no será construido.*
- *Los problemas de **comunicación** crecen a medida que el tamaño del equipo crece. Si predominan los problemas de comunicación, así sean obvios, el equipo y el proyecto estarán en problemas.*

La comprensión y la comunicación están estrechamente relacionadas: no podríamos comunicar efectivamente si el entendimiento de la conceptualización que se comparte no es aceptado unívocamente por cada uno de los involucrados; un buen entendimiento pasa por discernir ¿cuál de esa información formal y explícitamente comunicada representa conocimiento válido y relevante?

Pareciera que las lecciones aprendidas expuestas en el ensayo de Wirth tuvieran que ver menos con la ingeniería de software y más con la gestión y representación del conocimiento, pero comprensión y comunicación están naturalmente sugeridas en la definición formal de ontología como mecanismo formal para la representación del conocimiento y como componente clave en la gestión del mismo aún dentro de la ingeniería del software.


De acuerdo con Tom Gruber (Ling Liu, 2009) “*En el contexto de la informática, una ontología define un conjunto de primitivas de representación con las cuales se modela un dominio de conocimiento o discurso. Las primitivas de representación son típicamente clases (o conjuntos), atributos (o propiedades) y asociaciones (o relaciones entre sus miembros).*”

No se debe al azar esta coincidencia; una ontología define y representa los términos y conceptos relevantes a un tópico o área de interés particular (dominio) y un sistema de información es la representación (parcial o total) de un sistema de gestión del mundo real (dominio particular).

Si bien no queremos – ni debemos – desde la ingeniería del software – hacernos con la responsabilidad y la carga de manejar el formalismo y el rigor metodológico de una ontología si requerimos organizar y representar los conceptos y sus relaciones dentro del ámbito o contexto particular donde ocurre nuestro sistema de información (o de gestión, propiamente dicho) para comunicarlo y compartirlo con el grupo.

Pero la comprensión y la comunicación de una realidad, dentro del trabajo colaborativo y en equipo del desarrollo de software, si justifica la elaboración de una versión ligera de ontología, variación que efectivamente existe y se denomina modelo conceptual (o de dominio). Si para alguien no fuera suficiente esta justificación, se podrían citar algunos estudios y recomendaciones que demuestran su valor para agregar calidad, en términos cuantitativos y cualitativos, al proceso de ingeniería de software (Ben-Ari, 2006), (Vom Brocke, 2006), (Krogstie, 1995).

Cuantitativos porque un buen modelado conceptual ayuda a reducir el esfuerzo y el tiempo en elaborar los modelos posteriores de análisis (lógicos) y de diseño (físicos). Cualitativos porque un buen modelo conceptual impacta en

	<b>Manual de elaboración de modelos conceptuales</b>	CÓDIGO: PI-M01 PÁGINA: 4 de 10 VERSIÓN: 01
---	--	---

mejores modelos lógicos y físicos que producen una implementación en código que al final redunda en una sustancial disminución de defectos o fallas en el software.

## 2. OBJETIVO

Organizar y representar, de manera semi formal y unívoca, el conocimiento de un área o campo específico asociado a un sistema de gestión o de información. Está orientado a describir semántica y aseveraciones sobre la información del dominio particular que representa; la elaboración de dicho modelo es abstracta e independiente de consideraciones de diseño o de tecnología, es decir se identifican las cosas fundamentales sin dar demasiada importancia a ejemplos o instancias particulares y sin dejarse influenciar por la eventual participación de estos conceptos en elementos o componentes de software o de una solución de informática.

## 3. ALCANCE

Comienza con las definiciones y conceptos relacionados continúa con los formalismos de su representación y termina con la descripción de los pasos requeridos para obtenerlo.

## 4. DEFINICIONES

El modelo conceptual es un entregable inicial dentro del proceso de construcción de software, esto quiere decir que está antes del tiempo y del espacio de las disciplinas de análisis y diseño, podríamos afirmar que está a caballo entre el modelado de negocio y los requerimientos. Por esta razón para crear un modelo conceptual es suficiente con una buena definición y explicación de conceptos o entidades de negocio y de sus relaciones. Pero su vigencia y alcance estará presente a lo largo de todas las fases del proyecto.

Dado que su utilidad primordial es dar contexto y comprensión a los requerimientos y compartir ese entendimiento entre los miembros del equipo, un modelo conceptual aplica tanto para proyectos de desarrollo de software transaccionales (OLTP) como de inteligencia de negocios (OLAP).

Tiene como requisitos previos: Documento de visión, modelos de negocio, especificación de requerimientos (en forma de casos de uso de alto nivel o historias de usuario); es deseable que exista una especificación detallada de requerimientos y una versión del glosario de proyecto.

### 4.1 Modelos y diagramas

Para evitar confusión entre los términos modelo y diagrama es imprescindible hacer una distinción y no tratarlos como sinónimos. La simple representación gráfica de un modelo (el diagrama) es tan sólo una vista parcial (y opcional) de éste, que muestra un subconjunto de su notación o sintaxis concreta. La gran utilidad de un modelo está en lo que éste responde o dice sobre la realidad (o sistema) que representa, es decir, su significado o semántica, similar al sistema real pero más simplificado y fácil de usar.

Teniendo sintaxis y semántica (notación y significado), los modelos son constructos lingüísticos que comprenden conjuntos de declaraciones (proposiciones con valor de verdad) que, o bien describen un Sistema Bajo Estudio (como es el caso del modelo conceptual), o bien prescriben (o especifican) un sistema posible.

Un modelo descriptivo es válido si todas sus proposiciones son verdaderas, un modelo prescriptivo se considera válido por principio y el sistema que se construya (y que represente) debe tener conformidad con sus proposiciones.

Además de la validez del modelo en relación con la realidad que representa (descriptiva o prescriptiva), los modelos deben tener conformidad con la sintaxis que estructura la lógica del lenguaje que determina las reglas de las expresiones bien formadas (sintaxis abstracta o meta modelo) y el conjunto de símbolos de su notación (sintaxis concreta o modelo), notación que puede ser gráfico o textual.

La distinción no admite controversia; el diagrama es un elemento gráfico (dibujo) que hace parte de un modelo. Con frecuencia un modelo contiene otros elementos diferentes a los componentes gráficos incluidos en los diagramas.

Solamente cuando un modelo es susceptible de representarse suficientemente con un diagrama se podría usar indistintamente ambos términos.

## 4.2 Notación y lenguaje

Un modelo conceptual se representa en uno o más diagramas de clases con un alto nivel de abstracción. Un diagrama de clases es un grafo acíclico dirigido (mapa) de los conceptos y sus relaciones. Este, o estos diagramas se elaboran en notación UML y hace uso fundamental de los elementos básicos gramaticales, ortográficos y semánticos de la versión 2.4.1 de este lenguaje para este tipo de diagrama, esto es: clases, atributos y relaciones.

La notación no sólo determina que el modelo se muestre en una notación icónica o gráfica (diagramas), también exige que se documente cada elemento identificado y/o mostrado en el diagrama, por lo que requiere un repositorio para incluir los elementos textuales que conforman dicha documentación.

## 4.3 Herramienta y ubicación

Los proyectos o equipos de trabajo, que consideren elaborar un diagrama conceptual, pueden utilizar la herramienta Enterprise Architect y ubicarlo en un repositorio que se haya creado en una base de datos SQL Server dentro de los servidores de DNP dispuestos para el proyecto o servicio.

## 5. Pasos para construir el modelo.

No hay una única forma de modelar un dominio - siempre hay alternativas viables y correctas -. La mejor solución es aquella que logra el objetivo de representar y de hacer explícito el conocimiento y el entendimiento de ese dominio. Existen diversos enfoques metodológicos tomados desde la gestión del conocimiento que implican gran formalismo y mayor rigor metódico, en este manual proponemos una metodología con pocos pasos, algunos de los cuales ya se han recorrido con otras actividades del esquema de desarrollo, estos pasos son:

1. Definir y delimitar el alcance del dominio.
2. Elaborar un glosario
3. Identificar conceptos dentro del glosario.
4. Definir relaciones entre conceptos
  - a. Relaciones jerárquicas o taxonómicas
    - i. Tipo Especialización generalización
    - ii. Tipo todo-parte, agregaciones y composiciones.
  - b. Relaciones no jerárquicas entre pares (asociaciones)
5. Elaborar los diagramas de clases utilizando los conceptos y sus relaciones identificadas

## 5.1. Definir y delimitar el alcance del dominio

La primera tarea es definir y acotar el dominio o campo de nuestro modelo. Esta actividad ya está trabajada en el documento de visión, puntualmente en la definición del alcance funcional y organizacional que le hemos dado al proyecto. Así que nuestro paso consiste en verificar que existe esta sección dentro del documento de visión.

## 5.2. Elaborar un glosario

El glosario del dominio del proyecto – o glosario del proyecto – es el primer insumo en esta identificación de conceptos, constituye un documento formal cuya estructura y contenido ha sido definido y cuenta con su propio manual de elaboración. (Ver Manual de elaboración del Glosario de Términos en la rebeca).

## 5.3. Identificar Conceptos dentro del glosario

Es altamente probable que ya exista una versión del glosario para el proyecto, puesto que usualmente se elabora paralelamente al documento de visión. Si ya existe se debe examinar para extraer los principales términos identificados y definidos allí que hacen referencia a sustantivos (o frases sustantivas), esto es, que nombran entidades del mundo real que tienen existencia discreta e identificable y que son susceptibles de expresarse y comprenderse a través de atributos y características diferenciales. Estos conceptos también tienen el carácter de universalidad, esto es, que no determinan instancias singulares (o individuales) del concepto.

Si el glosario aún no existe se debe comenzar a elaborar con las consideraciones del numeral anterior y con el propósito de identificar entidades como conceptos universales, pues este es un paso imprescindible en obtener un modelo de dominio.

## 6. Definir relaciones entre conceptos


### 6.1 Relaciones binarias

Debemos recordar – o reiterar – que el nivel de abstracción de nuestro modelo es de carácter conceptual, pero no obstante representa un sistema de gestión de la vida real y en este entorno es inusual y poco probable encontrar términos o conceptos aislados, por el contrario la mayor parte interactúan o se estructuran de diversas formas. La elaboración del modelo de dominio hace obligatorio explorar y descubrir esas relaciones.

El diagrama conceptual se orienta a describir esencialmente las relaciones binarias (entre dos conceptos), en una definición simple una relación es una conexión con contenido semántico - es decir, significativa e interesante – entre entidades dentro del problema de dominio, estas conexiones pueden involucrar tanto las propiedades (atributos) como las asociaciones de interacción, en cualquier caso las relaciones dentro de un modelo de dominio deben, además, considerarse desde el punto de vista estructural o estático.

### 6.2 Conceptos universales, instancias individuales

El término “Ministerio De Educación” lo utilizamos con un sentido único para referirnos a una entidad concreta siempre y cuando hayamos delimitado el alcance y el dominio de nuestro modelo conceptual (por ejemplo, el estado Colombiano) en cuyo caso podemos describirlo en términos de propiedades o atributos (ubicación, personas de contacto, persona a cargo de la dirección general, presupuesto de gastos asignados para un año, etc.) de lo contrario, sin un dominio delimitado, es una entidad o concepto abstracto (ente u organismo oficial y público

	<b>Manual de elaboración de modelos conceptuales</b>	CÓDIGO: PI-M01
		PÁGINA: 7 de 10 VERSIÓN: 01

encargado de tratar los asuntos administrativos relacionadas con la educación) que puede aplicarse de manera común a otros conceptos o entidades: “Ministerio de Educación de Chile”, “Ministerio de Educación de Perú”, “Ministerio de Educación de Bolivia”, “Ministerio de Educación de Colombia”, etc.

La interpretación del término “Ministerio De Educación” sin un dominio específico se denomina universal porque lo podemos asociar a un sustantivo (o frase sustantiva) común o genérica que agrupa una pluralidad de objetos que conocen por sus características comunes, sin expresar rasgos distintivos, es decir hace referencia a una clase de objetos. La lista de términos que particulariza a cada elemento frente a los demás de una misma clase o especie se llaman individuales porque distinguen instancias propias, en el caso del ejemplo, es un “Ministerio De Educación” acotado, en el nombre a un país particular, incluso pueden haber otras características o atributos para remarcar esa diferenciación: “Ministerio de Educación y Cultura de Uruguay”, “Ministerio de Educación de la Nación Argentina”.

La diferencia entre universal e individual (o clase e instancia) es intuitivamente comprendida y no requiere mayor explicación, la dificultad e importancia de la correcta interpretación de un concepto en una de estas categorías es fundamental en un modelo de dominio y depende precisamente de este alcance.

Las relaciones pueden ser homogéneas en el sentido de la categoría de elementos que conecta, esto es universales con universales, individuales con individuales o pueden ser heterogéneas.

Un tipo de relación heterogénea es la clasificación, que se da entre universales e individuos. La sola definición de una entidad universal declara tácitamente una relación con sus individuos particulares que se denomina instanciación, esta correspondencia entre instancias específicas con su entidad universal se conoce como del tipo “es un”, por ejemplo, el Departamento Nacional de Planeación “es una” Entidad Pública, aquí, “Entidad Pública” es el universal y “Departamento nacional de Planeación” es el individual

Esta asociación entre conceptos configura la forma más sencilla, común e implícita de relación en un modelo de dominio, tan implícita que no es necesario explicitarse dentro de un diagrama conceptual.

## 7. Relaciones jerárquicas o taxonómicas

## 8. Tipo especialización - generalización

Es la relación jerárquica en la que se identifica a los conceptos por su pertenencia a una categoría, en la que un concepto genérico se considera superordinado de otros conceptos más específicos. Los conceptos subordinados comparten las características del concepto genérico pero, además, poseen algunas peculiaridades propias que los diferencian y hacen más específicos. Se establece, por tanto, una relación que va en dos sentidos diferentes: vertical, es decir, la que se establece entre un concepto específico (subordinado) y su genérico (superordinado) y horizontal, la que sostienen varios conceptos específicos que poseen el mismo genérico y que se diferencian entre sí por poseer alguna característica distintiva (conceptos coordinados).

Son las relaciones del tipo “clase/sub-clase, “tipo/sub-tipo” o “padre/hijo. La generalización, como relación binaria, implica recíprocamente la especialización; en una relación de especialización/generalización los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del elemento general (el padre). Las jerarquías creadas con este tipo de relación permiten manejar la complejidad, ordenando los objetos dentro de un árbol de abstracción creciente (generalización) o decreciente (especialización), Estas estructuras son utilizadas convenientemente para implementar clasificaciones, particularmente clasificaciones taxonómicas.



## 9. Tipo todo – parte

Se refiere a la relación que existe entre conceptos que están formados por más de una parte y dichas partes constituyentes. En lingüística se denomina meronimia a este tipo de relaciones semánticas no-simétricas entre los significados de dos palabras dentro del mismo campo semántico. Se denomina merónimo a la palabra cuyo significado constituye una parte del significado total de otra palabra, a la cual se denomina holónimo, por ejemplo, hoja es un merónimo de libro y libro es merónimo de biblioteca pero a su vez: biblioteca es holónimo de libro, y libro es holónimo de hoja. En términos más coloquiales: un libro está compuesto de hojas y una biblioteca está compuesta de libros.

En el modelamiento conceptual es la típica asociación “todo/parte”, que especifica la relación entre el agregado (todo) y los componentes (partes); en los ejemplos anteriores, libro y biblioteca son el “todo” respecto a las “partes” hoja y libro.

Es, como puede verse, una asociación binaria con la excepción, entre otras, que las instancias no pueden tener agregaciones cíclicas, es decir, una parte no puede contener el todo (un libro no podría contener una biblioteca).

Adicionalmente existen características que la distinguen de otro tipo de asociación:

- La agregación es una relación asimétrica, no es entre pares.
- La agregación es una relación transitiva (la biblioteca también es, conceptualmente, un agregado de hojas).
- La agregación es una asociación que sólo puede ser binaria, darse entre dos clases.
- La agregación implica un acoplamiento fuerte, en dónde las consecuencias usualmente se propagan (si desaparecen todos los libros de una biblioteca, ya no existirá la biblioteca; si se agregan nuevos capítulos – hojas - a un libro, ya no será el mismo libro).

Debido a que la agregación no es simétrica, es importante distinguir visualmente cuál de las clases en la relación es el agregado y cuál es el componente, en la representación visual del diagrama, con UML, esto se logra adornando la asociación con un pequeño diamante en el lado del agregado.

El hecho de ser una agregación significa que una instancia de parte puede existir sin importar la existencia del todo (puedo tener libros así no tenga una biblioteca y puede haber hojas con contenido, así no hagan parte de un libro).

Adicionalmente existen otras formas de agregación más fuerte llamadas composiciones en donde lo dicho en el párrafo anterior no aplica, es decir, la existencia de la parte depende del todo y no es posible tener una parte si no se cuenta con el todo. Es también importante aclarar que, en la composición, la parte sólo puede pertenecer a un único todo.

En UML la representación visual de la composición y de la agregación están diferenciadas por elementos de la notación debido a que su carga semántica así lo exige, por tanto si así ocurre en la realidad, el modelo lo debe representar en concordancia.

## 10. Relaciones no jerárquicas

Estas son asociaciones binarias simples que se dan entre entidades o conceptos pares o del mismo nivel, es decir, ninguna de las entidades en la asociación tiene mayor precedencia o más importancia que la otra. Ejemplos de este tipo de relaciones son las de causa-efecto, actividad-lugar de realización, proceso-producto, etc.

Es importante que además de identificar y representar estas relaciones en un diagrama se documente el significado de dicha relación, desde la perspectiva de cada rol y se defina la cardinalidad de mapeo (uno a uno, uno a muchos, muchos a muchos) y la obligatoriedad u opcionalidad de ocurrencia de la relación.



## 10.1 Elaborar el modelo como diagrama(s) de clases

Como nuestro propósito – insistimos – es representar y comunicar conocimiento a un grupo plural de personas, los modelos con mayor efectividad en este objetivo son aquellos que facilitan su lectura y logran mayor eficacia en su comprensión; en tal sentido la densidad de elementos dentro del modelo es un factor determinante en su interpretación, pues un número muy grande de elementos obliga a representarlo en espacios (diagramas) con dimensiones de difícil manejo, por ejemplo, para imprimirlos requieren un dispositivo especial o el trabajo manual de pegar las partes; su lectura requiere exponerlos en una mesa o pared y obliga a un esfuerzo mayor de entendimiento por la cantidad de elementos y relaciones en el dibujo.

El lineamiento de la Oficina de Tecnologías y Sistemas de Información para presentar estos modelos requiere que los espacios de dibujo (los diagramas de clases) donde se distribuyen las entidades o conceptos no rebasen una hoja del tamaño que pueda lograrse en una impresora normal de las provistas por el DNP., esto es tamaño carta o tamaño oficio, si la complejidad o alcance del contexto real que se pretende modelar sobrepasa esta consideración, entonces se debe dividir el modelo en diagramas más pequeños que si cumplan este lineamiento. Siempre es posible, además de deseable, identificar temas conceptuales o de negocio para los cuales se elaboren diagramas de clases para conformar el modelo total. Si el análisis del dominio o contexto no juzga pertinente dividir un diagrama según estos lineamientos, se debe adjuntar una justificación para presentar diagramas con tamaños superiores a los ya mencionados.

El nivel de detalle de los modelos UML, es también un factor determinante no sólo para su comprensión sino para garantizar calidad; calidad del diagrama propiamente dicho y de su repercusión en los posteriores artefactos y modelos relacionados, incluyendo el software en si mismo (Ariadi Nugroho, 2008). En un modelo conceptual con alto nivel de abstracción los detalles no sólo son prescindibles en ocasiones son innecesarios y molestos.

Entonces, para completar cada diagrama que hace parte del modelo conceptual o de dominio bastan unas reglas simples de nomenclatura en notación UML y un bajo nivel de detalle para los conceptos o entidades; pero para las asociaciones, sí se debe proveer documentación de su significado y aportar la definición de su tipo (taxonómica, todo-parte, asociación simple), su multiplicidad o precisión cardinal de cada rol en relación identificada. Para la explicación de los conceptos se debe usar por lo menos una de dos alternativas:

1. Exponer alguna(s) propiedad(es) de información que sirva(n) para identificar, describir, cualificar o cuantificar cada instancia o ejemplo de un concepto;
2. Utilizar una nota de texto que aclare y dilucide el concepto.


## 11. Bibliografía.

Ariadi Nugroho, B. F. (2008). Empirical Analysis of the Relation between Level of Detail in UML Models and Defect Density. (Springer, Ed.) *Systems, MoDELS '08 Proceedings of the 11th international conference on Model Driven Engineering Languages and*.

Ben-Ari, M. a. (2006). Conceptual models of software artifacts. *Interacting with Computers*, 1336-150.

DNP, J. V. (18 de Junio de 2010). *Lineamientos Técnicos para Aplicativos y Portales Web Informáticos*. Obtenido de Guía de elaboración Glosario de Terminos: <http://larebeca/LinkClick.aspx?fileticket=7npFcELZ0qU%3d&tabid=724>

Española, R. A. (s.f.). RAE. Recuperado el 23 de Abril de 2012, de Diccionario de la Lengua: [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=modelo](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=modelo)

	<b>El futuro es de todos</b> DNP Departamento Nacional de Planeación	<b>Manual de elaboración de modelos conceptuales</b>	CÓDIGO: PI-M01
			PÁGINA: 10 de 10    VERSIÓN: 01

Krogstie, J. a. (1995). Defining quality aspects for conceptual models. *ISCO*, 216-231.

Ling Liu, T. O. (2009). *Encyclopedia of Database Systems*. Springer.

Perez, C. (2006). Explotación de los corpora textuales informatizados para la creación de bases de datos terminológicas basadas en el conocimiento escrito. *Revista electrónica Estudios de Lingüística Española (ELiEs)*, Universidad de Málaga.

Vom Brocke, J. a. (2006). Reusable conceptual models--requirements based on the design science research paradigm. *Proceedings of the 1st International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006)*, Claremont, 576-604.

Wirth, N. (1995). A Plea for Lean Software. *Computer Magazine*, 64 - 68.

**Fecha aprobación: 21 de Julio 2015**

Revisó:

\_\_\_\_\_  
Carlos Alberto Ferrer Infante  
Coordinador Grupo de Gestión de Proyectos

Aprobó:

\_\_\_\_\_  
Diego Alexander Gonzalez Flórez  
Jefe de Tecnologías y Sistemas de Información