

# Construcción de software y toma de decisiones

## TC2005B

**Dr. Esteban Castillo Juarez**

ITESM, Campus Santa Fe



esteban.castillojz@tec.mx

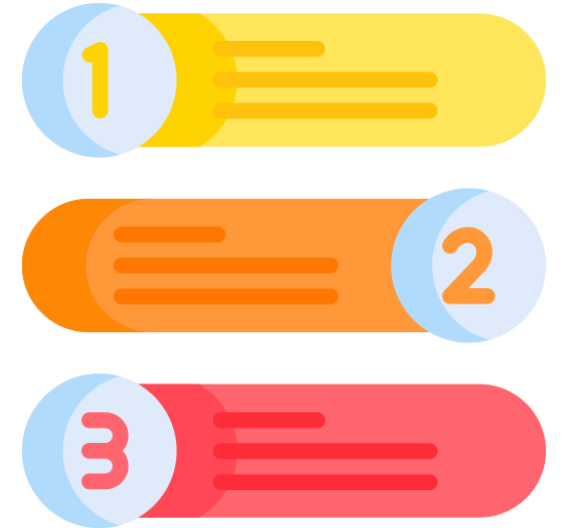
# Agenda

- Diagramas entidad-relación
- Ejemplos de diagramas entidad-relación
- Normalización tablas
- SCRUM para el reto

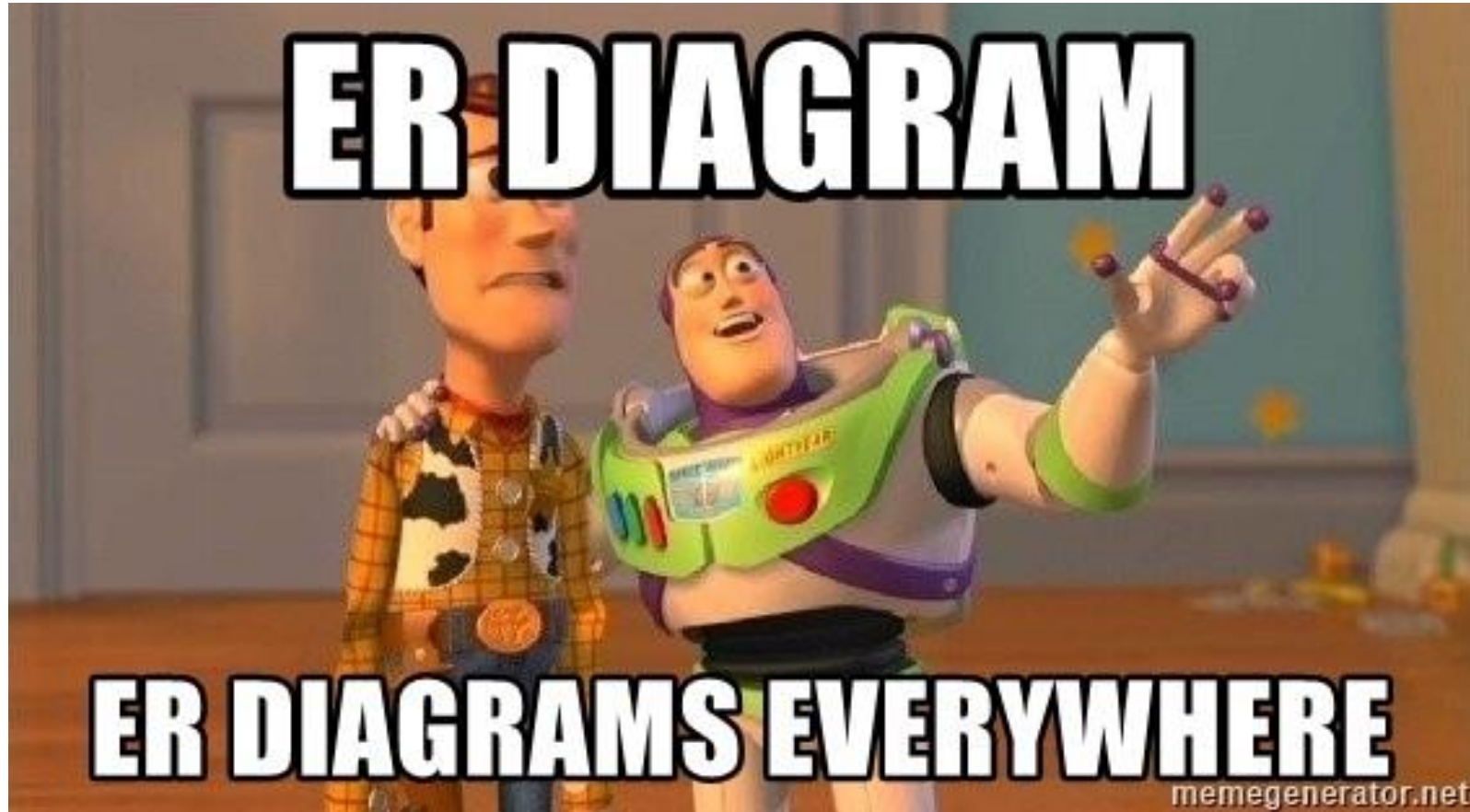
# Diagramas entidad-relación

A nivel teórico, los pasos a seguir a la hora de crear una base de datos son los siguientes:

1. **Recopilación de toda la información del cliente** para saber sus necesidades. En este caso dependemos de SCRUM con las historias de usuario y los diagramas de casos de uso de UML.
2. **Diseño del modelo entidad-relación** (modelo conceptual). Consiste en plasmar en un diagrama las **entidades**, **atributos** y **relaciones/líneas-de-comunicación** definidas en los requerimientos recogidos en la primera fase.



# Diagramas entidad-relación



# Diagramas entidad-relación

A nivel teórico, los pasos a seguir a la hora de crear una base de datos son los siguientes:

3. **Transformación del diagrama de entidad-relación en el modelo relacional**, basado en la creación de tablas genéricas.
4. **Toma de decisión** sobre qué Sistema de Gestión de Bases de Datos (SGBD) utilizaremos. Algunos de los más utilizados son Oracle y MySQL.
5. **Implementación del modelo relacional** en el programa que gestiona la base de datos, es decir, el modelo físico.



# Diagramas entidad-relación

Los elementos básicos de un modelo entidad-relación son las entidades, las relaciones, los atributos y las cardinalidades.



Nota: UML, permite ser flexible en como crear las relaciones, ya sea con un rombo o un rectángulo. De igual modo, también es flexible en como poner las cardinalidades.

# Diagramas entidad-relación

Elementos a considerar para crear una entidad

1. Una entidad es un objeto del mundo real sobre el que queremos almacenar información (Ej: una persona).
2. Las entidades están compuestas de atributos que son los datos que definen el objeto (para la entidad persona serían CURP, nombre, apellidos, dirección,...).
3. De entre los atributos habrá uno o un conjunto de ellos que no se repite; a este atributo o conjunto de atributos se le llama llave primaria, (para la entidad persona una clave seria CURP).



# Diagramas entidad-relación

Elementos a considerar para crear una entidad

4. En toda entidad siempre hay al menos una llave que en el peor de los casos estará formada por todos los atributos de la tabla. Ya que pueden haber varias llaves y necesitamos elegir una, lo haremos atendiendo a estas normas:
- Que sea única.
  - Que se tenga pleno conocimiento de ella.- ¿Por qué en las empresas se asigna a cada cliente un número de cliente?
  - Que sea mínima, ya que será muy utilizada por el gestor de base de datos.

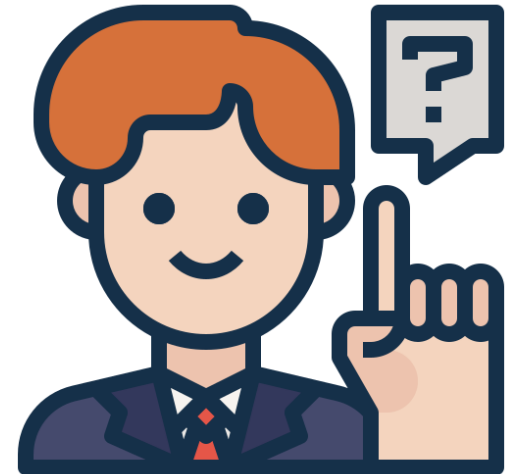




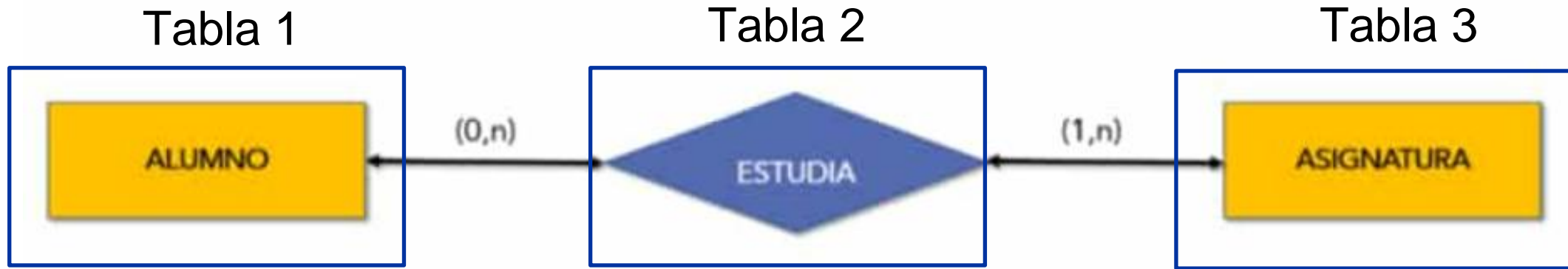
# Diagramas entidad-relación

## Elementos a considerar para crear una relación

1. Una relación es una asociación entre entidades, sin existencia propia en el mundo real que estamos modelando, pero necesaria para reflejar las interacciones existentes entre entidades.
2. **Las relaciones creadas pueden ser por si mismas tablas que se tengan que modelar en la base de datos.**
3. Las relaciones pueden ser de tres tipos según su grado de cardinalidad:
  - Relaciones 1-1
  - Relaciones 1-n
  - Relaciones n-n



# Diagramas entidad-relación



Las relaciones creadas PUEDEN ser tablas que tengan que modelar en la base de datos.

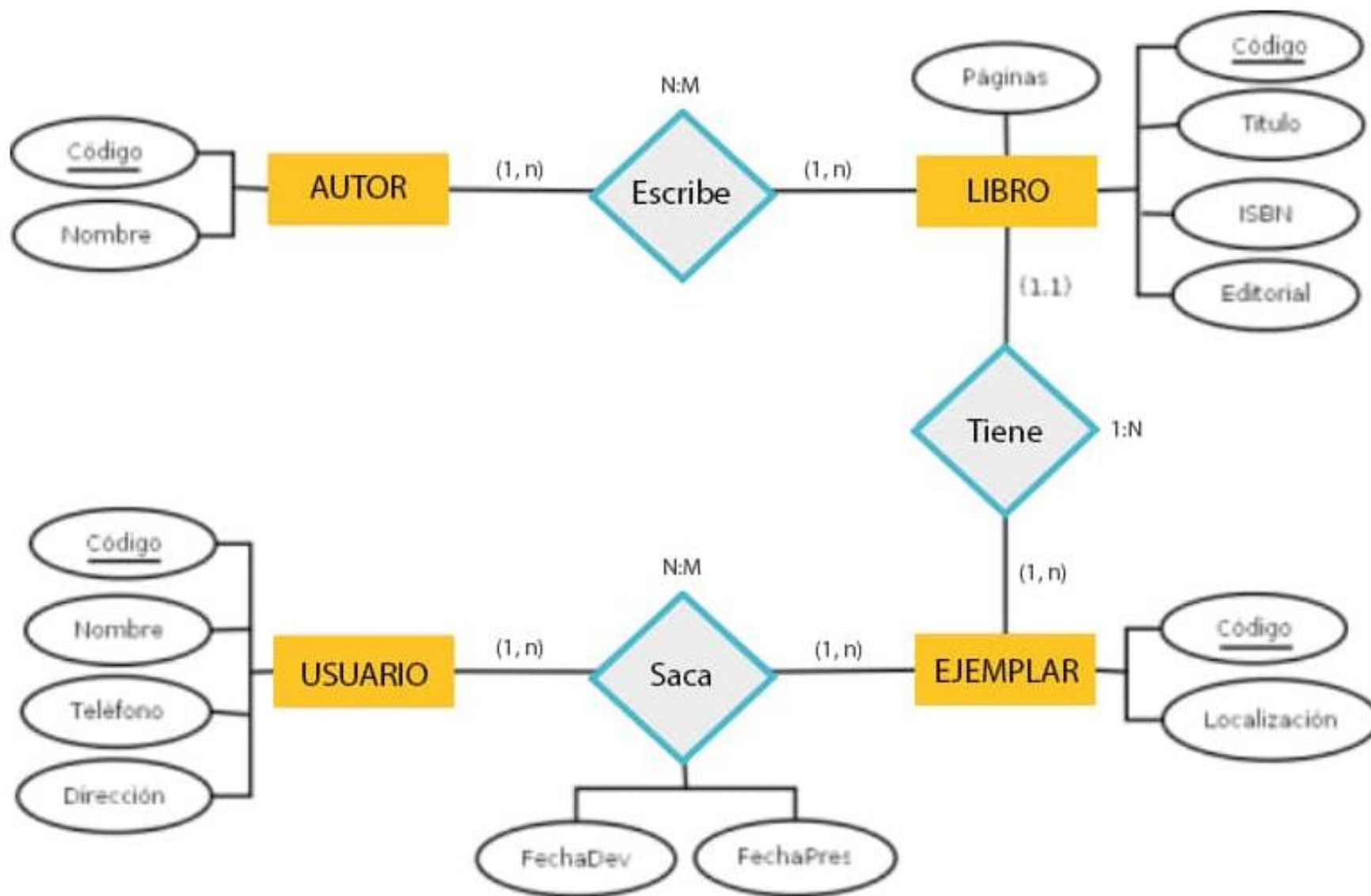
# Diagramas entidad-relación

Elementos a considerar para crear una relación con cardinalidad

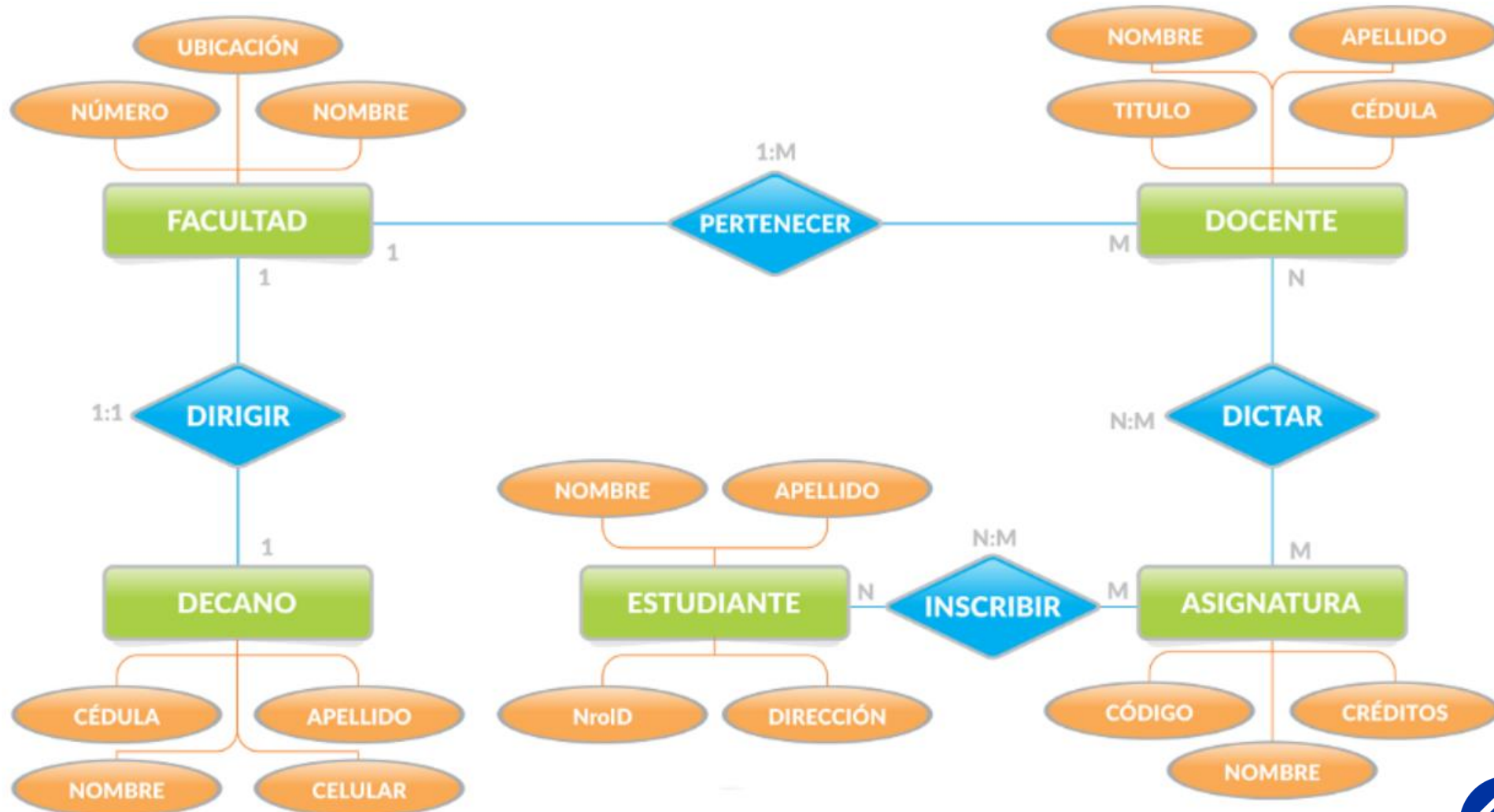
3. Las relaciones pueden ser de tres tipos:

- **Relaciones 1-1:** Las entidades que intervienen en la relación se asocian una a una (Ej: la entidad HOMBRE, la entidad MUJER y entre ellos la relación MATRIMONIO).
- **Relaciones 1-n:** Una ocurrencia de una entidad está asociada con muchas (n) de otra (Ej: la entidad EMPRESA, la entidad TRABAJADOR y entre ellos la relación TRABAJAR-EN).
- **Relaciones n-n:** Cada ocurrencia, en cualquiera de las dos entidades de la relación, puede estar asociada con muchas (n) de la otra y viceversa (Ej: la entidad ALUMNO, la entidad EMPRESA y entre ellos la relación MATRÍCULA).

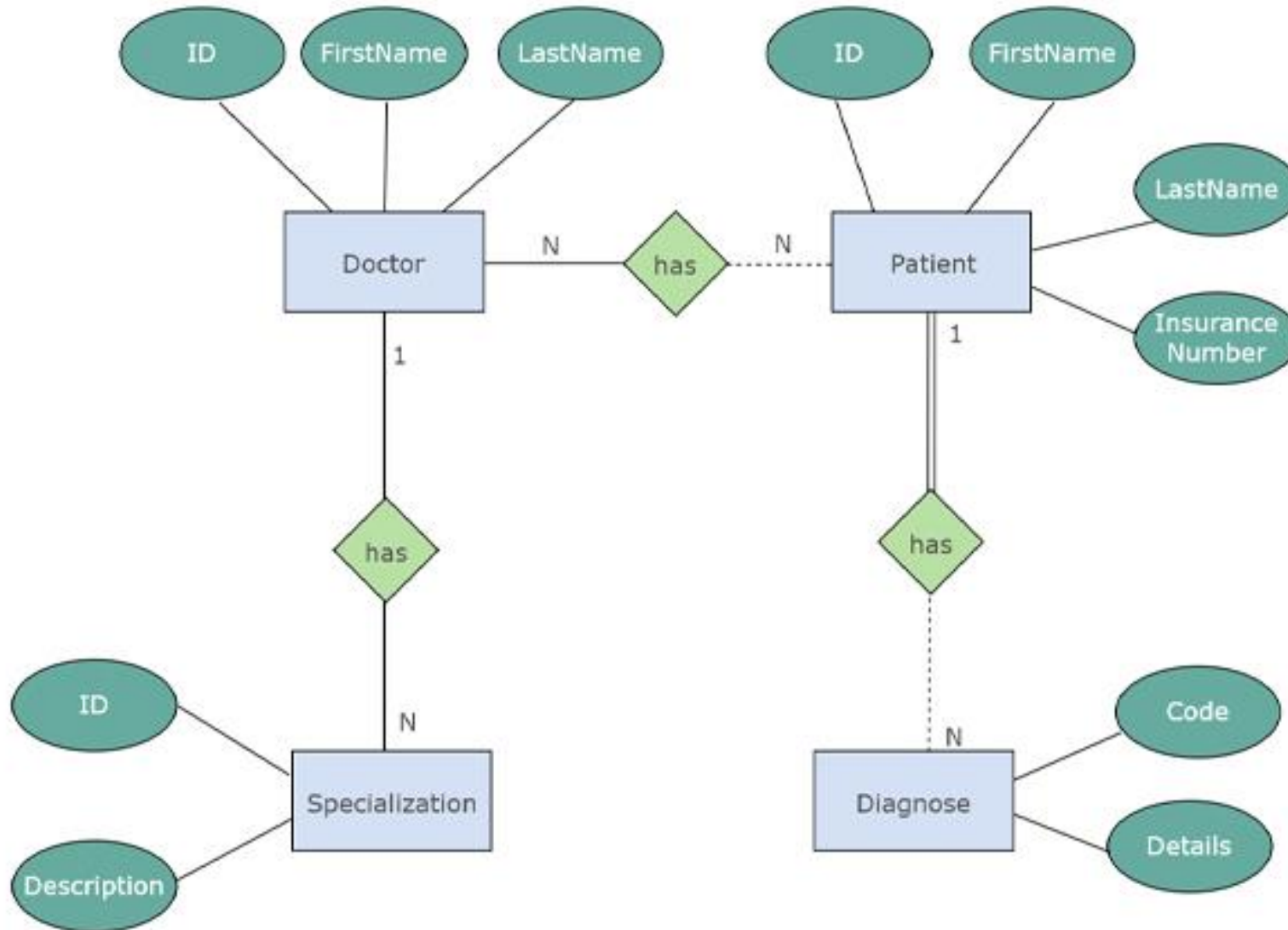
# Ejemplos de diagramas entidad-relación



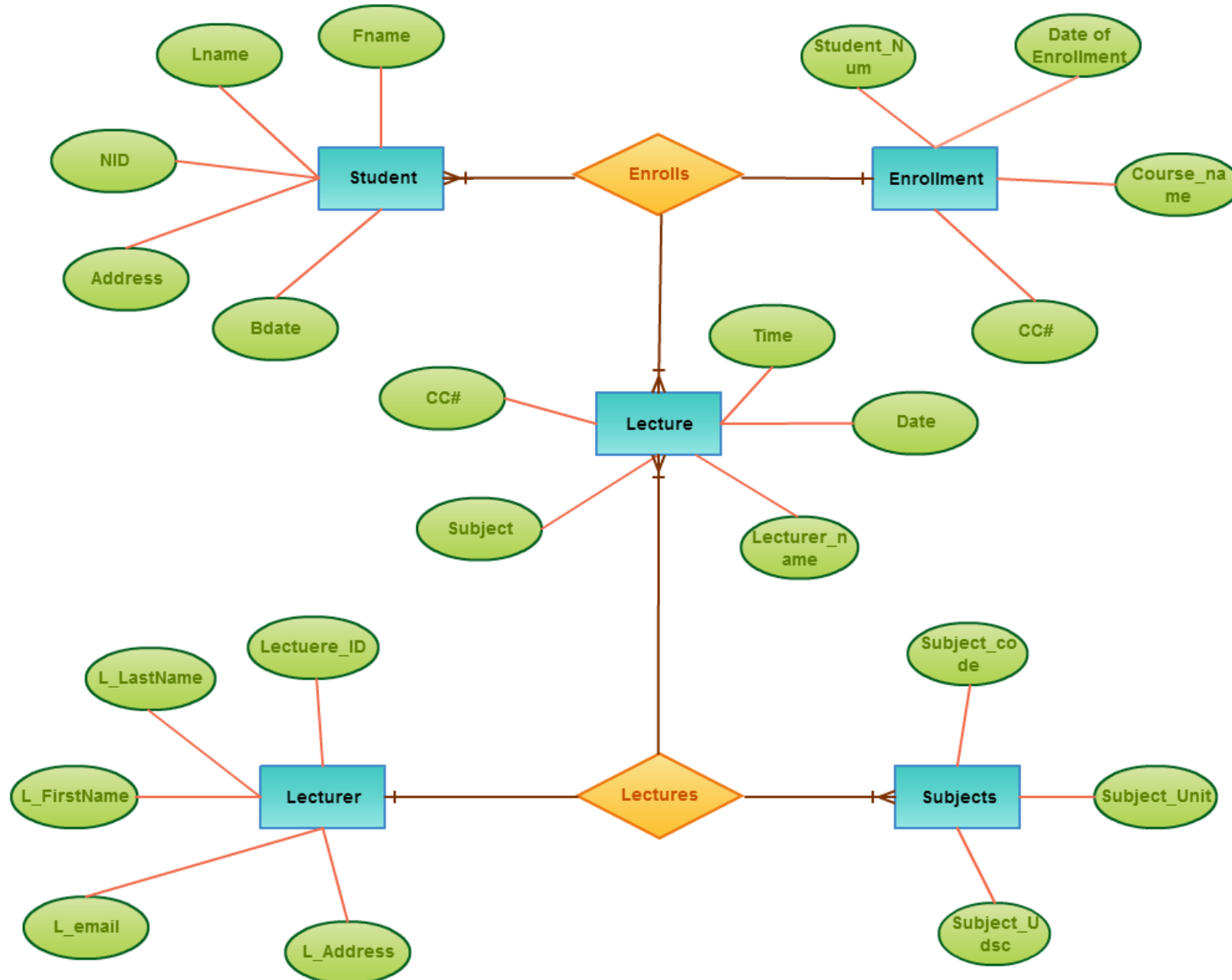
# Ejemplos de diagramas entidad-relación



# Ejemplos de diagramas entidad-relación

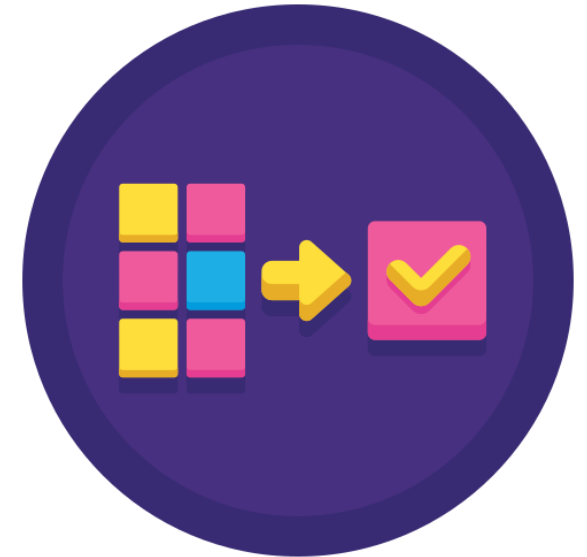


# Ejemplos de diagramas entidad-relación



# Normalización tablas

- La normalización de los datos es un concepto importante al diseñar conceptualmente el modelo entidad-relación.
- El objetivo principal de la normalización es la de reducir la redundancia de datos y mejorar la integridad de la información a almacenar.
- La normalización también facilita el proceso de rediseño y ampliación de la estructura de la base de datos.



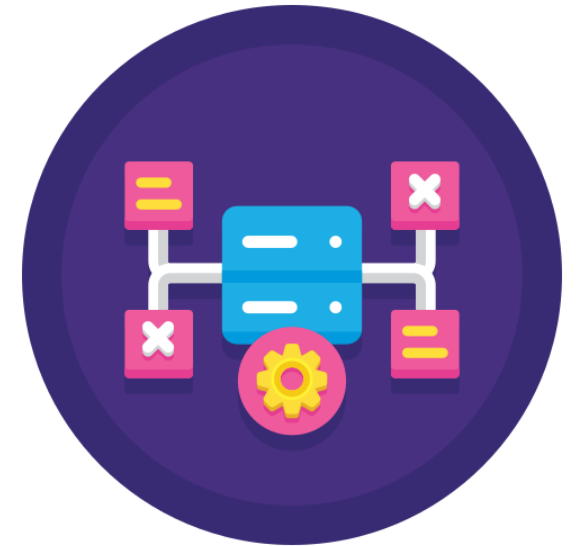


# Normalización tablas



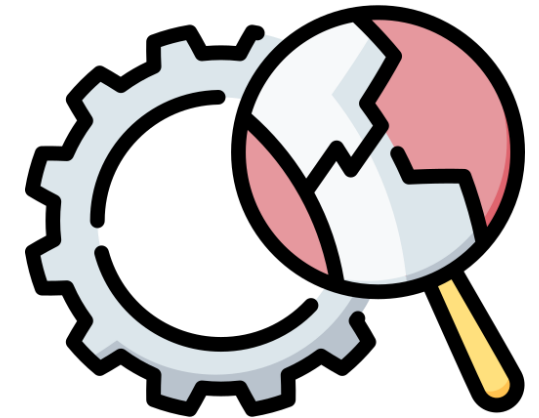
# Normalización tablas

- Oficialmente, hay seis formas normales, **pero la mayoría de los arquitectos de bases de datos solo se ocupan de las tres primeras formas.**
- Esto se debe a que el proceso de normalización es progresivo y no podemos lograr un nivel más alto de normalización de la base de datos a menos que se hayan satisfecho los niveles anteriores.
- Adiciónateme, el uso de las seis formas normales restringe demasiado el modelo de la base de datos y, en general, se vuelven muy complejas de implementar.



# Normalización tablas

- En resumen, las formas normales son conjuntos de criterios que utilizamos para mejorar la estructura de las bases de datos.
- La normalización es la transformación de los datos a un juego de estructuras de datos más pequeñas y estables.
- La normalización reduce la redundancia de datos y simplifica las dependencias entre columnas.



# Normalización tablas

## 1FN – Primera Forma Normal

Una tabla (o entidad) está en la primera forma normal si:

1. Todos los atributos son «atómicos». Por ejemplo, en un campo teléfono no tenemos varios números telefónicos.
2. La tabla contiene una llave primaria única. Por ejemplo el CURP para personas, la matrícula para vehículos o un simple id autoincremental. Si no tiene una llave, no es 1FN.



# Normalización tablas

## 1FN – Primera Forma Normal

Una tabla (o entidad) está en la primera forma normal si:

3. La llave primaria no contiene atributos nulos. No podemos tener filas para las que no haya un índice específico (por ejemplo, personas sin CURP o vehículos sin matrícula).
4. No debe existir variación en el número de columnas. Si algunas filas tienen 8 columnas y otras 3, pues no estamos en 1FN.



# Normalización tablas

## 1FN – Primera Forma Normal

Una tabla (o entidad) está en la primera forma normal si:

5. Los campos no llave deben identificarse por la llave. Es decir, que los campos simples dependen funcionalmente de la llave primaria para ser inequívocamente representados.
6. Todos los atributos o valores almacenados en las columnas, deben ser indivisibles.
7. No deben existir grupos de valores repetidos.



# Normalización tablas

## 1FN – Primera Forma Normal

Supongamos que tienes en una tabla una columna Dirección para almacenar la dirección completa, dato que se compondría del nombre de la calle, el número exterior, el número interior (puerta), el código postal, el estado y la capital.

id	Nombre	Dirección	Teléfono	URL
1	Anaya	J:l: Luca	92199932	Anaya.com
2	Pericles	C/Luna # 20-28018 Tlaxcala	99299492	Pericles.com



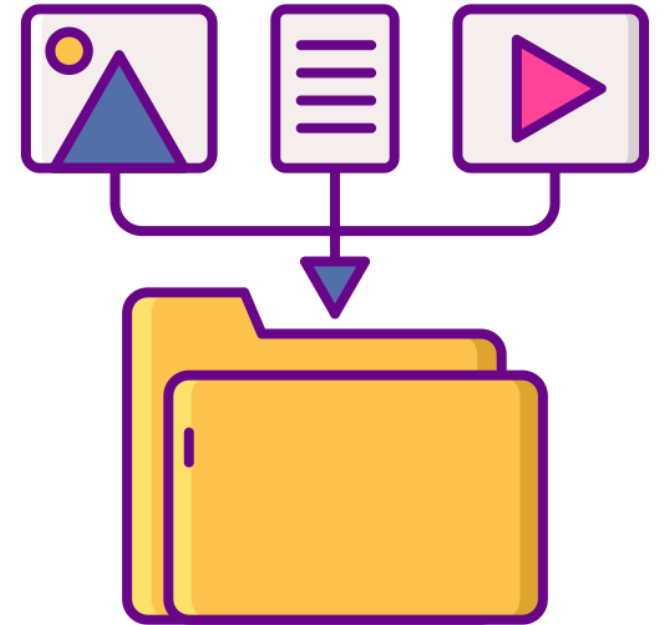
Calle	Número	Puerta	CP	Población	Provincia
Luna	20		28018	Tlaxcala	Tlaxcala

# Normalización tablas

## 1FN – Primera Forma Normal

- Una tabla con la estructura anterior plantea problemas a la hora de recuperar información.
- Imaginemos que necesitamos conocer todas las entradas correspondientes a una determinada población, o que queremos buscar a partir del código postal. Al ser la dirección completa una secuencia de caracteres de estructura libre no resultaría nada fácil.

Si separamos la dirección, existirán más columnas, pero cada una de ellas contendrá un valor simple e indivisible que facilitará la realización de operaciones.





# Normalización tablas

## 1FN – Primera Forma Normal


En cuanto al ultimo punto (el numero 7), se debe evitar la repetición de los datos de la población y provincia en cada una de las filas. Siempre que al muestrear la información de una tabla aparezcan datos repetidos, existe la posibilidad de crear una tabla independiente con ellos.

id	Nombre	calle	Número	Puerta	CP	Estado	Capital	Teléfono	URL
1	Anaya	Jl: Luca	15	2	28917	Tepic	Nayarit	93488345	Anaya.com
2	Pericles	Luna	20		28120	San Blas	Nayarit	88238188	Pericles.com
3	Mieres	Tajin	12	1	28120	San Blas	Nayarit	94989982	Mieres.es

# Normalización tablas

## 1FN – Primera Forma Normal

id	Nombre	calle	Número	Puerta	CP	Estado	Capital	Teléfono	URL
1	Anaya	Jl: Luca	15	2	28917	Tepic	Nayarit	93488345	Anaya.com
2	Pericles	Luna	20		28120	San Blas	Nayarit	88238188	Pericles.com
3	Mieres	Tajin	12	1	28120	San Blas	Nayarit	94989982	Mieres.es



Estado		CP	PK
CP	Estado	Capital	
28917	Tepic	Nayarit	
28120	San Blas	Nayarit	
23009	Jean	Jaen	

# Normalización tablas

## 2FN – Segunda Forma Normal

Una tabla está en 2FN si además de estar en 1FN cumple con lo siguiente:

- Se añade la necesidad de que no existan dependencias funcionales parciales.
- Lo anterior significa que todos los valores de las columnas de una fila deben depender de la llave primaria de dicha fila, entendiendo por llave primaria a los valores de todas las columnas que la formen, en caso de ser más de una.
- Las tablas que están ajustadas a la primera forma normal, y además disponen de una clave primaria formada por una única columna con un valor indivisible, cumplen ya con la segunda forma normal.

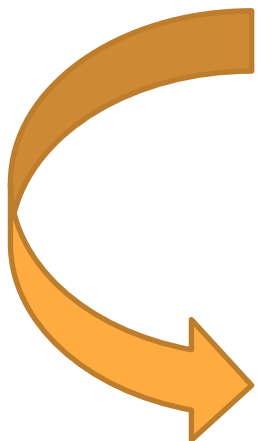
# Normalización tablas

## 2FN – Segunda Forma Normal

Llave primaria

No depende enteramente de código de área

<u>C TRABAJADOR</u>	<u>C AREA</u>	<u>N TRABAJADOR</u>	Q_HORAS
c0001	a0023	Ernesto	45
c0002	a0034	Jhonatan	26
c0003	a0019	Antonio	68



<u>C TRABAJADOR</u>	<u>C AREA</u>	Q_HORAS
c0001	a0023	45
c0002	a0034	26
c0003	a0019	68

<u>C TRABAJADOR</u>	N TRABAJADOR
c0001	Ernesto
c0002	Jhonatan
c0003	Antonio

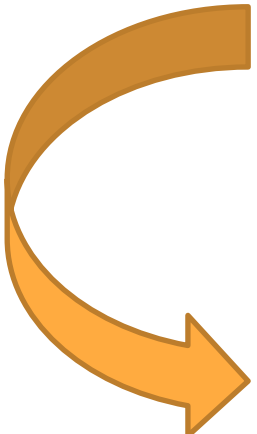
# Normalización tablas

## 2FN – Segunda Forma Normal

Llave primaria

No dependen enteramente de la llave

<u>C_CURSO</u>	<u>C_ALUMNO</u>	Q_NOTA	N_CURSO	N_ALUMNO
c0001	a0023	45	contabilidad	carlos
c0002	a0034	26	estadística	roberto
c0003	a0019	68	álgebra	frank



<u>C_CURSO</u>	N_CURSO
c0001	contabilidad
c0002	estadística
c0003	álgebra

<u>C_ALUMNO</u>	N_ALUMNO
a0023	carlos
a0034	roberto
a0019	frank

<u>C_CURSO</u>	<u>C_ALUMNO</u>	Q_NOTA
c0001	a0023	45
c0002	a0034	26
c0003	a0019	68

# Normalización tablas

## 2FN – Segunda Forma Normal

Llave primaria

<u>C_CIUDDAD</u>	<u>C_COMPTRANSPORTE</u>	Q_VEHICULOS	Q_CHOFERES
c0001	t0345	23	30
c0002	t0741	27	32
c0003	t0184	25	33

Llave primaria

<u>C_PROFESOR</u>	<u>C_AULA</u>	Q_PROMEDIO	Q_ALUMNOS
PCSIJCOR	SI204	13	37
PCSIJSIA	SW223	15	38
PCSIGALV	SI104	14	39

**Ambas están en segunda forma normal**

# Normalización tablas

## 3FN – Tercera Forma Normal

Una tabla está en 3FN si además de estar en 1FN y 2FN cumple con lo siguiente:

- En cuanto a la tercera forma normal, ésta indica que no deben existir dependencias **transitivas** entre las columnas de una tabla.
- Lo cual significa que las columnas que no forman parte de la llave primaria deben depender sólo de esa llave, nunca de otra columna no llave.
- Es importante mencionar que una dependencia funcional transitiva es cuando el cambio de una columna que no es clave puede hacer que cambie cualquier otra columna que no sea clave.



# Normalización tablas

## 3FN – Tercera Forma Normal

Llave primaria

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 <sup>rd</sup> Street 34	Mr.
3	Robert Phil	5 <sup>th</sup> Avenue	Mr.

Cambio en el nombre

May Change  
Salutation

La dependencia transitiva es:  
MEMBERSHIP ID → FULL NAMES  
FULL NAMES → SALUTATTION

La tabla no esta en 3FN



# Normalización tablas

Llave primaria

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3 <sup>rd</sup> Street 34	1
3	Robert Phil	5 <sup>th</sup> Avenue	1

**3FN – Tercera Forma Normal**

TABLA 1

Llave primaria

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Las tablas estan en 3FN

Tabla 2

Llave primaria

SALUTATION ID	SALUTATION
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

Tabla 3

# Normalización tablas

## 3FN – Tercera Forma Normal

Llave primaria

ID	Nombre	Puesto	Salario
1	Juan Pérez	Jefe de área	3000
2	José Sánchez	Administrativo	1500
3	Ana Díaz	Administrativo	1500

Las dependencias transitivas son:

ID → Puesto

Puesto → Salario

La tabla no esta en 3FN

# Normalización tablas

## 3FN – Tercera Forma Normal

Llave primaria

Puesto	Salario
Jefe de área	3000
Administrativo	1500

La tabla esta en 3FN

# Normalización tablas

## 3FN – Tercera Forma Normal

Llave primaria

Llave secundaria

ID	Nombre	Puesto
1	Juan Pérez	Jefe de área
2	José Sánchez	Administrativo
3	Ana Díaz	Administrativo

La tabla esta en 3FN

# Normalización tablas

## 1FN, 2FN y 3FN – ejemplo

- Para aclarar estos conceptos, veamos un ejemplo de normalización con una tabla students ficticia.

Student#	Advisor	Adv-Room	Class1	Class2	Class3
1022	Jones	412	101-07	143-01	159-02
4123	Smith	216	201-01	211-02	214-01

↓  
Llave primaria

# Normalización tablas

## 1FN, 2FN y 3FN – ejemplo

### Primera forma normal: sin grupos repetidos

- Las tablas deben tener una sola columna para cada atributo.
- Dado que un estudiante tiene varias clases, estas clases deben enumerarse en una tabla separada.
- Las columnas Class1, Class2 y Class3 en nuestra tabla no normalizada son indicaciones de problemas de diseño.

# Normalización tablas

## 1FN, 2FN y 3FN – ejemplo

### Primera forma normal: sin grupos repetidos

- Las hojas de cálculo a menudo tienen varias columnas para el mismo atributo (por ejemplo, dirección1, dirección2, dirección3), pero las tablas no deberían.
- Aquí hay otra forma de ver este problema: con una relación de uno a muchos, no coloque el lado uno y el lado muchos en la misma tabla. En su lugar, cree otra tabla en la primera forma normal eliminando el grupo repetitivo, por ejemplo, con Class#, como se muestra en la siguiente diapositiva:

# Normalización tablas

## 1FN, 2FN y 3FN – ejemplo

### Primera forma normal: sin grupos repetidos

Student#	Advisor	Adv-Room	Class#
1022	Jones	412	101-07
1022	Jones	412	143-01
1022	Jones	412	159-02
4123	Smith	216	201-01
4123	Smith	216	211-02
4123	Smith	216	214-01

Llave primaria



# Normalización tablas

## 1FN, 2FN y 3FN – ejemplo

### Segunda forma normal: Eliminación de datos redundantes

- Tomando en cuenta los múltiples valores del atributo `Class#` para cada valor `Student#` en la tabla anterior.
- `Class#` no depende funcionalmente de `Student#` (la llave principal), por lo que esta tabla no está en la segunda forma normal.
- Las siguientes dos tablas muestran la conversión a la segunda forma normal.

# Normalización tablas

## 1FN, 2FN y 3FN – ejemplo

### Segunda forma normal: Eliminación de datos redundantes

Tabla student

Student#	Advisor	Adv-Room
1022	Jones	412
4123	Smith	216

Llave primaria

Tabla Registration

Student#	Class#
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

Llave primaria

# Normalización tablas

## 1FN, 2FN y 3FN – ejemplo

### Tercera forma normal: Eliminación datos que no dependen de una llave

- En el ejemplo anterior, Adv-Room (el número de la oficina del asesor) depende funcionalmente del atributo Advisor.
- La solución es mover este atributo de la tabla students a una tabla de faculty, como se muestra en la siguiente diapositiva.

# Normalización tablas

## 1FN, 2FN y 3FN – ejemplo

### Tercera forma normal: Eliminación datos que no dependen de una llave

Tabla students

Student#	Advisor
1022	Jones
4123	Smith



Llave primaria

Tabla faculty

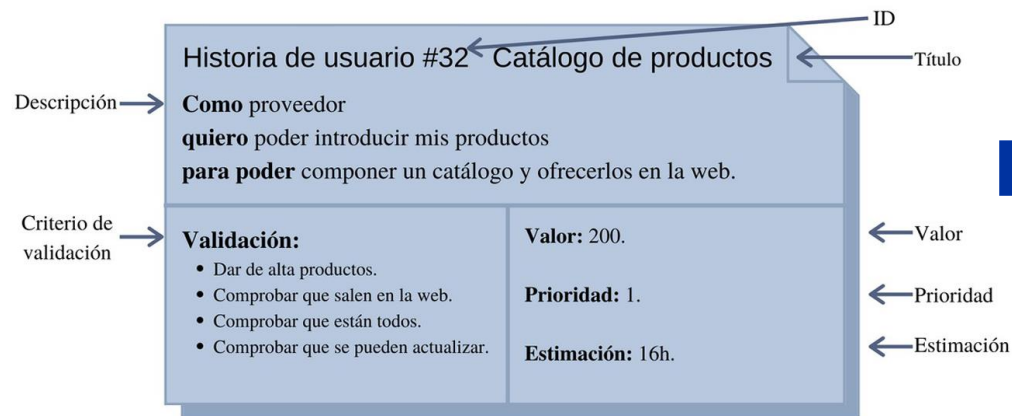
Name	Room	Dept
Jones	412	42
Smith	216	42

# Normalización tablas



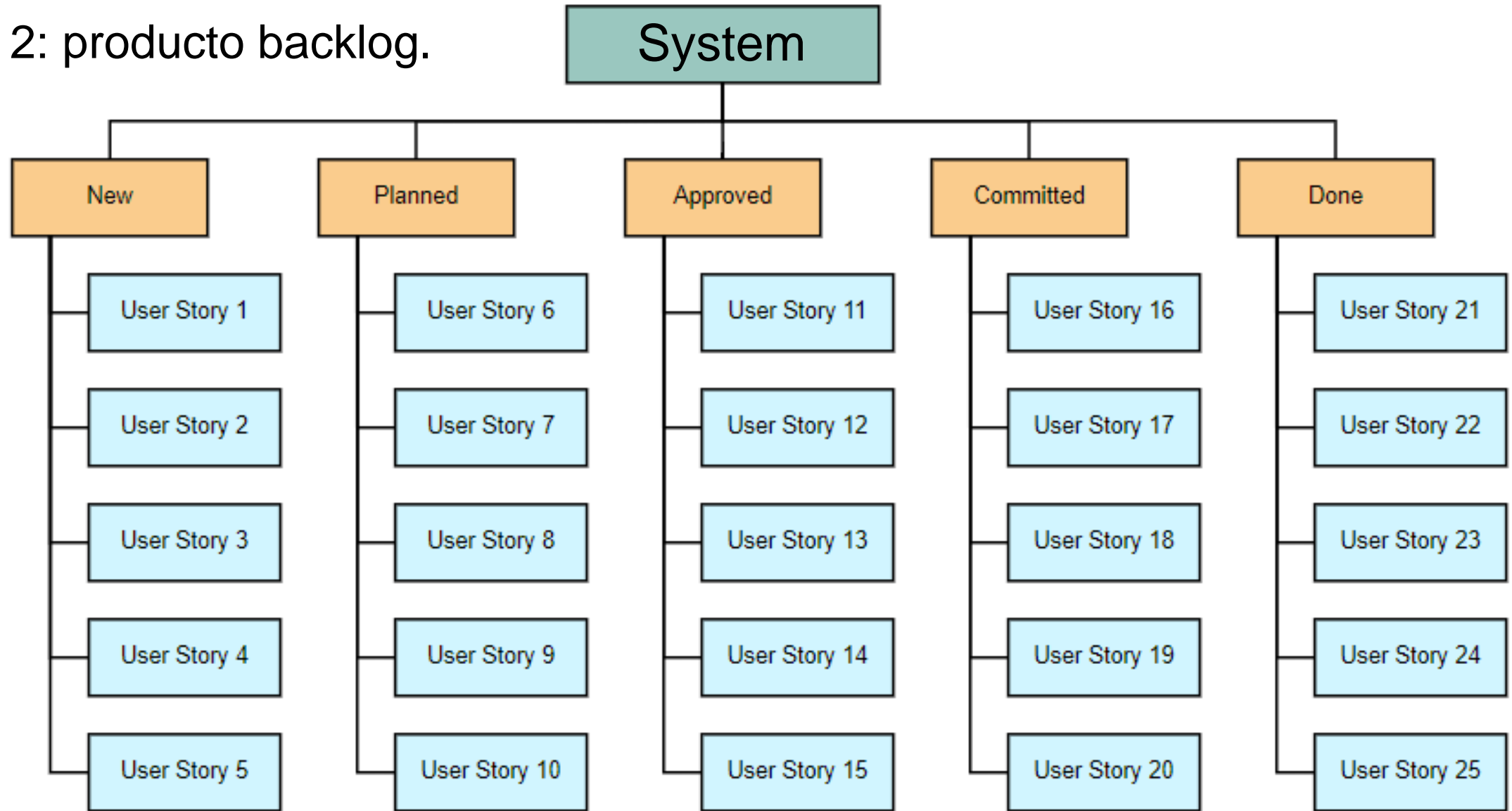
# SCRUM para el reto

Paso 1: se recolectan las historias de usuario del socio formador.



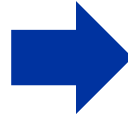
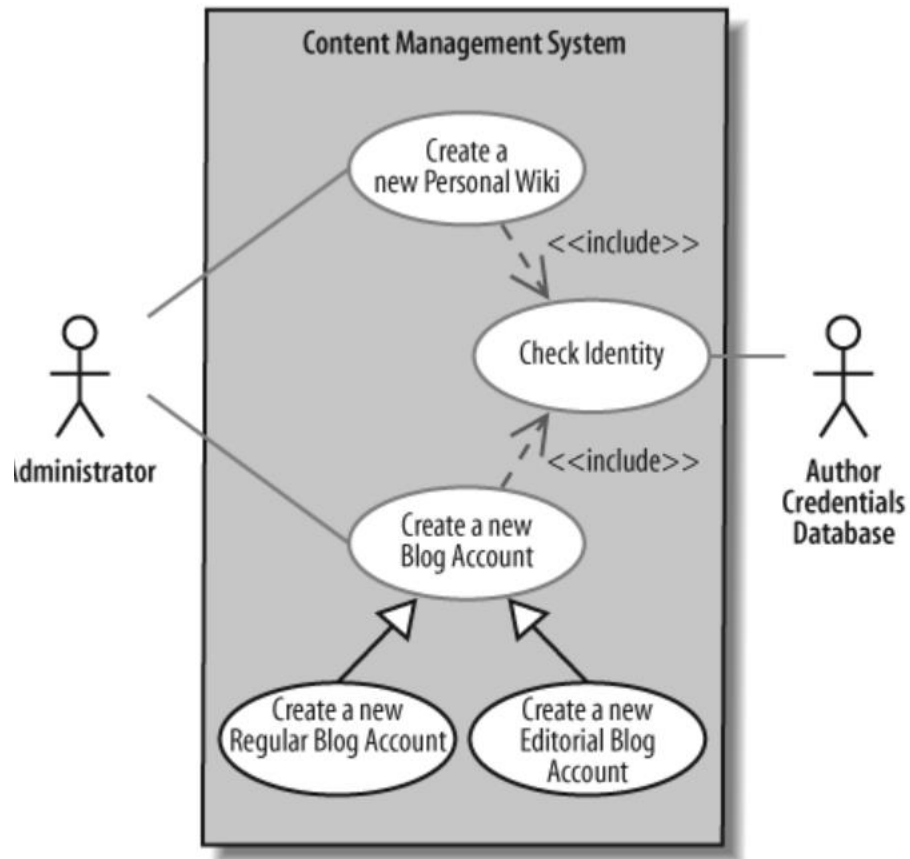
# SCRUM para el reto

Paso 2: producto backlog.



# SCRUM para el reto

Paso 3: se refinan las historias de usuario con casos de uso.

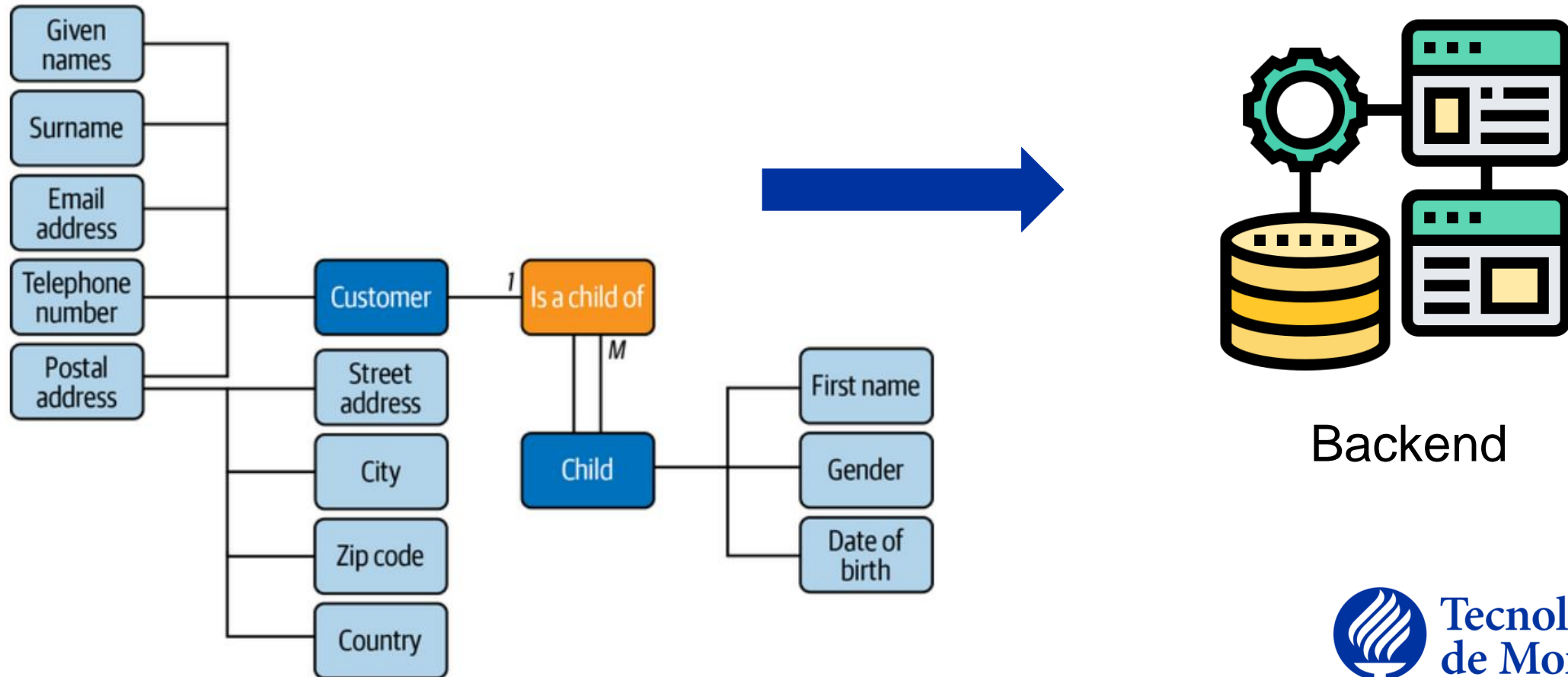


Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects an account type.
	3	The Administrator enters the author's details.
	4 include::Check Identity	The author's details are checked.
	5	The new account is created.
	6	A summary of the new blog account's details are emailed to the author.



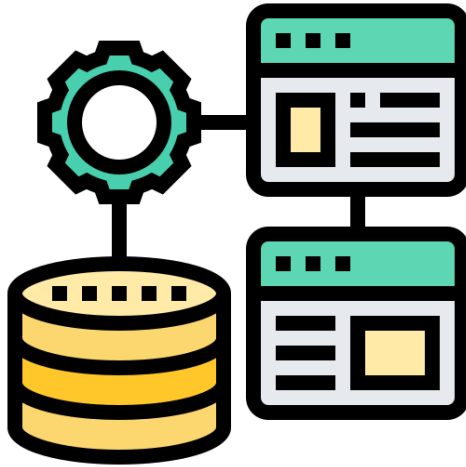
# SCRUM para el reto

Paso 4: se crean las tablas en el motor de bases de datos.

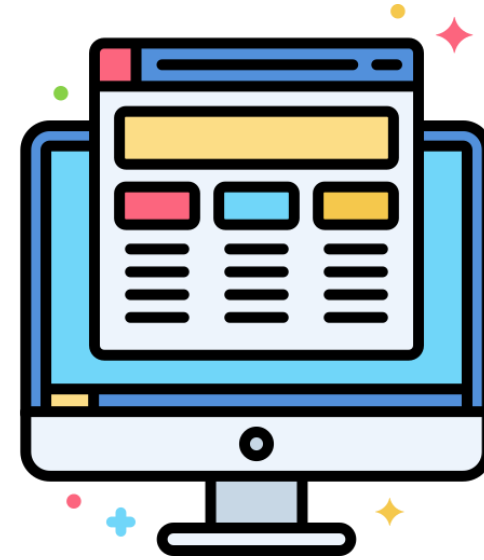


# SCRUM para el reto

Paso 5: se empieza a programar la parte del frontend.



Backend



Frontend

# SCRUM para el reto



# Referencias

- Sommerville, I., Software Engineering, 10th Edition, Pearson, 2016, IN, 1292096144, 9781292096148.
- Connolly Thomas M, Database systems : a practical approach to design, implementation and management, 5thed., London : Addison-Wesley, 2010, 9780321523068.
- Martel, A., Gestión practica de proyectos con SCRUM, , EEUU, : libre, 2016.
- <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

# Gracias!

## Preguntas...

