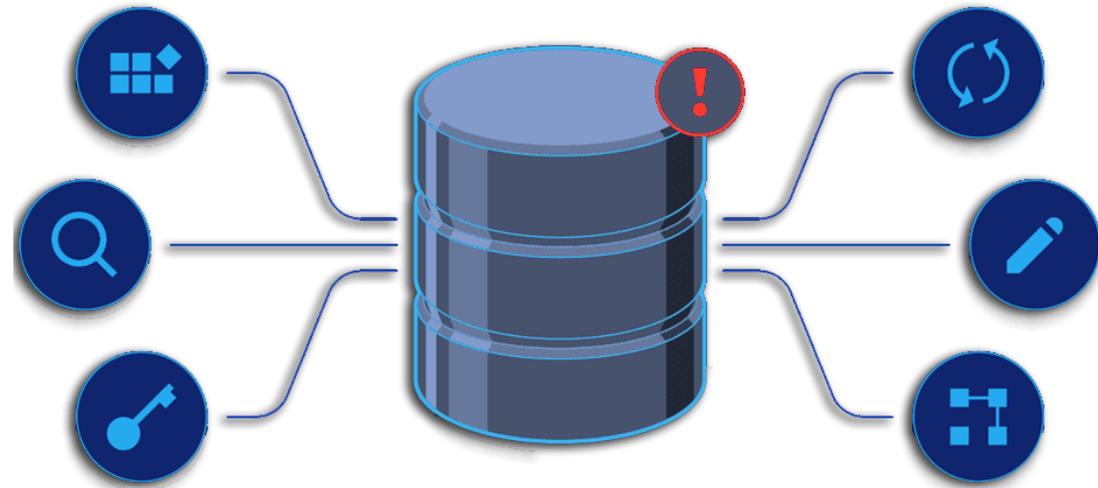




# Buenas Practicas en los Servidores de bases de Datos

Mtro. Armando Vega





## ¿Que son las “Buenas Prácticas”?

Por buenas o mejores prácticas se entiende un conjunto coherente de acciones que han rendido buen o incluso excelente servicio en un determinado contexto y que se espera que, en contextos similares, rindan similares resultados.





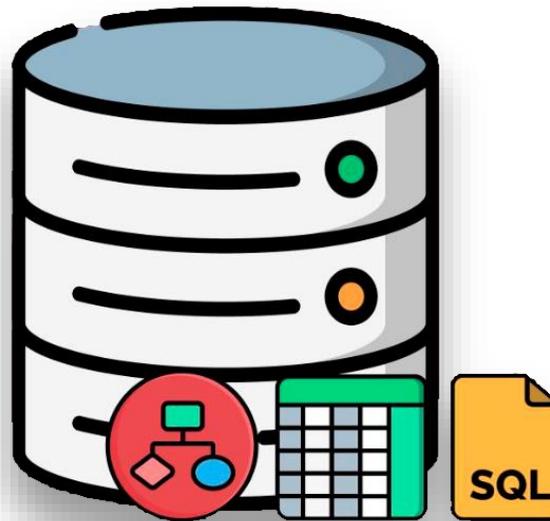
## Al final es:

- Un poco/mucho de sentido común.
- Hacer las cosas bien a la primera o máximo a la segunda.
- Comprometerse con lo que se esta haciendo.





# En el diseño



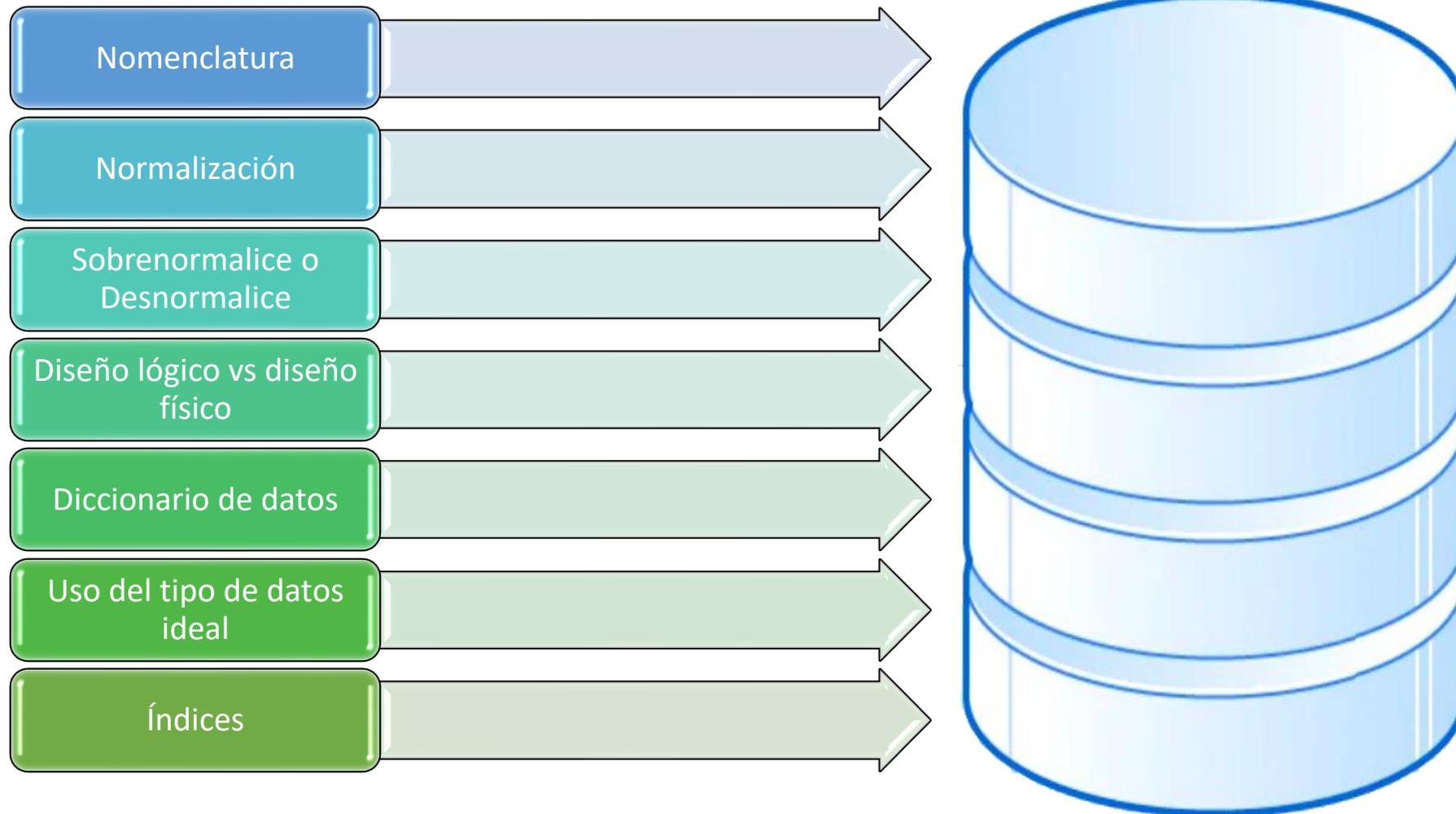


## ¿Qué información tengo que conocer?

- Los datos se almacenan para ser consumidos más tarde y el objetivo siempre es almacenarlos y recuperarlos de la manera más eficiente. Para lograr esto, el diseñador y sobre todo EL ADMINISTRADOR de la base de datos debe saber de antemano qué representarán los datos, cómo se va a adquirir y a qué velocidad, cuál será su volumen operativo (es decir, cuántos datos se esperan) y, finalmente, cómo se usará.



# ¿Qué elementos hay que considerar?





## Nomenclatura

- limite el tamaño del nombre de tabla o columna. Siempre teniendo un nombre descriptivo que entienda el programador.
- Los nombres que son iguales tienen el mismo significado. Evita tener campos que tengan el mismo nombre pero con diferentes tipos o significados; esto será confuso tarde o temprano.
- A menos que sea necesario, no seas redundante. Por ejemplo, en la tabla “Artículo”, no es necesario tener columnas como “Nombre del elemento”, “Precio del artículo” o nombres similares; “Nombre” y “Precio” son suficientes.
- Ten cuidado con las palabras reservadas para el Motor de Base de Datos.



## Nomenclatura

- Si se nombran restricciones, usa un prefijo que describa la restricción (por ejemplo, “PK” o “FK”), seguido del nombre de la tabla o tablas involucradas. Por supuesto, el uso de guiones bajos (“\_”) con moderación ayuda a que las cosas sean más legibles.
- Para nombrar índices, usa el prefijo “IDX” seguido del nombre de la tabla y la columna o columnas del índice o un número secuencial.



Nomenclatura

Ejemplo

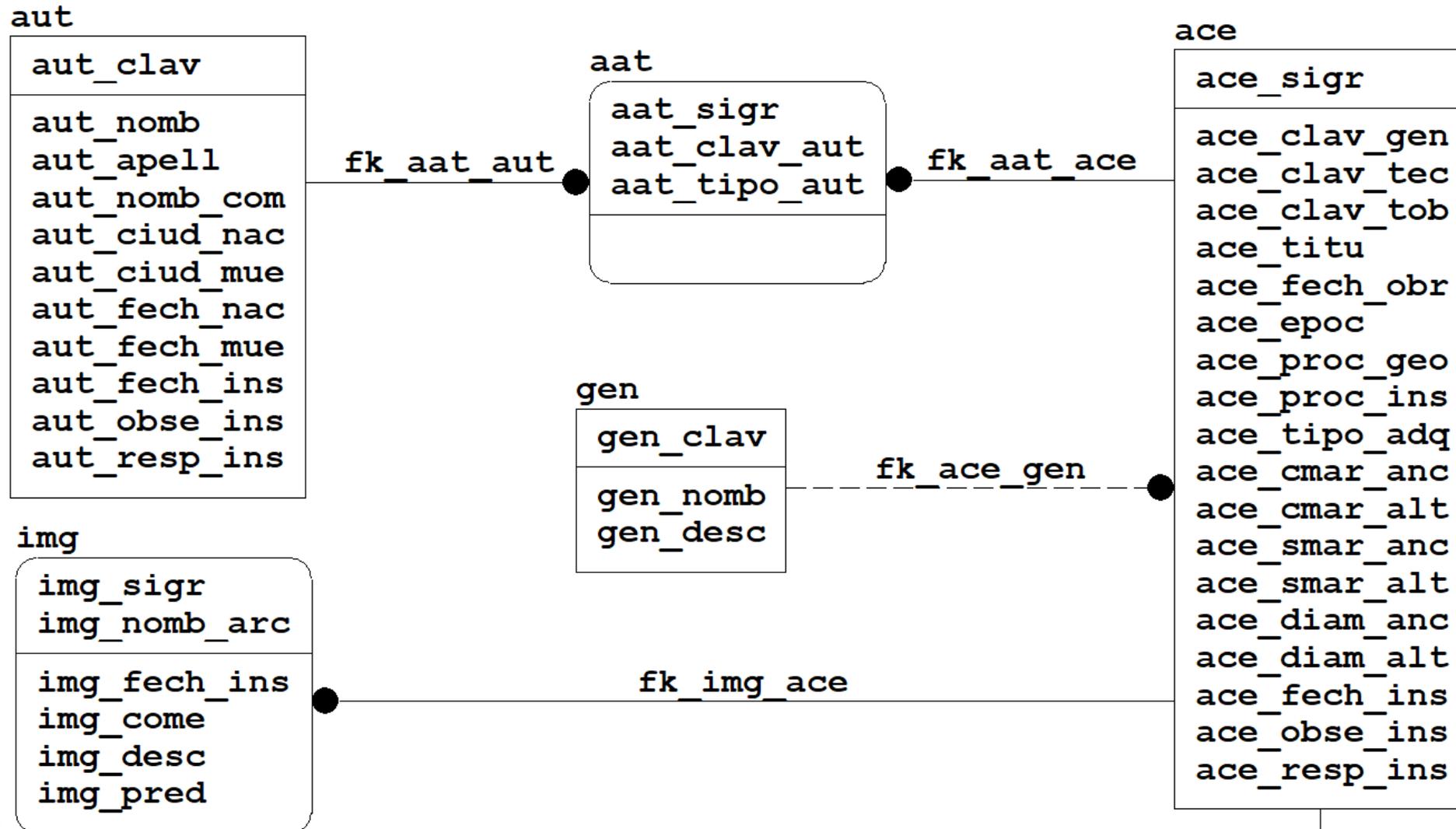
## Diseño para un museo (CONVENCIONES)

- Nombre de tablas a tres caracteres
- Nombre de campos en combinación de tres\_cuatro\_tres caracteres separado por “\_” y todo en minúsculas
- Nombre de llaves primarias **pk**\_nombretabla
- Llaves foráneas **fk**\_tabladependiente\_tabla\_independiente
- Reglas rgl\_nombredcampo
- Default dfl\_nombredelcampo
- Indice idx\_no-consecutivo



Nomenclatura

Ejemplo





# Normalización

## Características indispensables

- Cada atributo debe contener valores atómicos.
- Cada fila de la misma tabla debe ser única.
- Debe prevalecer un crecimiento vertical de los datos y no horizontal.
- No deben existir grupos repetidos de datos.

1FN

## Se basa en:

La característica del modelo relacional, según la cual un atributo debe ser atómico.

- Estar en 1FN.
- La relación debe tener una clave principal, de preferencia simple.
- Cada atributo de la tabla debe depender totalmente del atributo clave.

2FN

El concepto de dependencias funcionales.

- Debe estar en 2FN.
- No deben existir atributos no principales que dependan transitivamente del atributo clave.

3FN

El concepto de dependencias transitivas.



Normalización



# 1FN

Tabla «tutorías»

| Código Estudiante | NombreEstudiante          | Tutoría solicitada                             | Tutor asignado                             |
|-------------------|---------------------------|--|--|
| 45238             | Fernando Rodríguez Solano | Lenguaje Java - Cálculo Diferencial - Física I | Germán Lozano - Pedro Morales - Diego Ríos |
| 23456             | Janeth Castillo           | Física I                                       | Diego Ríos                                 |
| 76822             | Laura Pineda              | Lenguaje Java - Diseño de Algoritmos           | Germán Lozano                              |
| 432526            | Juan Esteban Sarmiento    | Cálculo Diferencial                            | Pedro Morales                              |
| ...               | ...                       | ...  | ...  |

Tabla estudiantes

| Código Estudiante | NombreEstudiante          |
|-------------------|---------------------------|
| 45238             | Fernando Rodríguez Solano |
| 23456             | Janeth Castillo           |
| 76822             | Laura Pineda              |
| 432526            | Juan Esteban Sarmiento    |
| ...               | ...                       |

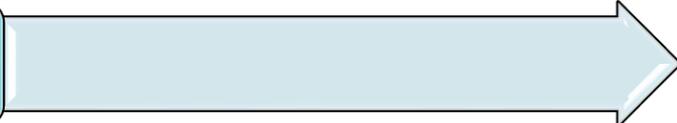
Tabla tutorías

| Código Tutoría | Código Estudiante | Tutoría solicitada  | Tutor asignado |
|----------------|-------------------|---------------------|----------------|
| 1              | 45238             | Lenguaje Java       | Germán Lozano  |
| 2              | 45238             | Cálculo Diferencial | Pedro Morales  |
| 3              | 45238             | Física I            | Diego Ríos     |
| 4              | 23456             | Física I            | Diego Ríos     |
| 5              | 76822             | Lenguaje Java       | Germán Lozano  |





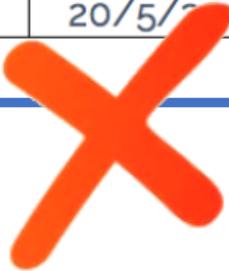
Normalización



2FN

Tabla de proyectos

| Código Proyecto | Nombre Proyecto     | Ciudad        | Fecha Inicio | Jefe Proyecto      | Identificación | Teléfono |
|-----------------|---------------------|---------------|--------------|--------------------|----------------|----------|
| 1               | Mirador del Este    | Villavicencio | 20/10/2017   | Mauricio Serrato   | 82173912       | 37268384 |
| 2               | Panorama III        | Barranquilla  | 1/12/2017    | Víctor Contreras   | 34123793       | 21241445 |
| 3               | Arboleda Campestre  | Ibagué        | 15/5/2018    | Sandra Otálora     | 52098443       | 43720923 |
| 4               | Bosque de la Sabana | Madrid        | 20/5/2018    | Germán Aristizabal | 32479471       | 64520873 |





Normalización



# 3FN



Tabla conductores

| Código Conductor | NombreConductor   | Documento | FechaNacimiento | PlacaVehículo | Marca     | Modelo |
|------------------|-------------------|-----------|-----------------|---------------|-----------|--------|
| 1                | Rigoberto Morales | 82173912  | 21/10/1975      | ASD-321       | Renault   | 2015   |
| 2                | Juan Lozano       | 34123793  | 14/12/1968      | FGH-652       | Chevrolet | 2012   |
| 3                | Luis Sánchez      | 77265136  | 15/5/1980       | DCF-783       | Mazda     | 2016   |
| 4                | Raúl Ortiz        | 32479471  | 12/5/1979       | MHK-207       | Kia       | 2009   |

Dependencia funcional



Dependencia transitiva

Tabla «conductorres»

| Código Conductor | NombreConductor   | Documento | FechaNacimiento |
|------------------|-------------------|-----------|-----------------|
| 1                | Rigoberto Morales | 82173912  | 21/10/1975      |
| 2                | Juan Lozano       | 34123793  | 14/12/1968      |
| 3                | Luis Sánchez      | 77265136  | 15/5/1980       |
| 4                | Raúl Ortiz        | 32479471  | 12/5/1979       |



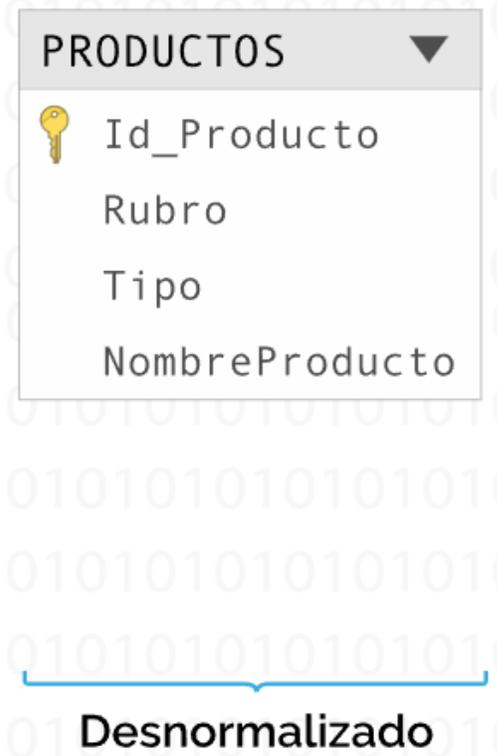
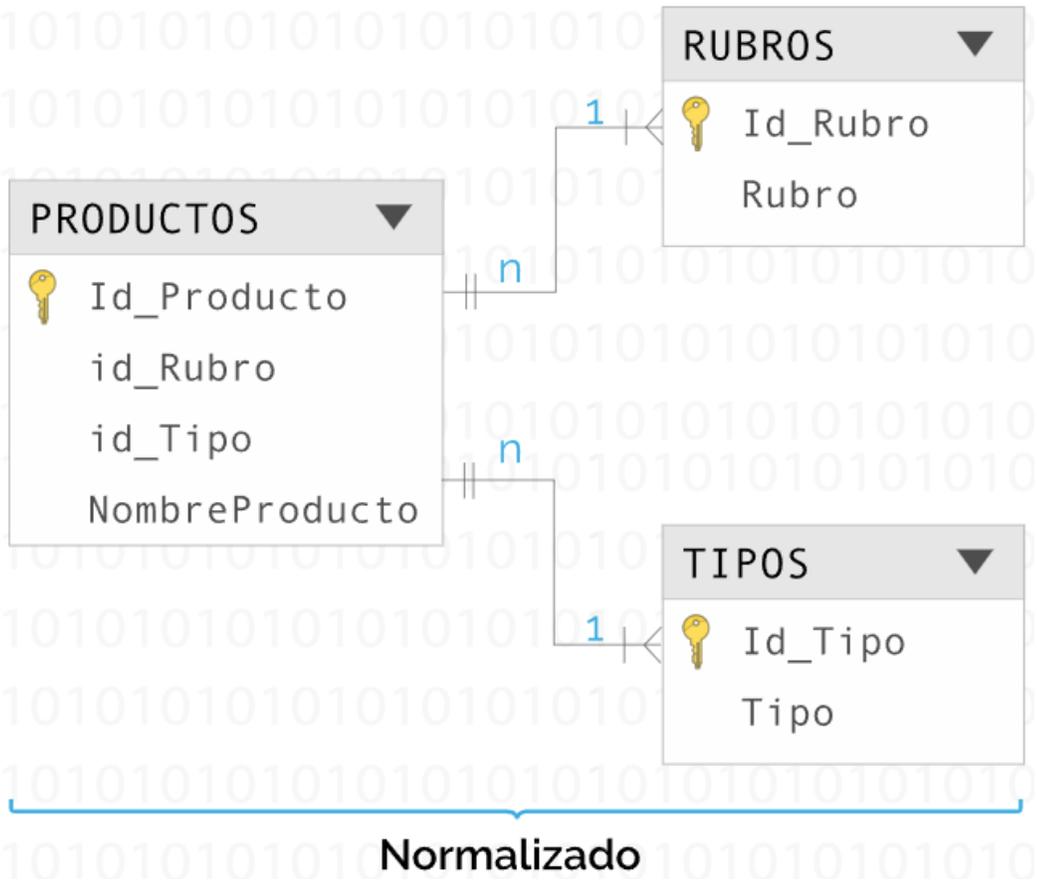
| Código Conductor | PlacaVehículo |
|------------------|---------------|
| 1                | ASD-321       |
| 2                | FGH-652       |
| 3                | DCF-783       |
| 4                | MHK-207       |

Tabla «vehiculos»

| PlacaVehículo | Marca     | Modelo |
|---------------|-----------|--------|
| ASD-321       | Renault   | 2015   |
| FGH-652       | Chevrolet | 2012   |
| DCF-783       | Mazda     | 2016   |
| MHK-207       | Kia       | 2009   |

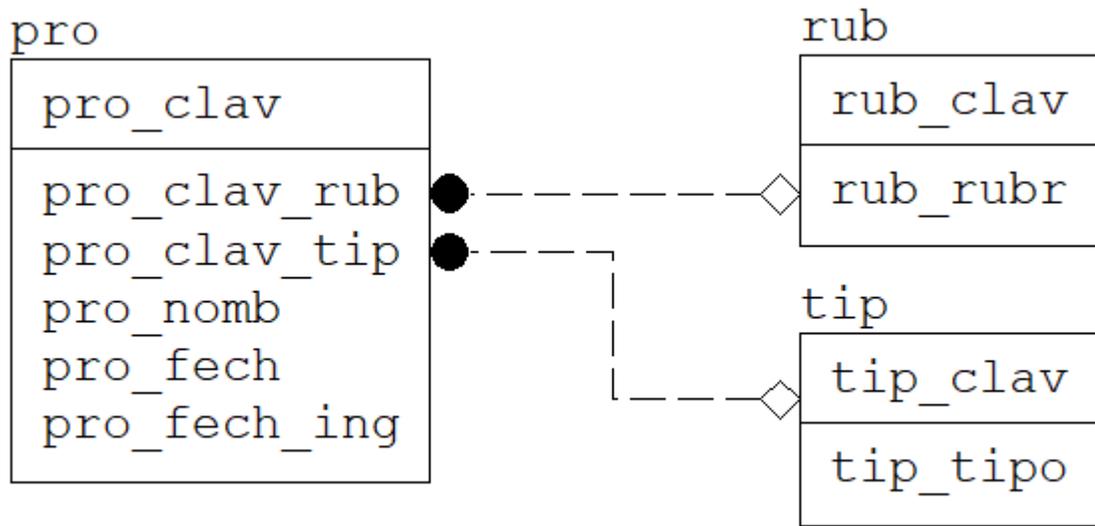


# Desnormalice

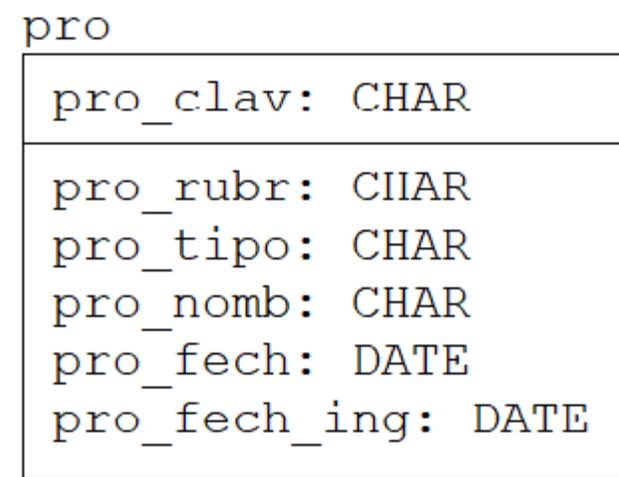




**Desnormalice**



**NORMALIZADO**



**DESNORMALIZADO**





Sobrenormalice

| datos |     |     |     |     |
|-------|-----|-----|-----|-----|
| a     | b   | c   | d   | e   |
| 1     | ... | ... | ... | ... |
| 2     | ... | ... | ... | ... |
| 3     | ... | ... | ... | ... |
| 4     | ... | ... | ... | ... |
| 5     | ... | ... | ... | ... |
| 6     | ... | ... | ... | ... |
| 7     | ... | ... | ... | ... |
| 8     | ... | ... | ... | ... |
| 9     | ... | ... | ... | ... |
| 10    | ... | ... | ... | ... |

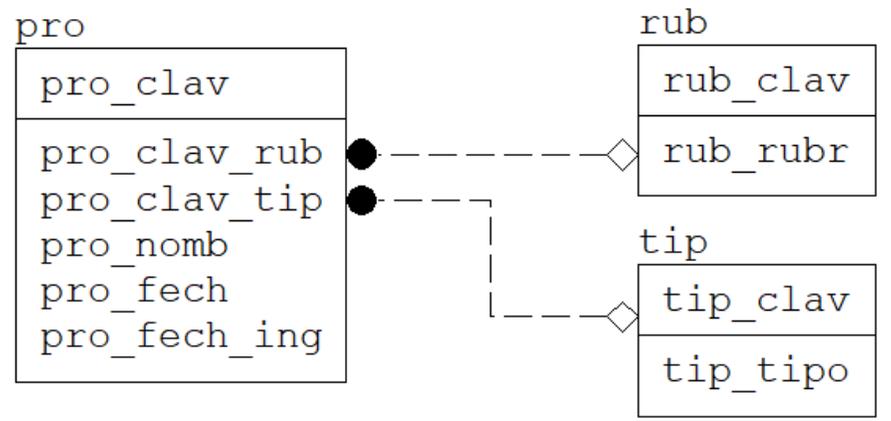


| datos_1 |     |     |     |     |
|---------|-----|-----|-----|-----|
| a       | b   | c   | d   | e   |
| 1       | ... | ... | ... | ... |
| 2       | ... | ... | ... | ... |
| 3       | ... | ... | ... | ... |
| 4       | ... | ... | ... | ... |
| 5       | ... | ... | ... | ... |

| datos_2 |     |     |     |     |
|---------|-----|-----|-----|-----|
| a       | b   | c   | d   | e   |
| 6       | ... | ... | ... | ... |
| 7       | ... | ... | ... | ... |
| 8       | ... | ... | ... | ... |
| 9       | ... | ... | ... | ... |
| 10      | ... | ... | ... | ... |

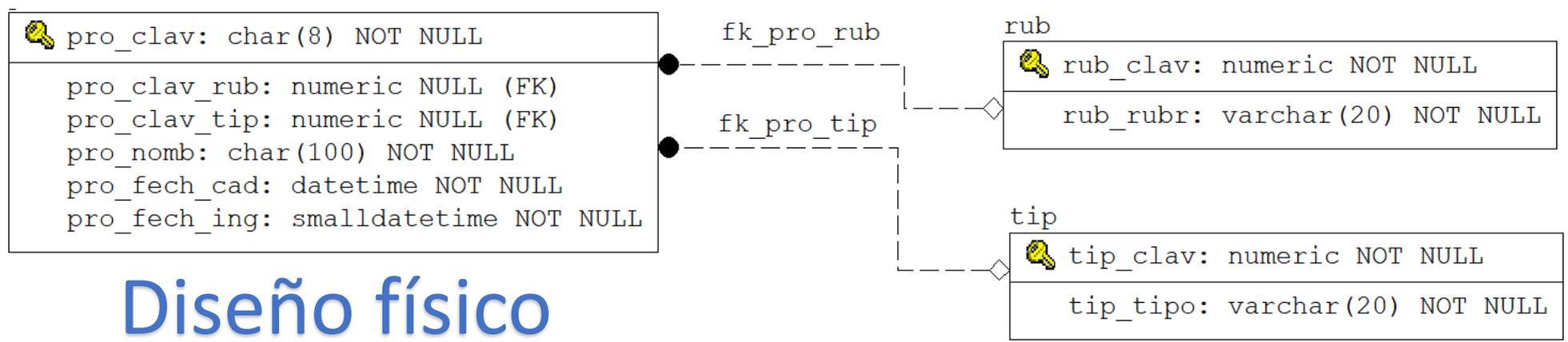


Diseño lógico vs diseño físico



# Diseño lógico

## VS.



# Diseño físico

# Diccionario de datos

## MUSEO DE OBRAS ARTISTICAS SISTEMA INTEGRAL DE EJEMPLO

### DICCIONARIO DE DATOS

| Nombre del campo<br>ajustándose a las reglas definidas de la nomenclatura | Descripción del nombre del campo | Tamaño del campo en bytes  | Tipo de datos dependiendo del servidor | Nombre de la llave foránea indicando que ese campo esta en ella |
|---|----------------------------------|--|--|---|
| aat_sigr  | CLAVE SIGROA DE LA OBRA          | Clave unica e irrepetible  | 1{0-9}7                                | 4 numeric(7) NOT NULL pk_aat fk_aat_sce                         |
| aat_clav_aut  | CLAVE DEL AUTOR DE LA OBRA       | Descripción del contenido del campo utilizando expresiones regulares   | 1{0-9}3                                | NOT NULL pk_aat fk_aat_aut ind1                                 |
| aat_tipo_aut  | TIPO DE AUTOR                    | Tipo de Autor<br>1=autor principal<br>2=coa autor<br>3=editor<br>4=grabador<br>5=inventor<br>6=dibujante<br>7=pintor<br>depende de la participación del tipo de autor. | 1{0-9}2                                | NOT NULL pk_aat ind2  |

Descripción detallada del contenido del campo

Si el campo aceptara o no valores NULL

Campo que esta contenido en la llave primaria

Nombre del índice indicando que ese campo esta contenido en el indice



## Diccionario de datos



```
create table aat (  
aat_sigr          numeric(7) not null,  
aat_clav_aut     numeric(3) not null,  
aat_tipo_aut     numeric(2) not null,  
aat_fech_cap     date       not null,  
aat_prec         money      null)
```

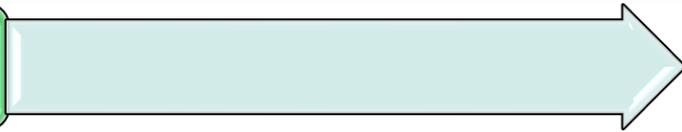
aat

```
aat_sigr: numeric(7)  
aat_clav_aut: numeric(3)  
aat_tipo_aut: numeric(2)
```

```
aat_fech_cap: date  
aat_prec: money
```



## Diccionario de datos



- `create rule rgl_aat_tipo as @gbd_tipo in (1,2,3,4,5,6,7)`
- `bindrule rgl_aat_tipo, "aat.aat_tipo"`
- `alter table aat add constraint pk_aat primary key (aat_sigr, aat_clav_aut, aat_tipo_aut)`
- `primarykey aat, aat_sigr, aat_clav_aut, aat_tipo_aut`
- `alter table bai add constraint fk_aat_aut foreign key (aat_clav_aut) references aut(aut_clav)`
- `foreignkey aat, aut, aat_clav_aut`
- `create index ind01 on aat(aat_clav_aut, aat_tipo_aut)`

aat

```
aat_sigr: numeric(7)
aat_clav_aut: numeric(3)
aat_tipo_aut: numeric(2)
```

```
aat_fech_cap: date
aat_prec: money
```



Uso del tipo de datos ideal



aat  
aat\_sigr  
aat\_clav\_aut  
aat\_tipo\_aut  
aat\_fech\_cap  
aat\_prec

| CAMPO        | CHAR | VARCHAR | INT | DATE | DATETIME | MONEY |
|--------------|------|---------|-----|------|----------|-------|
| aat_sigr     |      |         | X   |      |          |       |
| aat_clav_aut |      |         | X   |      |          |       |
| aat_tipo_aut |      |         | X   |      |          |       |
| aat_fech_cap |      |         |     | X    |          |       |
| aat_prec     |      |         |     |      |          | X     |

Uso del tipo de  
datos ideal

| Tipo de dato  | Rango almacenado   | Bytes |
|---------------|--|-------|
| int[eger]     | $-2^{31}$ and $2^{31} - 1$<br>(-2,147,483,648 - 2,147,483,647) | 4     |
| Smallint      | $-2^{15}$ and $2^{15} - 1$<br>(-32,768 and 32,767)             | 2     |
| Money         | 922,337,203,685,477.58   | 8     |
| date          | Enero 1, 0001 a Diciembre 31, 9999                             | 4     |
| datetime      | Enero 1, 1753 al Diciembre 31, 9999                            | 8     |
| smalldatetime | Enero 1, 1900 al Junio 6, 2079                                 | 4     |
| Bigdatetime   | Enero 1, 0001 a Diciembre 31, 9999                             | 8     |
| Char(n)       | Carácter   | <n>   |
| varchar(n)    | Carácter variable  | NR    |



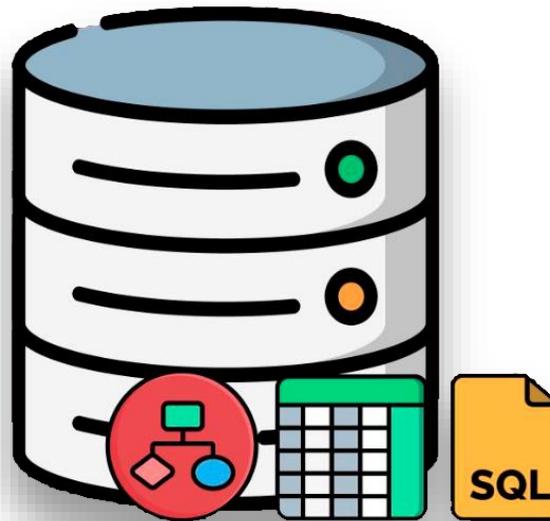
# Índices

| <i>titles</i> |              |
|---------------|--------------|
| title_id      | tid          |
| title         | varchar(80)  |
| type          | char(12)     |
| pub_id        | char(4)      |
| price         | money        |
| advance       | money        |
| total_sales   | int          |
| notes         | varchar(200) |
| pubdate       | datetime     |
| contract      | bit          |

- Create index ind\_01 on titles(title)
- Create index ind\_02 on titles(type)
- Create index ind\_03 on titles(pubdate)
  
- Create unique index ind\_01 on titles(title)
- Create index ind\_02 on titles(title,type)
  
- La indexación es siempre una decisión delicada ya que demasiada indexación puede ser tan mala como muy poca y el tipo de datos de las columnas a indexar tiene una gran influencia en el resultado final.



# En la administración





*La responsabilidad y alcance de las funciones de un administrador de base de datos, es realmente importante y de alta demanda para una organización.*

- **Su misión es garantizar la continuidad del servicio de acceso a los datos.**
  - Aprovechar las características que tiene cada servidor de bases de datos
  - Respaldos y recuperación de desastres.
  - Plan de seguridad de la información
  - Buenas herramientas de alerta

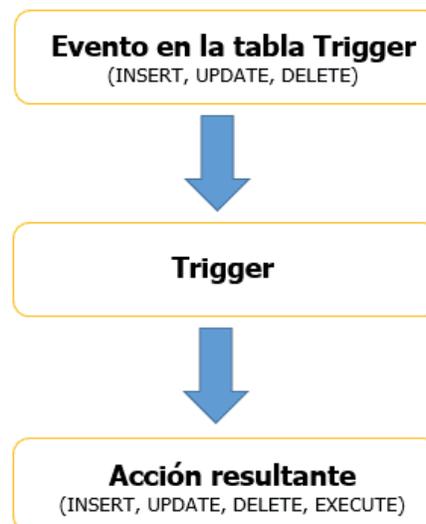




## Aprovechar las características que tiene cada servidor de bases de datos

- constraints
- triggers
- views
- cursores
- Funciones
- Procedimientos almacenados
- Reglas
- defaults
- Índices unique
- grant

```
-- =====  
-- Author:      <Author,,Name>  
-- Create date: <Create Date,,>  
-- Description: <Description,,>  
-- =====  
CREATE PROCEDURE <Procedure_Name, sysname, ProcedureName>  
-- Add the parameters for the stored procedure here  
<@Param1, sysname, @p1> <Datatype_For_Param1, , int> = <Default_Value_For_Param1, , 0>,  
<@Param2, sysname, @p2> <Datatype_For_Param2, , int> = <Default_Value_For_Param2, , 0>  
  
AS  
BEGIN  
-- SET NOCOUNT ON added to prevent extra result sets from  
-- interfering with SELECT statements.  
SET NOCOUNT ON;  
  
-- Insert statements for procedure here  
SELECT <@Param1, sysname, @p1>, <@Param2, sysname, @p2>  
END  
GO
```



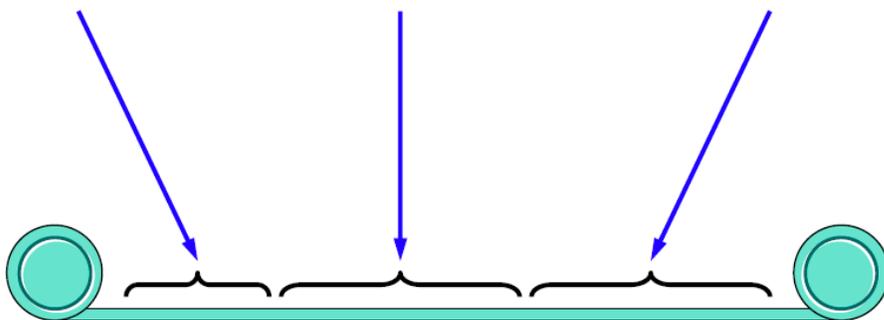


# Respaldos y recuperación de desastres.

dump database db1  
to "/dev/nrmt4"  
with init

dump database db2  
to "/dev/nrmt4"

dump database db3  
to "/dev/nrmt4"  
with unload



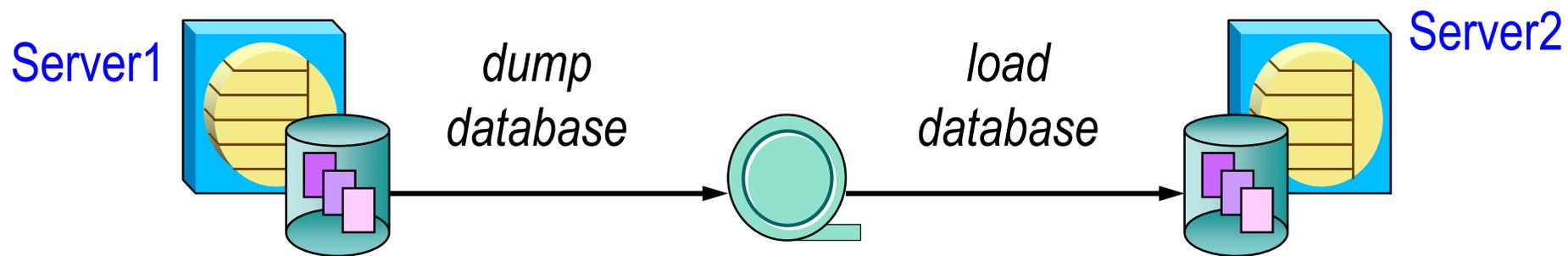
- ¿Quién?
- ¿Cuándo?
- ¿Cómo?
- ¿Dónde?

- **¿Qué tanta información están dispuestos a perder?**





## Respaldos y recuperación de desastres.



- Probar los respaldos
  - Es un esquema de replicación
  - Se requiere tener recursos disponibles
  - **NO ES UN DESPERDICIO DE RECURSOS**
- 
- **¿Qué tanta información están dispuestos a perder?**



# Plan de seguridad de la información

|  |    |
|--|----|
| 1. Introducción .....  | 3  |
| 2. Objetivo .....  | 3  |
| 3. Alcance .....   | 3  |
| 4. Base Legal .....  | 3  |
| 5. Definiciones de Términos .....                              | 3  |
| 6. Entorno de TI .....   | 5  |
| 6.1. Organización de TI .....                                  | 5  |
| 6.2. Centro de Cómputo e Infraestructura Tecnológica .....     | 5  |
| 6.3. Sistemas de Información .....                             | 6  |
| 7. Evaluación de Riesgos .....                                 | 6  |
| 8. Procedimientos de seguridad de información .....            | 6  |
| 8.1. Organización de la seguridad .....                        | 7  |
| 8.2. Clasificación y control de activos de información .....   | 10 |
| 8.3. Seguridad de personal .....                               | 12 |
| 8.4. Seguridad física y ambiental .....                        | 15 |
| 8.5. Seguridad de comunicaciones y operaciones .....           | 18 |
| 8.6. Control de accesos .....                                  | 24 |
| 8.7. Adquisición, desarrollo y mantenimiento de sistemas ..... | 29 |
| 8.8. Gestión de incidentes .....                               | 32 |
| 8.9. Gestión de continuidad del negocio .....                  | 34 |
| 8.10. Cumplimiento .....                                       | 36 |
| 9. Elaboración, validación, aprobación y vigencia .....        | 37 |
| 10. Anexos .....   | 38 |

- ¿Para que?
- ¿Quién?
- ¿Cuándo?
- ¿Cómo?
- ¿Dónde?
- Protección de datos personales.





## Buenas herramientas de alerta (ejemplo)

- curl en cualquier maquina unix ó Linux una herramienta que sabiéndola usar nos permite mandar mensajes a telegram

```
#!/bin/bash
VAR1=$(/usr/bin/curl -Is https://www.unam.mx > /dev/null 2>&1; echo $?)
if [ $VAR1 -eq 0 ]; then
    curl -s -X POST https://api.telegram.org/bot13:1Hg/sendMessage -d
chat_id=-399 -d text="*UNAM ARRIBA" > /dev/null 2>&1
else
    curl -s -X POST https://api.telegram.org/bot13:1Hg/sendMessage -d
chat_id=-399 -d text="UNAM ABAJO*" > /dev/null 2>&1
fi
```





# MUCHAS GRACIAS

## ¿Preguntas?

Mtro. Armando Vega

[avega@unam.mx](mailto:avega@unam.mx)