


Open   Opened 3 days ago by  **Rubén Montero**

# NoSQL

## Resumen

- Hablaremos de bases de datos NoSQL
- Instalaremos eXistDB y visitaremos el *dashboard* de administración
- Introduciremos XQuery
- Crearemos un sencillo programa en XQuery que genera un XML de *Hola mundo*
- Lo guardaremos en nuestro repositorio

## Descripción

Hemos trabajado detenidamente [bases de datos relacionales](#) usando el lenguaje de consultas SQL y el *mapeo objeto-relacional*. Exponemos los datos mediante una [fachada REST](#) y, en resumen, nos hemos ganado una buena porción del pastel del desarrollo *full-stack*.

Pero, ¿eso es todo?

### ¡Nunca es todo!

Así es.

Dediquemos un rato a otro tipo de tecnología que también pertenece al mundo del acceso a datos, pero no guarda relación con las bases de datos SQL. De hecho, se llama...

## NoSQL

Las [bases de datos NoSQL](#) son sistemas de bases de datos **no** estructurados.

- Son aptas para manejar tipos de datos complejos y que cambian constantemente
- Se pueden agregar datos sin definir previamente su estructura
- Posibilitan procesar rápidamente grandes volúmenes de datos y realizar desarrollos ágiles
- Principalmente, los datos son almacenados en forma de *documentos* XML ó JSON

[MongoDB](#) y [ElasticSearch](#) son dos de las bases de datos NoSQL más populares.

Nosotros trabajaremos con...

### eXist DB

También es una de las plataformas NoSQL más conocida:

- Se centra en el manejo de **documentos XML**
- Ofrece mucho más potencial que aquello a lo que estamos acostumbrados en SQL. Veremos cómo mediante unos *scripts* [XQuery](#) se puede, por ejemplo, crear páginas web generadas dinámicamente
- Es de licencia abierta

## ¡Instalemos [eXist DB](#)!

### 1. Asegurémonos de tener Java disponible

Abre un terminal (cmd.exe) de Windows. Teclea:

```
java -version
```

¿Ves la salida correcta del comando?

```
openjdk version "18.0.2.1" 2022-08-18
OpenJDK Runtime Environment (build 18.0.2.1+1-1)
```

¡Sí!

Excelente. Puedes avanzar hasta el apartado de **Descargar e instalar eXist**

## Un momento, ¡no lo veo!

Si no puedes usar Java desde la línea de comandos, seguramente instalaste el [JDK](#) cuando descargaste [IntelliJ IDEA](#) pero no está disponible en las variables de entorno.

## Un segundo... ¿Qué?

Abre el explorador de archivos de Windows.

Ve a *Este equipo* > *C:* > *Usuarios* > *Developer*. Si ves una carpeta llamada `.jdk\`, ¡bien! Ábrela. Dentro habrá una carpeta `openjdk-x.y.z\`, siendo `x.y.z` la versión de Java que instalaste. Dentro, otra carpeta llamada `bin\`.

En la carpeta `bin\` se encuentra Java.

## Entonces, ¿por qué no funciona `java -version`?

Porque el directorio `C:/Users/Developer/.jdk/openjdk-x.y.z/bin` no está en el PATH.

## ¿Qué es el PATH?

Una *variable de entorno* que está en todos los sistemas operativos.

Cuando escribes un *comando* (e.g.: `java`) en un terminal (cmd.exe), Windows *busca* ese archivo ejecutable en los directorios que pertenecen al PATH. Si **no** lo encuentra, salta el error que estarás viendo:

```
"java" no se reconoce como un comando interno o externo,  
programa o archivo por lotes ejecutable.
```

## ¿Y qué hacemos?

1. Pulsa la tecla de Windows y teclea *variables de entorno*. Deberías ver una opción *Panel de Control* > **Editar variables de entorno del sistema**. Haz *click*
2. En la ventana que sale, haz *click* en el botón *Variables de entorno...* de la parte inferior.
3. Verás un diálogo con dos recuadros blancos. En el inferior (*Variables del sistema*), busca "Path".
4. Dale a *Editar...*
5. Vamos a pulsar *Nuevo* y a escribir la ruta que descubrimos anteriormente: `C:/Users/Developer/.jdk/openjdk-x.y.z/bin`. Asegúrate que `x.y.z` se corresponde con tu versión de Java y que la ruta es correcta y existe en tu sistema
6. Aceptamos, aceptamos y aceptamos

Cuando se cierre el último diálogo, abre un **nuevo** terminal (cmd.exe). Ahora, `java -version` debería generar la salida esperada.

## Descargar e instalar eXist

Desde la [página de eXist](#) habrá un enlace de Descarga que lleva a GitHub. Allí, veremos las [página de descarga de las distintas releases](#).

Descargamos el `.jar` de la más reciente. Será `exist-installer-x.y.z.jar`.

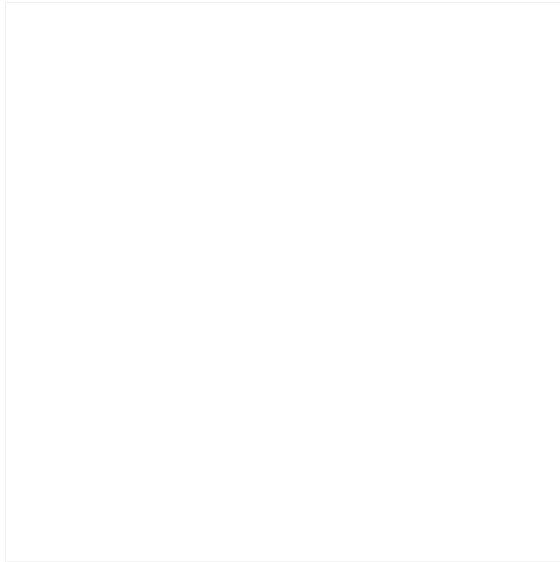
## Instalación

Seguiremos los [pasos de la guía oficial de instalación](#).

En primer lugar, abrimos un terminal Windows (cmd.exe) y usando `cd` cambiamos hasta el directorio de Descargas.

**Lanzamos** el instalador con:

```
java -jar exist-installer-x.y.z.jar
```



Cuando lleguemos al paso que solicita **nombre de usuario** y **contraseña** de administrador, introducimos unas credenciales cualesquiera de las que nos acordemos después. No haremos nada serio, así que pueden ser `admin` , `admin` .

El resto de la instalación la completamos manteniendo todas las opciones por defecto.

## Tras la instalación

Ejecutamos **eXist DB** pulsando en la tecla de windows y buscando el programa:



Deberemos **confirmar** la configuración.

Luego, el programa procederá a **instalar eXist como servicio Windows**. Aceptamos todo y esperamos a que el proceso termine.

## Cuando termine...

Verás abajo a la derecha el *popup* de Windows indicando que el servicio está corriendo.

¡Enhorabuena! Has instalado tu base de datos NoSQL.

Puedes visitar el *dashboard* de administración desde tu navegador favorito en:

- <http://localhost:8080/exist>

## ¿Y ahora?

Probaremos **eXide**, la herramienta que tienes a disposición para escribir programas en [XQuery](#) y administrar los datos de tu base de datos.

## ¿Qué es [XQuery](#)?

Un **lenguaje de consulta** que permite trabajar con bases de datos o documentos XML.

Es como SQL, pero para bases de datos XML. Se trata de un lenguaje amplio y complejo, recomendado por el [W3C](#).

Abre **eXide** para comprobar su funcionamiento.

## Un primer programa sencillo

Hay una [guía con muy buenos ejemplos](#) para comenzar con XQuery.

Nosotros vamos a escribir un sencillo programa que *genera un XML de "Hola mundo"*:

```
xquery version "3.1";

let $message := 'Hello there'
return
<results>
  <message>{$message}</message>
</results>
```

## Guardarlo

### Nueva colección

Haz click en *File > Manage*. Verás un diálogo.

Entra en **apps/**, y, desde allí, usa el tercer botón de la barra superior dentro del diálogo. Tiene un icono de una pequeña carpeta. Indica *Create collection*.

Dale el nombre **hello** a tu nueva colección. Luego, *Close*.

## Guardando

Ya estamos listos para darle a *Save* y entrar en **apps/hello/**. Introduce tus credenciales si aún no estás *logueado*.

Guarda tu *script* como **hello.xql**. Ya podremos darle a **Run**.

## Salida de ejecución

Nuestro *Hola mundo* **generará un XML** que se abrirá en una pestaña nueva.

Vamos a **guardar** el archivo generado (*click* derecho > Guardar como) y salvar **hello.xql.xml** en nuestro repositorio, **dentro** de la carpeta **python-sessions/exist/results/**.

Hemos comprobado que funciona el entorno y hecho una breve introducción.

¡Tarea terminada!

## Por último

Verifica que tu código pasa el *test* asociado a la tarea. Tendrás que ejecutar **test\_016exist\_hello\_world.py** usando Python.

Haz **commit** y **push** para subir los cambios al repositorio.



Rubén Montero @ruben.montero changed milestone to [%Sprint 5](#) 3 days ago