



30  
años

Licenciada por  
SUNEDU  
para que puedas  
salir adelante

## SESIÓN 07:

Gestionar Sistemas de Información para satisfacer necesidades organizacionales de forma innovadora, respondiendo a estándares de calidad



# Resultado de Aprendizaje

Aplica los conceptos del modelo Entidad – Relación y Normalización para el desarrollo de modelos según casos.

## TEMARIO:

### TEMA 3: Diseño de Una Base de Datos.

#### Contenido

- 🔍 Normalización
- 🔍 Modelo Lógico
- 🔍 Transformación del MER al modelo Lógico
- 🔍 Modelo Físico





# NORMALIZACIÓN

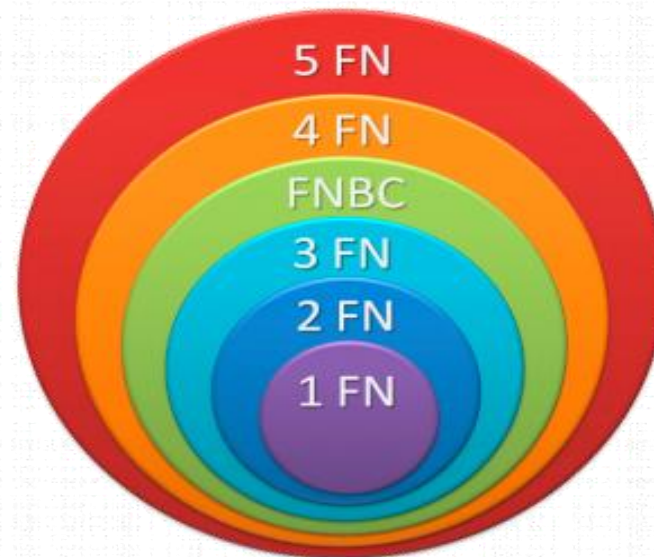
Permiten asegurar que un esquema relacional cumple unas ciertas propiedades, evitando:

- ✓ La redundancia de los datos: repetición de datos en un sistema.
- ✓ Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- ✓ Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- ✓ Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.



# FORMAS NORMALES

Proporcionan los criterios para determinar el grado de vulnerabilidad de una tabla a inconsistencias y anomalías lógicas. Cuanto más alta sea la forma normal aplicable a una tabla, menos vulnerable será a inconsistencias y anomalías.





# PRIMERA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

Una Relación está en 1FN si y sólo si cada atributo es atómico.

**Ejemplo:** Supongamos que tenemos la siguiente tabla con datos de alumnado de un centro de enseñanza secundaria.

DNI	Nombre	Curso	FechaMatrícula	Tutor	LocalidadAlumno	ProvinciaAlumno	Teléfonos
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla	660111222
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla	660222333 660333444 660444555
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla	
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba	
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba	661000111 661000222

Como se puede observar, esta tabla no está en **1FN** puesto que el campo Teléfonos contiene varios datos dentro de una misma celda y por tanto no es un campo cuyos valores sean atómicos.



# PRIMERA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

La solución es la siguiente:

Alumnos						
DNI	Nombre	Curso	FechaMatricula	Tutor	LocalidadAlumno	ProvinciaAlumno
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba

Teléfonos	
DNI	Teléfono
11111111A	660111222
22222222B	660222333
22222222B	660333444
22222222B	660444555
55555555E	661000111
55555555E	661000222



## SEGUNDA FORMA NORMAL

Pregrado

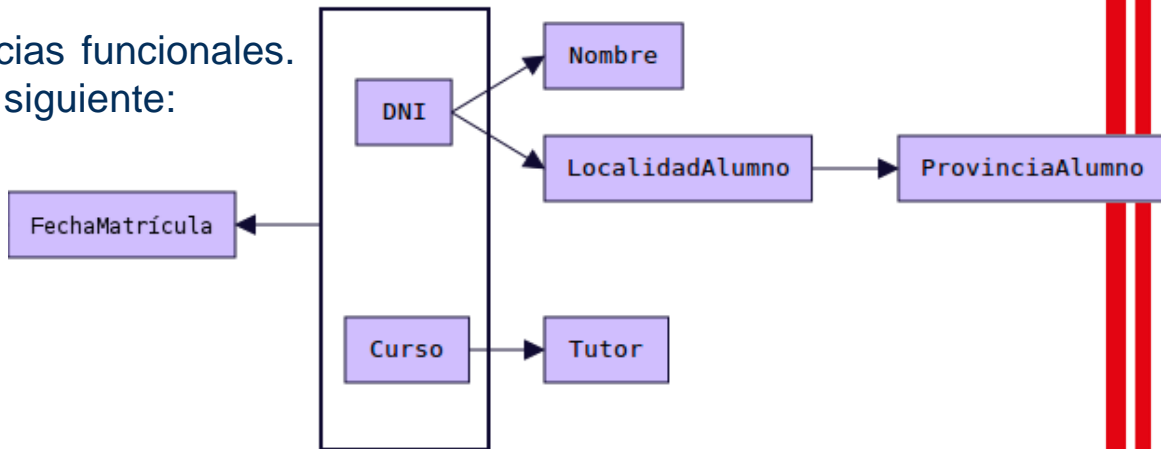
Ingeniería de  
Sistemas

Una Relación esta en 2FN si y sólo si está en 1FN y todos los atributos que no forman parte de la Clave Principal tienen dependencia funcional completa de ella.

**Ejemplo:** Seguimos con el ejemplo anterior. Trabajaremos con la siguiente tabla:

DNI	Nombre	Curso	FechaMatrícula	Tutor	LocalidadAlumno	ProvinciaAlumno
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba

Vamos a examinar las dependencias funcionales.  
El gráfico que las representa es el siguiente:





## ANÁLISIS DEL EJEMPLO

- ✓ Siempre que aparece un **DNI** aparecerá el Nombre correspondiente y la LocalidadAlumno correspondiente.
- ✓ Por tanto **DNI** → **Nombre** y **DNI** → **LocalidadAlumno**.
- ✓ Por otro lado siempre que aparece un Curso aparecerá el Tutor correspondiente.
- ✓ Por tanto **Curso** → **Tutor**. Los atributos **Nombre** y **LocalidadAlumno** no dependen funcionalmente de **Curso**, y el atributo **Tutor** no depende funcionalmente de **DNI**.
- ✓ El único atributo que sí depende de forma completa de la clave compuesta **DNI** y **Curso** es **FechaMatrícula: (DNI,Curso) → FechaMatrícula**.





## ANÁLISIS DEL EJEMPLO

- ✓ A la hora de establecer la Clave Primaria de una tabla debemos escoger un atributo o conjunto de ellos de los que dependan funcionalmente el resto de atributos. Además debe ser una dependencia funcional completa.
- ✓
- ✓ Si escogemos **DNI** como clave primaria, tenemos un atributo (**Tutor**) que no depende funcionalmente de él.
- ✓ Si escogemos **Curso** como clave primaria, tenemos otros atributos que no dependen de él.
- ✓ Si escogemos la combinación (**DNI, Curso**) como clave primaria, entonces sí tenemos todo el resto de atributos con dependencia funcional respecto a esta clave. Pero es una dependencia parcial, no total (salvo **FechaMatrícula**, donde sí existe dependencia completa). Por tanto esta tabla no está en **2FN**.

Alumnos			
DNI	Nombre	Localidad	Provincia
11111111A	Eva	Écija	Sevilla
22222222B	Ana	Écija	Sevilla
33333333C	Susana	El Villar	Córdoba
44444444D	Juan	El Villar	Córdoba
55555555E	José	Écija	Sevilla

Matrículas		
DNI	Curso	FechaMatrícula
11111111A	1ESO-A	01-Julio-2016
22222222B	1ESO-A	09-Julio-2016
33333333C	1ESO-B	11-Julio-2016
44444444D	2ESO-A	05-Julio-2016
55555555E	2ESO-A	02-Julio-2016

Cursos	
Curso	Tutor
1ESO-A	Isabel
1ESO-B	Roberto
2ESO-A	Federico



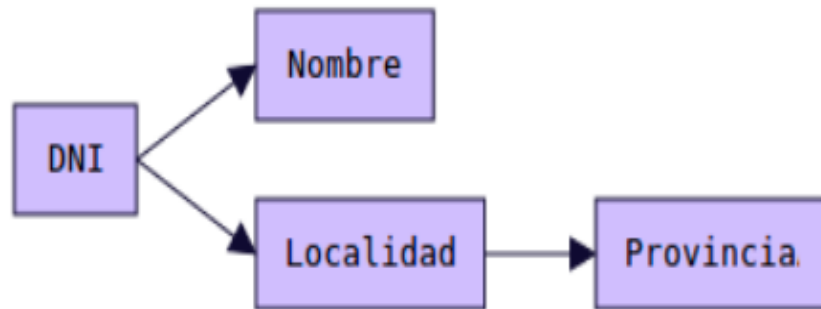
## TERCERA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

Una Relación esta en **3FN** si y sólo si está en **2FN** y no existen dependencias transitivas. Todas las dependencias funcionales deben ser respecto a la clave principal. Seguimos con el **ejemplo** anterior. Trabajaremos con la siguiente tabla:

Alumnos			
DNI	Nombre	Localidad	Provincia
11111111A	Eva	Écija	Sevilla
22222222B	Ana	Écija	Sevilla
33333333C	Susana	El Villar	Córdoba
44444444D	Juan	El Villar	Córdoba
55555555E	José	Écija	Sevilla



Las dependencias funcionales existentes son las siguientes. Como podemos observar existe una dependencia funcional transitiva: **DNI** → **Localidad** → **Provincia**



# TERCERA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

Para que la tabla esté en **3FN**, no pueden existir dependencias funcionales transitivas.  
Para solucionar el problema deberemos crear una nueva tabla. El resultado es:

Alumnos		
DNI	Nombre	Localidad
11111111A	Eva	Écija
22222222B	Ana	Écija
33333333C	Susana	El Villar
44444444D	Juan	El Villar
55555555E	José	Écija

Localidades	
Localidad	Provincia
Écija	Sevilla
El Villar	Córdoba



# RESULTADO FINAL

Pregrado

Ingeniería de  
Sistemas

Para que la tabla esté en **3FN**, no pueden existir dependencias funcionales transitivas.  
Para solucionar el problema deberemos crear una nueva tabla. El resultado es:

Alumnos		
DNI	Nombre	Localidad
11111111A	Eva	Écija
22222222B	Ana	Écija
33333333C	Susana	El Villar
44444444D	Juan	El Villar
55555555E	José	Écija

DNI	Teléfono
11111111A	660111222
22222222B	660222333
22222222B	660333444
22222222B	660444555
55555555E	661000111
55555555E	661000222

Matrículas		
DNI	Curso	FechaMatrícula
11111111A	1ESO-A	01-Julio-2016
22222222B	1ESO-A	09-Julio-2016
33333333C	1ESO-B	11-Julio-2016
44444444D	2ESO-A	05-Julio-2016
55555555E	2ESO-A	02-Julio-2016

Localidades	
Localidad	Provincia
Écija	Sevilla
El Villar	Córdoba

Cursos	
Curso	Tutor
1ESO-A	Isabel
1ESO-B	Roberto
2ESO-A	Federico



Una Relación esta en **FNBC** si está en **3FN** y no existe solapamiento de claves candidatas. Solamente hemos de tener en cuenta esta forma normal cuando tenemos varias claves candidatas compuestas y existe solapamiento entre ellas. Pocas veces se da este caso.

**Ejemplo:** Tenemos una tabla con información de proveedores, códigos de piezas y cantidades de esa pieza que proporcionan los proveedores.  
Cada proveedor tiene un nombre único. Los datos son:

Suministros			
CIF	Nombre	CódigoPieza	CantidadPiezas
S-11111111A	Ferroman	1	10
B-22222222B	Ferrotex	1	7
M-33333333C	Ferropet	3	4
S-11111111A	Ferroman	2	20
S-11111111A	Ferroman	3	15
B-22222222B	Ferrotex	2	8
B-22222222B	Ferrotex	3	4

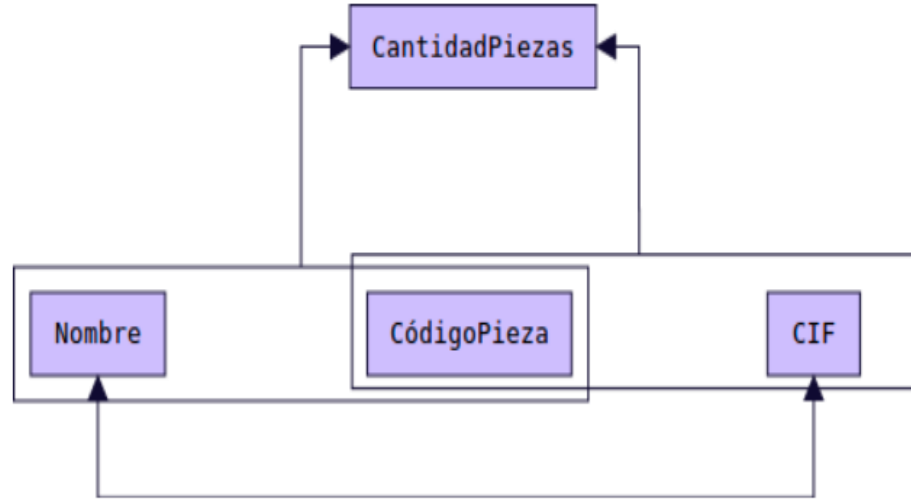


# FORMA NORMAL BOYCE-CODD: FNBC

Pregrado

Ingeniería de  
Sistemas

El gráfico de dependencias funcionales es el siguiente:



El atributo **CantidadPiezas** tiene dependencia funcional de dos claves candidatas compuestas, que son:

- ✓ **(NombreProveedor, CódigoPieza)**
- ✓ **(CIFProveedor, CódigoPieza)**

Existe también una dependencia funcional en doble sentido (que no nos afecta):  
**NombreProveedor <-> CIFProveedor.**



# FORMA NORMAL BOYCE-CODD: FNBC

Pregrado

Ingeniería de  
Sistemas

Para esta tabla existe un solapamiento de 2 claves candidatas compuestas. Para evitar el solapamiento de claves candidatas dividimos la tabla. La solución es:

Suministros		
CIF	CódigoPieza	CantidadPiezas
S-11111111A	1	10
B-22222222B	1	7
M-33333333C	3	4
S-11111111A	2	20
S-11111111A	3	15
B-22222222B	2	8
B-22222222B	3	4

Proveedores	
CIF	Nombre
S-11111111A	Ferroman
B-22222222B	Ferrotex
M-33333333C	Ferropet





# CUARTA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

Una Relación esta en **4FN** si y sólo si está en **3FN** (o **FNBC**) y las únicas dependencias multivaluadas son aquellas que dependen de las claves candidatas.

**Ejemplo:** Tenemos una tabla con la información de nuestros alumnos y alumnas y las asignaturas que cursan así como los deportes que practican.

Alumnado		
Estudiante	Asignatura	Deporte
11111111A	Matemáticas	Natación
11111111A	Matemáticas	Baloncesto
11111111A	Lengua	Natación
11111111A	Lengua	Baloncesto
22222222B	Matemáticas	Fútbol
22222222B	Matemáticas	Natación

Alumnado		
Estudiante	Asignatura	Deporte
11111111A	Matemáticas, Lengua	Baloncesto
22222222B	Matemáticas	Fútbol, Natación



## CUARTA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

Para normalizar esta tabla, debemos darnos cuenta que la oferta de asignaturas está compuesta por un conjunto de valores limitado. Igual sucede con los deportes. Por tanto existen dos dependencias multivaluadas:

• **Estudiante** →→ **Asignatura**

• **Estudiante** →→ **Deporte**

Por otro lado no existe ninguna dependencia entre la asignatura cursada y el deporte practicado. Para normalizar a **4FN** creamos 2 tablas:

EstudiaAsignatura	
Estudiante	Asignatura
11111111A	Matemáticas
11111111A	Lengua
22222222B	Matemáticas

PracticaDeporte	
Estudiante	Deporte
11111111A	Natación
11111111A	Baloncesto
22222222B	Fútbol
22222222B	Natación

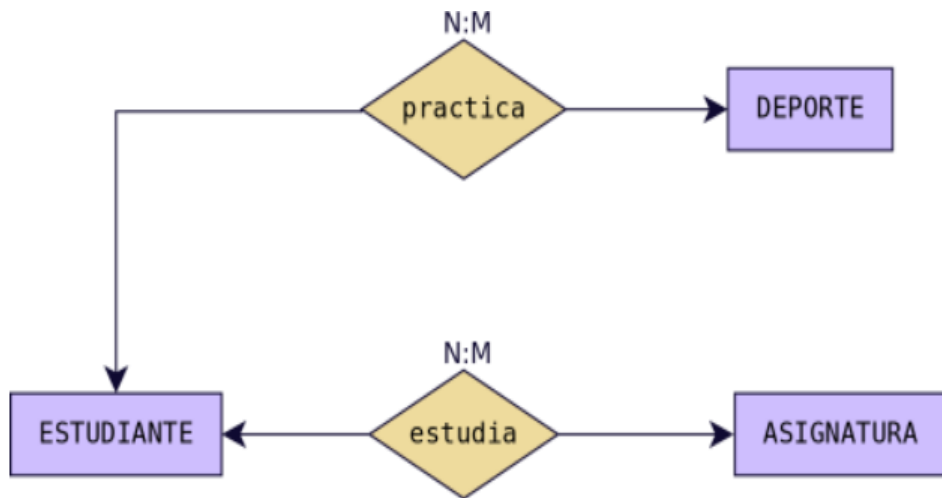


Diagrama E/R equivalente



## QUINTA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

También conocida como forma normal de proyección-uniión (**PJ/NF**). Una tabla se dice que está en **5NF** si y sólo si está en **4NF** y cada dependencia de unión (**join**) en ella es implicada por las claves candidatas.

**Ejemplo:** Tenemos una tabla con varios proveedores que nos proporcionan piezas para distintos proyectos. Asumimos que un Proveedor suministra ciertas Piezas en particular, un Proyecto usa ciertas Piezas, y un Proyecto es suplido por ciertos Proveedores, entonces tenemos las siguientes dependencias multivaluadas:

Proveedor  $\twoheadrightarrow$  Pieza

Pieza  $\twoheadrightarrow$  Proyecto

Proyecto  $\twoheadrightarrow$  Proveedor

**Se puede observar como se produce un ciclo:**

Proveedor  $\twoheadrightarrow$  Pieza  $\twoheadrightarrow$  Proyecto  $\twoheadrightarrow$  Proveedor (nuevamente)



# QUINTA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

Suministros		
Proveedor	Pieza	Proyecto
E1, E4, E6	PI3, PI6	PR2, PR4
E2, E5	PI1, PI2	PR1, PR3
E3, E7	PI4, PI5	PR5, PR6

Suministros		
Proveedor	Pieza	Proyecto
E1	PI3	PR2
E1	PI3	PR4
E1	PI6	PR2
E1	PI6	PR4
E4	PI3	PR2
E4	PI3	PR4
E4	PI6	PR2
E4	PI6	PR4
E6	PI3	PR2
E6	PI3	PR4
E6	PI6	PR2
E6	PI6	PR4
E2	PI1	PR1
E2	PI1	PR3
E2	PI2	PR1
E2	PI2	PR3
E5	PI1	PR1
E5	PI1	PR3
E5	PI2	PR1
E5	PI2	PR3
E3	PI4	PR5
E3	PI4	PR6
E3	PI5	PR5
E3	PI5	PR6
E7	PI4	PR5
E7	PI4	PR6
E7	PI5	PR5
E7	PI5	PR6

Descomponemos la tabla en 3 tabla nuevas:

**Proveedor-Pieza, Pieza-Proyecto, Proyecto-Proveedor**

Proveedor-Pieza	
Proveedor	Pieza
E1	PI3
E1	PI6
E4	PI3
E4	PI6
E6	PI3
E6	PI6
E2	PI1
E2	PI2
E5	PI1
E5	PI2
E3	PI4
E3	PI5
E7	PI4
E7	PI5

Pieza-Proyecto	
Pieza	Proyecto
PI3	PR2
PI3	PR4
PI6	PR2
PI6	PR4
PI1	PR1
PI1	PR3
PI2	PR1
PI2	PR3
PI4	PR5
PI4	PR6
PI5	PR5
PI5	PR6

Proyecto-Proveedor	
Proyecto	Proveedor
PR2	E1
PR4	E1
PR2	E4
PR4	E4
PR2	E6
PR4	E6
PR1	E2
PR3	E2
PR1	E5
PR3	E5
PR5	E3
PR6	E3
PR5	E7
PR6	E7



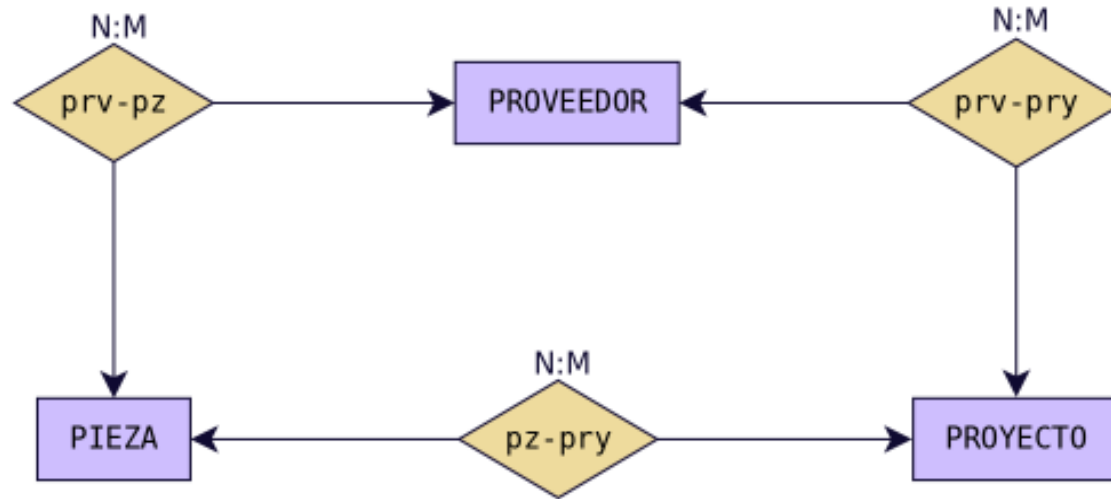
# QUINTA FORMA NORMAL

Pregrado

Ingeniería de  
Sistemas

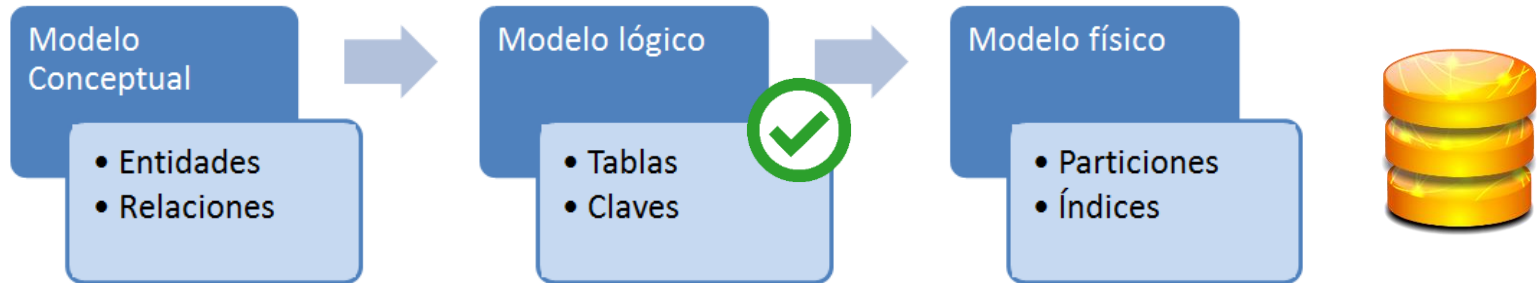
El producto natural de estas 3 tablas nos da la tabla original. Proveedor-Pieza |x| Pieza-Proyecto |x| Proyecto-Proveedor = Suministros

## Diagrama E/R equivalente





Un modelo de datos lógicos describe los datos con el mayor detalle posible, independientemente de cómo se implementarán físicamente en la base de datos.



## CARACTERÍSTICAS

- ✓ Incluye todas las entidades y relaciones entre ellos.
- ✓ Todos los atributos para cada entidad están especificados.
- ✓ La clave principal para cada entidad está especificada.
- ✓ Se especifican las claves externas (claves que identifican la relación entre diferentes entidades).
- ✓ La normalización ocurre en este nivel.



## Los pasos para diseñar el modelo de datos lógicos son los siguientes:

- ✓ Especifique claves primarias para todas las entidades.
- ✓ Encuentra las relaciones entre diferentes entidades.
- ✓ Encuentra todos los atributos para cada entidad.
- ✓ Resuelva las relaciones de muchos a muchos.
- ✓ Normalización.



# DIFERENCIA: MODELO CONCEPTUAL Y LÓGICO

Pregrado

Ingeniería de  
Sistemas

CARACTERÍSTICA	CONCEPTUAL	LÓGICO
Nombres de Entidades	✓	✓
Relaciones de Entidades	✓	✓
Atributos	✓	✓
Claves Primarias		✓
Claves Externas		✓





# TRANSFORMACIÓN: MODELO CONCEPTUAL AL LÓGICO

Pregrado

Ingeniería de  
Sistemas

Previo a la aplicación de las reglas de transformación de esquemas entidad-relación a esquemas relacionales(Lógico) es conveniente la preparación de los esquemas entidad-relación mediante la aplicación de unas reglas que faciliten y garanticen la fiabilidad del proceso de transformación.

Estas reglas preparatorias se basan en la aplicación de la 1FN y su objetivo es eliminar las siguientes anomalías:

- ✓ Atributos con valores múltiples
- ✓ Atributos compuestos

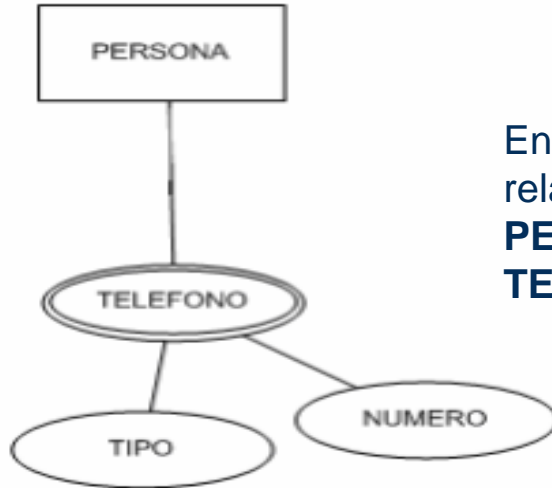


# ELIMINACIÓN: ATRIBUTOS MÚLTIPLES

Pregrado

Ingeniería de  
Sistemas

Todos los atributos múltiples se deben transformar en un tipo de entidad débil por existencia con una relación de **muchos a muchos** o de **uno a muchos**, según sea el caso, con el tipo de entidad sobre el cual estaba definido. Si se considera que la nueva entidad creada resulta ambigua, se le pueden añadir atributos o heredarlos de la otra entidad. Suponemos para el siguiente **ejemplo** que una persona puede tener varios números de teléfono.



En este caso sería necesario expresar el esquema relacional de la forma:

**PERSONA** ( ... lista de atributos ... )

**TELEFONO**(idPersona\*, numero, tipo)



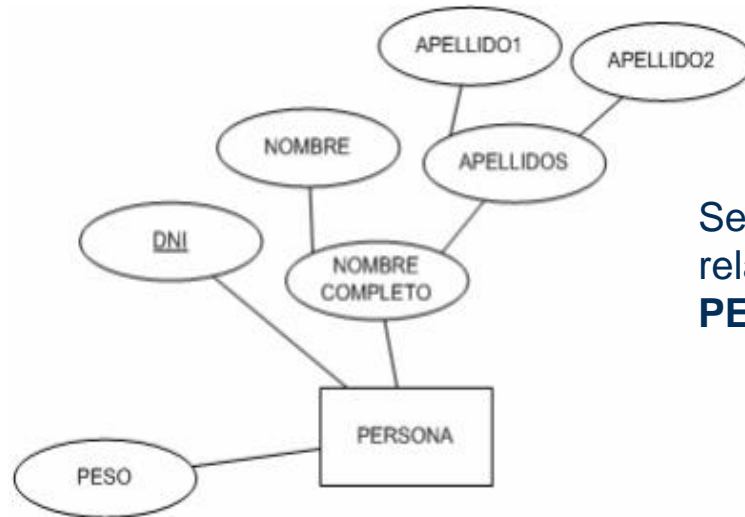
# ELIMINACIÓN: ATRIBUTOS COMPUESTOS

Pregrado

Ingeniería de  
Sistemas

Todos los atributos compuestos **deben ser descompuestos en atributos simples** que quedan asociados a la misma entidad.

El esquema entidad-relación:



Se expresaría mediante el siguiente esquema relacional:

**PERSONA** (DNI, nombre, apellido1, apellido2, peso)



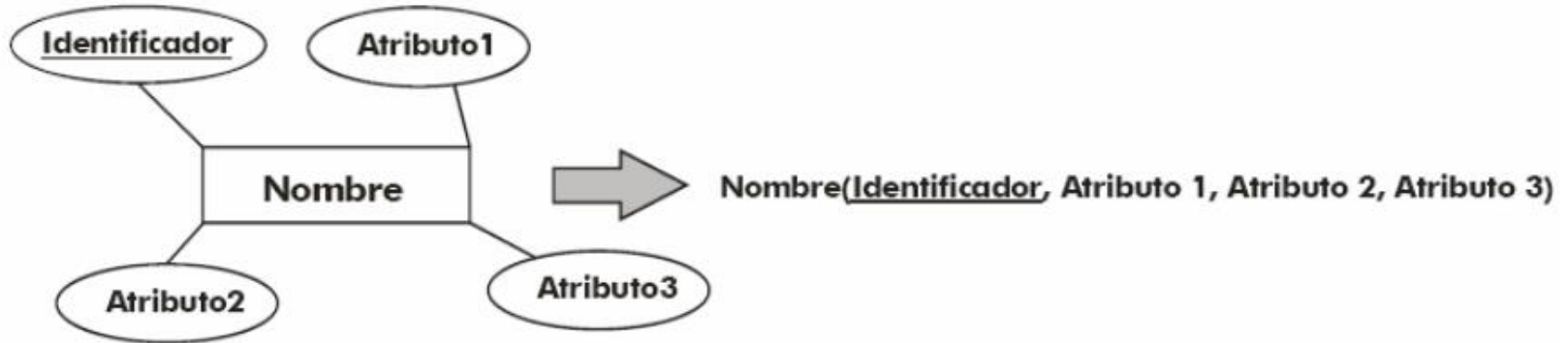
# TRANSFORMACIÓN: ENTIDADES FUERTES

Pregrado

Ingeniería de  
Sistemas

- ✓ **Entidades.** Las entidades pasan a ser tablas.
- ✓ **Atributos.** Los atributos pasan a ser columnas.
- ✓ **Identificadores principales.** Pasan a ser claves primarias.
- ✓ **Identificadores candidatos.** Pasan a ser claves candidatas.

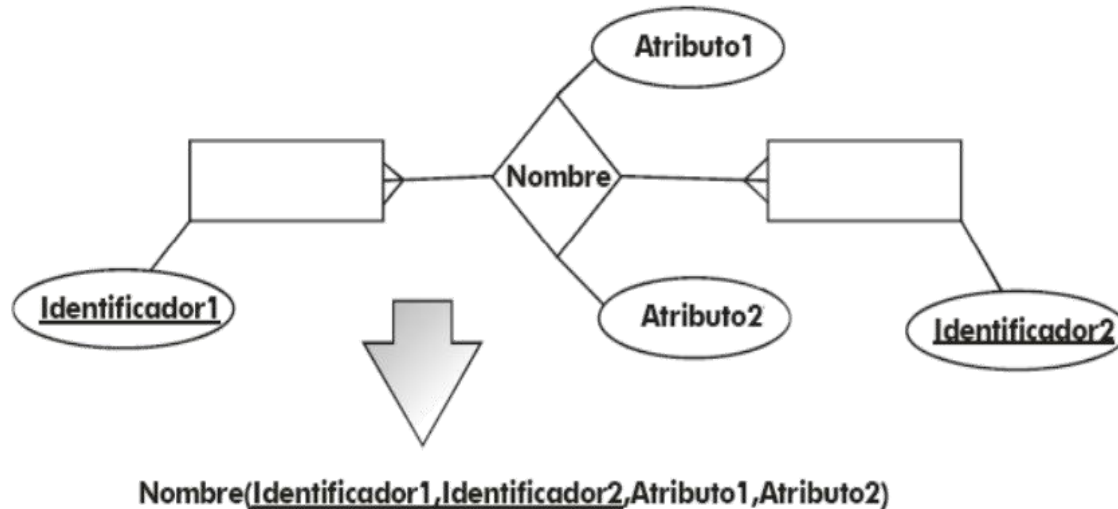
Esto hace que la transformación se produzca según este ejemplo:





## Relaciones varios a varios

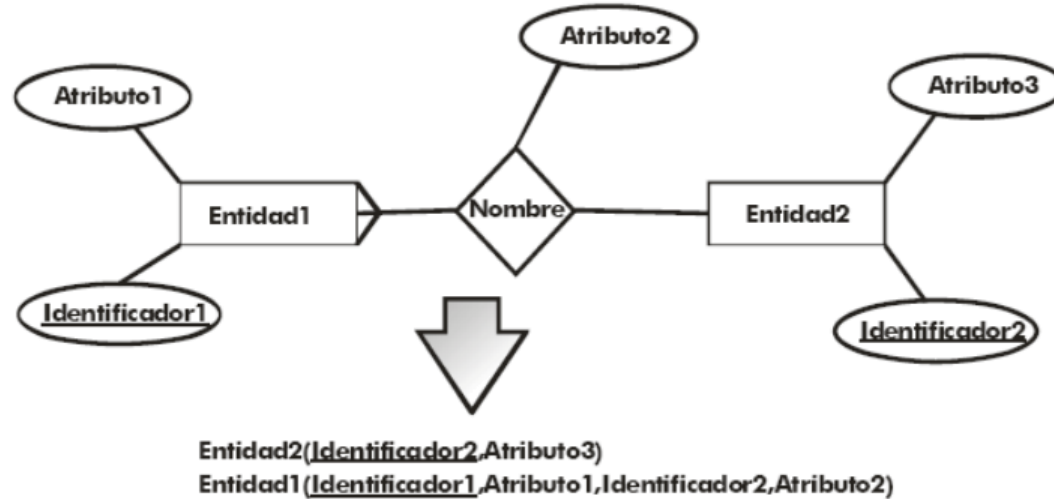
En las relaciones varios a varios la relación se transforma en una tabla cuyos atributos son: los atributos de la relación y las claves de las entidades relacionadas (que pasarán a ser claves externas). La clave de la tabla la forman todas las claves externas.





## Relaciones uno a varios o uno a uno

Las relaciones de tipo uno a varios no requieren ser transformadas en una tabla en el modelo relacional. En su lugar la tabla del lado varios (tabla relacionada) incluye como clave externa el identificador de la entidad del lado uno (tabla principal).



En el caso de que el número mínimo de la relación sea de cero (puede haber ejemplares de la entidad uno sin relacionar), se deberá permitir valores nulos en la clave externa identificador2. En otro caso no se podrán permitir (ya que siempre habrá un valor relacionado).



# TRANSFORMACIÓN: RELACIONES

Pregrado

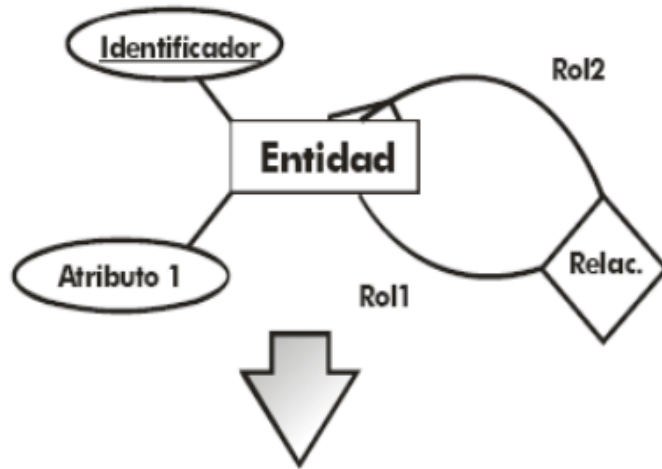
Ingeniería de  
Sistemas

En el caso de las **relaciones uno a uno**, ocurre lo mismo: **la relación no se convierte en tabla**, sino que se coloca en una de las tablas (en principio daría igual cuál) el identificador de la entidad relacionada como clave externa. En el caso de que una entidad participe opcionalmente en la relación, entonces es el identificador de ésta el que se colocará como clave externa en la tabla que representa a la otra entidad.

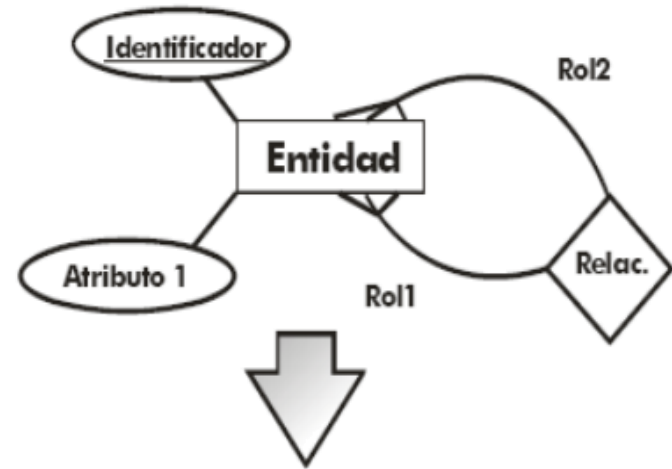


## Relaciones reflexivas

Las relaciones reflexivas o recursivas se tratan de la misma forma que las otras, sólo que un mismo atributo puede figurar dos veces en una tabla como resultado de la transformación.



Entidad(Identificador, Atributo1, Identificador Rol 1)



Entidad(Identificador, Atributo1)

Relac(Identificador Rol 1, Identificador Rol 2, Atributo1)



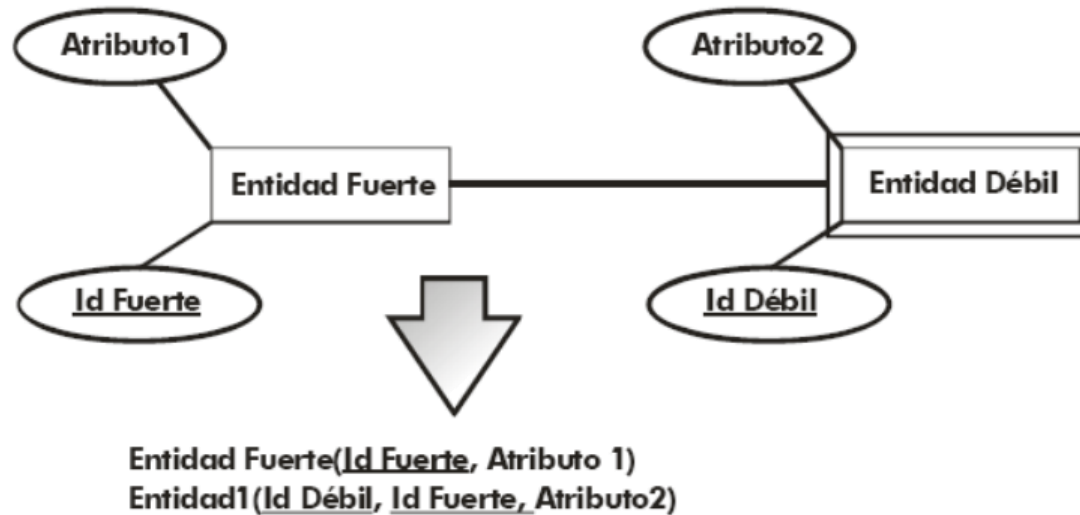


# TRANSFORMACIÓN: ENTIDADES DÉBILES

Pregrado

Ingeniería de  
Sistemas

Toda **entidad débil** incorpora una relación implícita con una entidad fuerte. Esta relación no necesita incorporarse como tabla en el modelo relacional. Sí se necesita incorporar la clave de la entidad fuerte como clave externa en la **entidad débil**. Es más, normalmente esa clave externa forma parte de la clave principal de la tabla que representa a la entidad débil.



En ocasiones el identificador de la **entidad débil** es suficiente para identificar los ejemplares de dicha entidad, entonces ese identificador quedaría como clave principal, pero el identificador de la entidad fuerte seguiría figurando como clave externa en la **entidad débil**.



# MODELO FÍSICO

Pregrado

Ingeniería de  
Sistemas

muestra todas las estructuras de tabla, incluidos el nombre de columna, el tipo de datos de columna, las restricciones de columna, la clave principal, la clave externa y las relaciones entre las tablas.



## CARACTERÍSTICAS

- ✓ Especificación de todas las tablas y columnas.
- ✓ Las claves externas se usan para identificar relaciones entre tablas.
- ✓ La desnormalización puede ocurrir según los requisitos del usuario.



# MODELO FÍSICO

Pregrado

Ingeniería de  
Sistemas

Las consideraciones físicas pueden hacer que el modelo de datos físicos sea bastante diferente del modelo de datos lógicos.

El modelo de datos físicos será diferente para diferentes Sistemas de Gestión de Base de datos. **Por ejemplo**, el tipo de datos para una columna puede ser diferente entre **Oracle y SQL Server**.



**Los pasos básico para el diseño del modelo de datos físicos son los siguientes:**

- ✓ Convertir entidades en tablas.
- ✓ Convertir relaciones en claves externas.
- ✓ Convertir atributos en columnas.
- ✓ Modificar el modelo de datos físicos en función de las restricciones / requisitos físicos.



# DIFERENCIA: MODELO CONCEPTUAL, LÓGICO Y FÍSICO

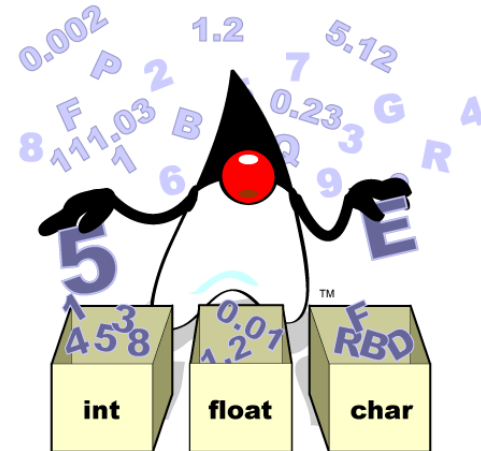
Pregrado

Ingeniería de  
Sistemas

CARACTERÍSTICA	CONCEPTUAL	LÓGICO	FÍSICO
Nombres de Entidades	✓	✓	✓
Relaciones de Entidades	✓	✓	✓
Atributos	✓	✓	✓
Claves Primarias		✓	✓
Claves Externas		✓	✓
Nombres de Tabla			✓
Nombres de Columna			✓
Tipos de datos			✓

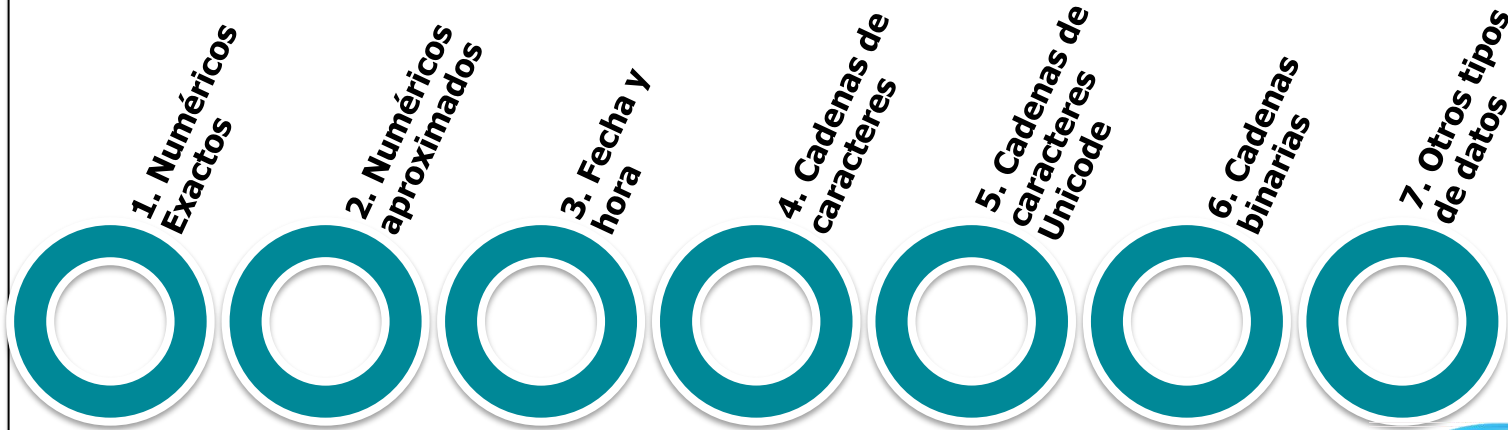
## ¿Qué es un tipo de dato?

Es un atributo de los datos que indica al ordenador (y/o al programador) sobre la clase de datos que se va a trabajar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar.





## ¿Cómo está organizado los tipos de datos?



# TIPOS DE DATOS POR CATEGORÍA



## Numéricos exactos

TIPOS DE DATOS	INTERVALO (DESCRIPCIÓN)
bigint	De $-2^{63}$ (-9.223.372.036.854.775.808) a $2^{63}-1$ (9.223.372.036.854.775.807)
bit	1,0
int	De $-2^{31}$ (-2.147.483.648) a $2^{31}-1$ (2.147.483.647)
money	De -922.337.203.685.477,5808 hasta 922.337.203.685.477,5807
smallmoney	De - 214.748,3648 a 214.748,3647
numeric	De $-10^{38}+1$ y $10^{38}-1$
decimal	De $-10^{38}+1$ y $10^{38}-1$
smallint	De $-2^{15}$ (-32.768) a $2^{15}-1$ (32.767)
tinyint	De 0 a 255

## Numéricos aproximados

TIPOS DE DATOS	INTERVALO (DESCRIPCIÓN)
float	De $-1,79E+308$ a $-2,23E-308$ , 0 y de $2,23E-308$ a $1,79E+308$
real	De $-3,40E+38$ a $-1,18E-38$ , 0 y de $1,18E-38$ a $3,40E+38$

## Fecha y Hora

TIPOS DE DATOS	INTERVALO (DESCRIPCIÓN)
date	De 0001-01-01 a 31.12.99
smalldatetime	De 1900-01-01 a 2079-06-06
datetime	De 01.01.53 a 31.12.99
datetime2	De 0001-01-01 00:00:00.0000000 a 9999-12-31 23:59:59.9999999
datetimeoffset	De 0001-01-01 00:00:00.0000000 a 9999-12-31 23:59:59.9999999 (en UTC)



# TIPOS DE DATOS POR CATEGORÍA



## Cadenas de caracteres

TIPOS DE DATOS	INTERVALO (DESCRIPCIÓN)
char	Datos de cadena no Unicode de longitud <b>fija</b> . Su valor está entre 1 y 8.000
varchar	Datos de cadena no Unicode de longitud <b>variable</b> . Su valor está entre 1 y 8.000
text	Datos no Unicode de longitud <b>variable</b> . Su valor máximo es (2.147.483.647) bytes.

## Cadenas de caracteres Unicode

TIPOS DE DATOS	INTERVALO (DESCRIPCIÓN)
nchar	Datos de cadena Unicode de longitud <b>fija</b> . Su valor está entre 1 y 4.000
nvarchar	Datos de cadena Unicode de longitud <b>variable</b> . Su valor está entre 1 y 4.000
ntext	Datos Unicode de longitud <b>variable</b> . Su valor máximo $2^{30} - 1$ (1.073.741.823) bytes.

## Cadenas binarias

TIPOS DE DATOS	INTERVALO (DESCRIPCIÓN)
binary	Datos binarios de longitud <b>fija</b> . Su valor oscila entre 1 y 8.000
varbinary	Datos binarios de longitud <b>variable</b> . Su valor oscila entre 1 y 8.000
image	Datos binarios de longitud <b>variable</b> . Desde 0 hasta $2^{31}-1$ (2.147.483.647) bytes.

## Otros tipos de datos

cursor	timestamp
hierarchyid	uniqueidentifier
sql_variant	xml
tabla	Tipos espaciales

### IMPORTANTE:

Los tipos de datos **ntext**, **text** e **image** se quitarán en una versión futura de Microsoft SQL Server. Evite su uso en nuevos trabajos de desarrollo y piense en modificar las aplicaciones que los usan actualmente. Utilice **nvarchar(max)**, **varchar(max)** y **varbinary(max)** en su lugar.



Universidad **César Vallejo**

Licenciada por Sunedu  
para que puedas salir adelante