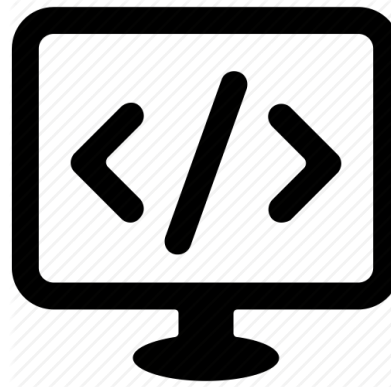


# GESTIÓN DE BASES DE DATOS

## BNF

NOTACIÓN BACKUS-NAUR-FORM  
PARA LA ESPECIFICACIÓN DE SINTAXIS DE  
LENGUAJES FORMALES



Francisco Pérez  
franciscoprdv@gmail.com

[Esta obra está bajo una licencia de Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

- Es un metalenguaje que permite expresar las construcciones admitidas por un lenguaje formal, es decir, permite especificar la *sintaxis* de un lenguaje formal.

¿QUÉ ES?

¿DÓNDE SE USA?

- Generalmente, aparece en la documentación de cualquier lenguaje formal (shell script, Java, C#, SQL, etc.), para especificar su sintaxis.

- Al ser un estándar de facto, no existe una versión normalizada. Por esa razón, en la documentación de distintos lenguajes pueden encontrarse distintas adaptaciones de BNF, aunque suelen ser bastante similares.

¿ESTÁNDAR?

BNF

¿PARA QUÉ SE USA?

- Para expresar cómo debe escribirse una determinada proposición o sentencia en lenguaje determinado.
- Mediante BNF se pueden expresar aspectos como qué palabras clave forman la sentencia, qué parámetros son opcionales y cuáles obligatorios, etc.

- BNF puede proporcionar información ambigua, incompleta o de difícil comprensión en sentencias complejas muy estructuradas.

¿DESVENTAJAS?

¿CÓMO SE USA?

- La notación BNF recoge cinco tipos de cláusulas básicas, que pueden combinarse para especificar las construcciones válidas de una proposición o sentencia del lenguaje

**literal**

## LITERALES Y PALABRAS CLAVE QUE FORMAN EL CUERPO DE LA SENTENCIA

- Se especifican directamente, sin etiquetarse.
- Su inclusión en la sentencia es necesaria para que ésta sea correcta.

**<obligatorio>**

## CLÁUSULA OBLIGATORIA

- Representa una cláusula que debe especificarse necesariamente para que la sentencia sea correcta.

**[opcional]**

## CLÁUSULA OPCIONAL

- Representa una cláusula que puede incluirse u omitirse en la sentencia.

**{iterable}**

## CLÁUSULA ITERABLE

- Representa una cláusula que puede aparecer más de una vez en la sentencia.

**opción1 | opción2**

## CLÁUSULA DE EXCLUSIÓN

- Representa un conjunto de valores mutuamente excluyentes.
- Solamente puede seleccionarse uno de los valores.

**literal****LITERALES Y PALABRAS CLAVE QUE FORMAN EL CUERPO DE LA SENTENCIA**

- Se especifican directamente, sin etiquetarse.
- Su inclusión en la sentencia es necesaria para que ésta sea correcta.

**EJEMPLO:****CREATE DATABASE**

- La sintaxis especifica las palabras clave 'CREATE' y 'DATABASE'.
- Ambas son obligatorias y deben especificarse en el mismo orden en que aparecen.

## CLÁUSULA OBLIGATORIA

*<obligatorio>*

- Representa una cláusula que debe especificarse necesariamente para que la sentencia sea correcta.

### EJEMPLO:

**CREATE DATABASE** *<nombre\_base\_de\_datos>*

- La sintaxis especifica las palabras clave '**CREATE**' y '**DATABASE**'.
- Ambas son obligatorias y deben especificarse en el mismo orden en que aparecen.
- Además, la sintaxis introduce la cláusula obligatoria *<nombre\_base\_de\_datos>*

- Ejemplo de sentencia válida

- **CREATE DATABASE** Test

- Ejemplos de sentencias no válidas

- **CREATE DATABASE** → Falta la cláusula obligatoria *<nombre\_base\_de\_datos>*

- **CREATE** Test → Falta la palabra clave '**DATABASE**'

[*opcional*]

## CLÁUSULA OPCIONAL

- Representa una cláusula que puede incluirse u omitirse en la sentencia.

### EJEMPLO:

DECLARE @<nombre\_de\_variable> [AS] <tipo\_de\_dato>

- La sintaxis especifica la palabra clave '**DECLARE**'.
- A continuación, debe incluirse la cláusula obligatoria <nombre\_de\_variable>, **que debe comenzar con el literal '@'**.
- La palabra clave '**AS**' está especificada como una cláusula opcional, por lo que puede incluirse u omitirse en la sentencia.
- Finalmente, debe incluirse la cláusula obligatoria <tipo\_de\_dato>

- Ejemplos de sentencias válidas:

- DECLARE @i AS INT
- DECLARE @j INT

## CLÁUSULA ITERABLE

**{*iterable*}**

- Representa una cláusula que puede aparecer más de una vez en la sentencia.

### EJEMPLO:

**DECLARE { @<nombre\_de\_variable> [AS] <tipo\_de\_dato> [ , ... ] }**

- La sintaxis especifica la palabra clave '**DECLARE**'.
- A continuación, debe incluirse la cláusula obligatoria <nombre\_de\_variable>, **que debe comenzar con el literal '@'**.
- La palabra clave '**AS**' está especificada como una cláusula opcional, por lo que puede incluirse u omitirse en la sentencia.
- Finalmente, debe incluirse la cláusula obligatoria <tipo\_de\_dato>
- La secuencia { @<nombre\_de\_variable> [**AS**] <tipo\_de\_dato> } puede repetirse, separando cada proposición con un carácter ','.

- Ejemplos de sentencias válidas:
  - **DECLARE @i AS INT**
  - **DECLARE @j INT, @k AS INT**

## CLÁUSULA DE EXCLUSIÓN

*opción1* | *opción2*

- Representa un conjunto de valores mutuamente excluyentes.
- Solamente puede seleccionarse uno de los valores.

### EJEMPLO:

**DROP DATABASE** <nombre\_base\_de\_datos | **CURRENT**>

- La sintaxis especifica las palabras clave '**DROP**' y '**DATABASE**'.
- Ambas son obligatorias y deben especificarse en el mismo orden en que aparecen.
- Además, la sintaxis introduce la cláusula <nombre\_base\_de\_datos | **CURRENT**>, que indica que, obligatoriamente, debe especificarse o bien <nombre\_base\_de\_datos>, o bien la palabra clave **CURRENT**.

- Ejemplos de sentencias válidas:
  - **DROP DATABASE** Test
  - **DROP DATABASE CURRENT**
- Ejemplo de sentencia no válida:
  - **DROP DATABASE**