

Contenidos a Trabajar

1. Base de Datos

• Objetivos de las Bases de datos

- Áreas de Aplicación de los Sistemas de Bases de datos

• Arquitectura del Sistema Gestor de Bases de datos

2. SQL vs NoSQL

- SQL

- ¿Usar SQL o NoSQL?

3. ¿Cómo hago las transformaciones?

4. Instalar PostgreSQL

5. Repaso SQL

Base de Datos

Objetivo de las bases de datos

Un **sistema gestor de bases de datos (SGBD)** consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además los sistemas de base de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización.

Una base de datos se puede definir como:

- Una colección interrelacionada de datos, almacenados en un conjunto sin redundancias innecesarias cuya finalidad es la de servir a una o más aplicaciones de la manera más eficiente.
- Conjunto de datos organizados con características afines entre sí, que identifican a un ente en especial, cualquier colección de información interrelacionada, es una base de datos.
- Conjunto autodescriptivo de registros integrados; autodescriptivo ya que puede contener información del usuario, descripción de la Base de Datos, que nos permiten representar las relaciones entre los datos. En donde la información que se tiene almacenada esta disponible para todos los usuarios del sistema, en los que los datos redundantes pueden eliminarse o al menos minimizarse.

Los componentes de una base de datos son:

Los datos: El componente fundamental de una base de datos son los datos que están interrelacionados entre si, formando un conjunto con un mínimo de redundancias.



El software: los datos, para que puedan ser utilizados por diferentes usuarios y diferentes aplicaciones, deben estar estructurados y almacenados de forma independiente de las aplicaciones. Para ello se utiliza un software o conjunto de programas que actúa de interfaz entre los datos y las aplicaciones. A este software se le denomina Sistema de Gestión de Base de Datos (SGBD). El SGBD crea y organiza la base de datos, y además atiende todas las solicitudes de acceso hechas a la base de datos tanto por los usuarios como por las aplicaciones.

Recurso Humano:

- **Informático:** son los profesionales que definen y preparan la base de datos.
- **Usuarios:** Son los que tienen acceso a la base de datos para ingresar o manipular la información contenida en esta, dependiendo del nivel de usuario o control de seguridad establecido.

Áreas de Aplicación de los Sistemas de Bases de datos

Las bases de datos son ampliamente usadas. Las siguientes son algunas de sus aplicaciones más representativas:

Banca: Para llevar el control de la información de los clientes, cuentas, préstamos y todas las transacciones bancarias.

Líneas aéreas: Para llevar el control de todas las planificaciones de vuelos de una aerolínea y las reservaciones hechas por los clientes. Las líneas aéreas fueron de las primeras en usar las bases de datos de forma distribuida geográficamente (las terminales situadas en todo el mundo accedían al sistema de base de datos centralizado a través de las líneas telefónicas y otras redes de datos).

Escuelas: Para llevar el control de los estudiantes, horarios, cursos, inscripciones y calificaciones.

Transacciones de tarjetas de crédito: Para llevar el control de compras con tarjeta de crédito y generación mensual de estados de cuenta de los usuarios.

Telecomunicaciones: Para guardar un registro de las llamadas realizadas, generación mensual de facturas, control del saldo de las tarjetas de prepago y almacenar información sobre las redes de comunicaciones.

Finanzas: Para almacenar información sobre grandes empresas, ventas y compras de documentos formales financieros, como bolsa y bonos.



Ventas: Para tener el control de la información de los clientes, productos y la información de las ventas generadas de forma diaria o mensual según el control que se desee tener.

Producción: Para la gestión de la cadena de producción, seguimiento de la producción y control de inventarios en almacén.

Recursos humanos: Para llevar el control de información sobre los empleados, salarios, impuestos y con estopoder generar la nómina.

Arquitectura del Sistema Gestor de Bases de datos

La mayoría de usuarios de un sistema de bases de datos no están situados actualmente junto al sistema de bases de datos, sino que se conectan a él a través de una red.

Se puede diferenciar entonces entre las máquinas cliente, en donde trabajan los usuarios remotos de la base de datos, y las máquinas servidor, en las que se ejecuta el sistema de bases de datos.

Las aplicaciones de bases de datos se dividen usualmente en dos o tres partes, como se ilustra en la Figura 1.2. En una arquitectura de dos capas, la aplicación se divide en un componente que reside en la máquina cliente, que llama a la funcionalidad del sistema de bases de datos en la máquina servidor mediante instrucciones del lenguaje de consultas. Los estándares de interfaces de programas de aplicación como ODBC y JDBC se usan para la interacción entre el cliente y el servidor.

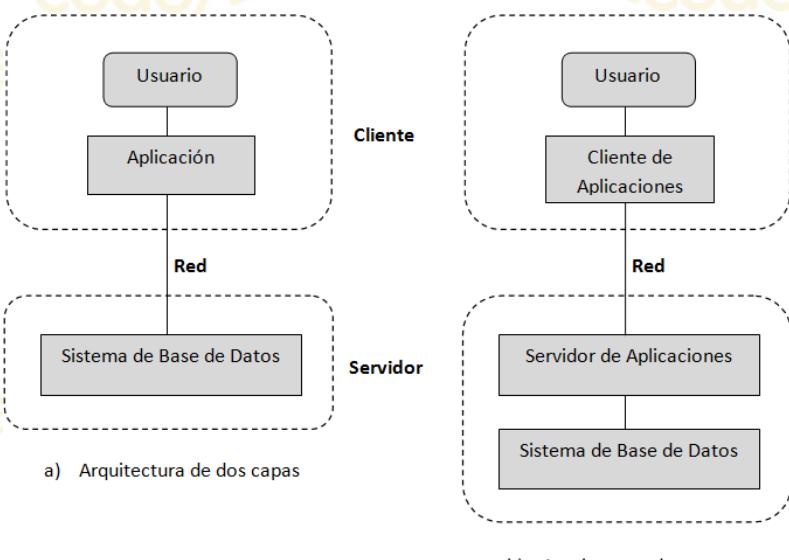
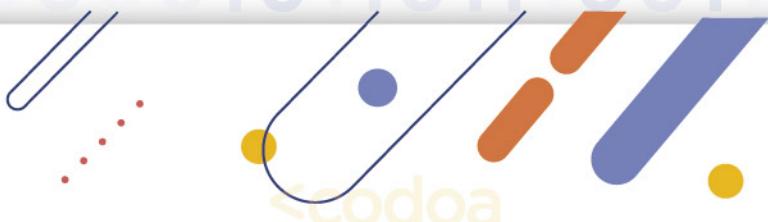


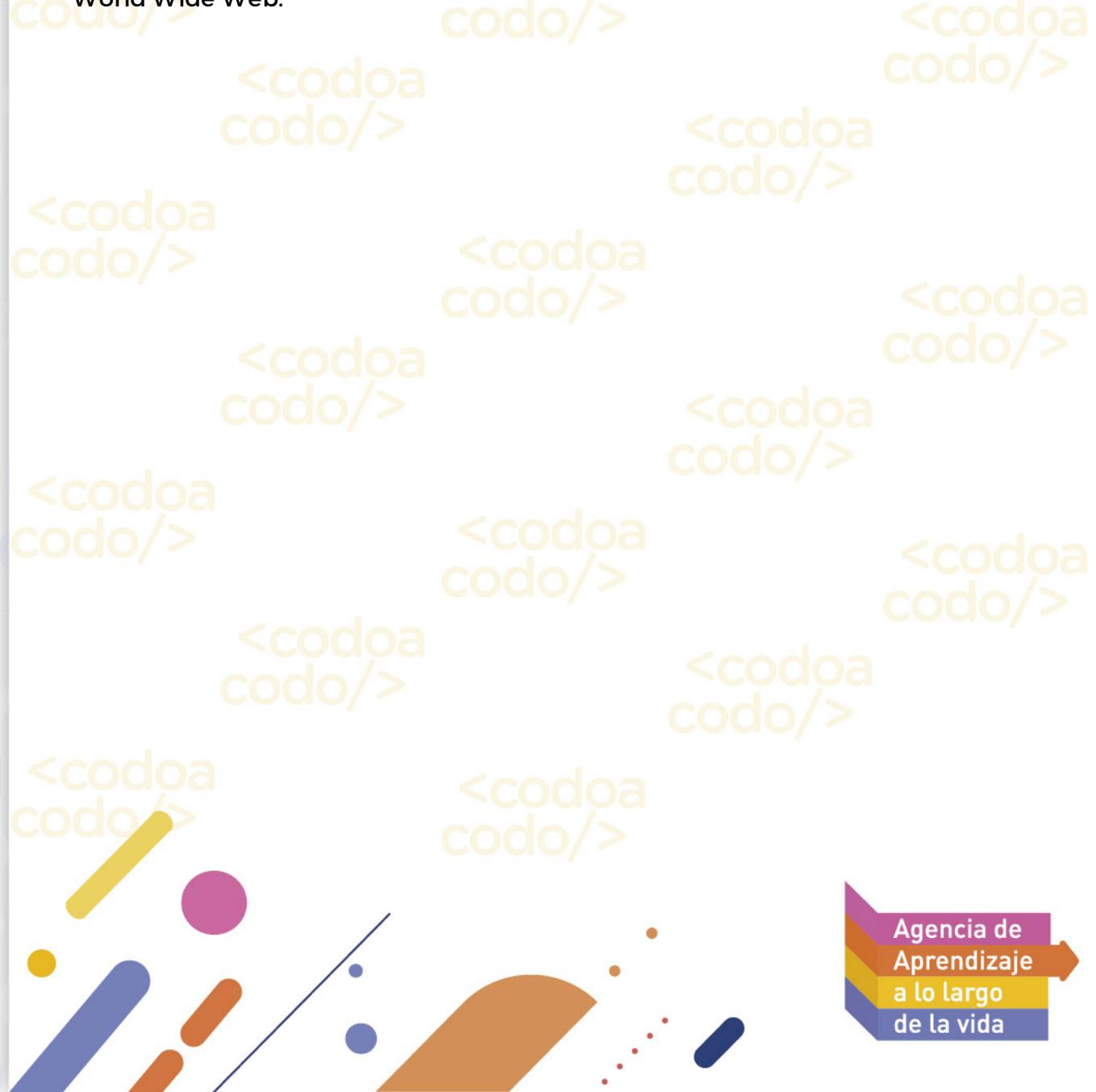
Figura 1.2. Arquitectura de dos y tres capas.



En cambio, en una arquitectura de tres capas, la máquina cliente actúa simplemente como frontal y no contiene ninguna llamada directa a la base de datos. En su lugar, el cliente se comunica con un servidor de aplicaciones, usualmente mediante una interfaz de formularios.

El servidor de aplicaciones, a su vez, se comunica con el sistema de bases de datos para acceder a los datos.

La lógica de negocio de la aplicación, que establece las acciones a realizar bajo determinadas condiciones, se incorpora en el servidor de aplicaciones, en lugar de ser distribuida a múltiples clientes. Las aplicaciones de tres capas son más apropiadas para grandes aplicaciones, y para las aplicaciones que se ejecutan en World Wide Web.





SQL vs NoSQL

SQL

SQL es el lenguaje del sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones; usado por muchos sitios web grandes y populares, como Wikipedia, Google (no para búsquedas), Facebook, Twitter, Flickr, y YouTube.

La diferencia conceptual entre SQL y NoSQL, es que resuelven escenarios completamente diferentes y excluyentes el uno del otro; ya que para lo que resulta ideal SQL, no lo es NoSQL y al revés.

Diferencias:

- SQL permite combinar de forma eficiente diferentes tablas para extraer información relacionada, mientras que NoSQL no lo permite o muy limitadamente.
- NoSQL permite distribuir grandes cantidades de información; mientras que SQL facilita distribuir bases de datos relacionales.
- SQL permite gestionar los datos junto con las relaciones existentes entre ellos; en NoSQL no existe este tipo de utilidades.
- NoSQL permite un escalado horizontal sin problemas – por su capacidad de distribución; mientras que escalar SQL resulta más complicado.

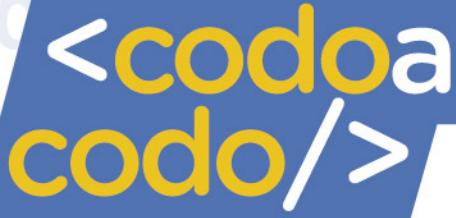
¿Usar SQL o NoSQL?

¿CUÁNDΟ USAR SQL?

No existe una respuesta concreta, porque se presume que cualquier cosa que se deba guardar, debe ser en una base de datos relacional. Sin embargo, los programadores podrían sugerir que en gaming o desarrollos de animación, no sería necesario.

En la mayoría de las opiniones, una base de datos relacional puede ser usada los siguientes ámbitos:

- Educación: para estructurar información, y aportar conocimiento lógico al estudiante.



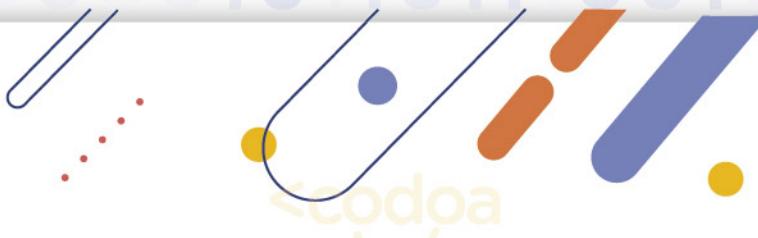
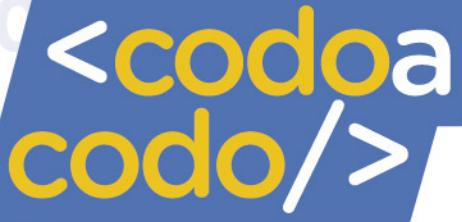
- Desarrollos web: para mantener jerarquía de datos, siempre y cuando la capacidad de concurrencia, almacenamiento y mantenimiento no sean de considerable dificultad y la información sea consistente.
- Negocios: inteligencia y análisis de negocios, son temas que requieren el uso de SQL para facilitar el consumo de la información y la identificación de patrones en los datos.
- Empresarial: porque tanto el software a la medida y el software empresarial, poseen la característica de mantener información con estructura consistente.

¿CUÁNDO USAR NOSQL?

Básicamente se utilizan en:

- Redes sociales: casi obligatorio.
- Desarrollo Web: debido a la poca uniformidad de la información que se encuentra en Internet; aunque también puede emplearse SQL.
- Desarrollo Móvil: debido a la tendencia - en crecimiento- de Bring Your Own Device.
- BigData: debido a la administración de grandísimas cantidades de información y su evidente heterogeneidad.

Lo más importante que debes saber es que las NoSQL no son un sustituto de las SQL, sino que son una alternativa que ofrece otras posibilidades, lo que las convierte en más interesantes para determinados casos como por ejemplo proyectos que requieren una alta escalabilidad en los recursos son escasos y la integridad de los datos no es lo más importante, como sí ocurre en cambio en aplicaciones especializadas por ejemplo entransacciones bancarias.



¿Cómo hago las transformaciones?

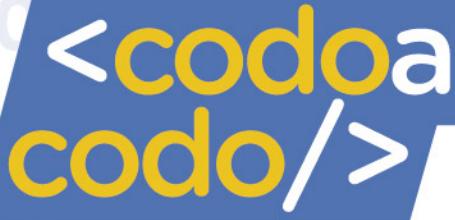
Todas las clases que representen entidades de mi sistema, y no sean solo objetos software creados para llevar a cabo alguna acción en particular, se transforman en tablas, donde los atributos de la clase pasan a ser campos de la tabla, y se selecciona el campo identificador que debe cumplir con dichas características, creando uno específicamente para cumplir con dicha función en caso de ser necesario (recordar que las clases no necesitan un identificador único).

A su vez, todas las operaciones no son persistidas en las tablas ya que en el modelo lógico de datos no hay operaciones, solo datos.

En cuanto a las asociaciones, dependerá de la cardinalidad de estas. En los casos donde haya una asociación ya sea una agregación o composición de uno a muchos, la tabla que represente la clase que aparecía con la cardinalidad 1 poseerá una clave primaria, y la otra tabla además de tener una clave primaria tendrá una clave foránea a la tabla antes mencionada. Cuando las asociaciones sean de muchos a muchos, para poder generar este tipo de relación en tablas, se deberá generar una tabla intermedia que represente solamente a las relaciones y tenga las claves primarias de las tablas relacionadas.

Por último, cuando tengamos una relación de generalización (que se implementará a través de la herencia en algún lenguaje de programación), existirán tres formas de llevar a cabo dicha relación en el manejo de tablas:

1. Se pasan los atributos de la superclase a cada una de las subclases (los atributos se repetirán en cada una de las subclases), y la superclase no se representa como tabla.
2. Se pasan los atributos de las subclases a la superclase, que contendrá todos los atributos de manera opcional, y las subclases no se representan como tablas.
3. La superclase y las subclases se transforman en tablas, y la clave primaria de las subclases es la misma que la de la superclase, teniendo estas subclases una clave que es primaria y foránea al mismo tiempo.



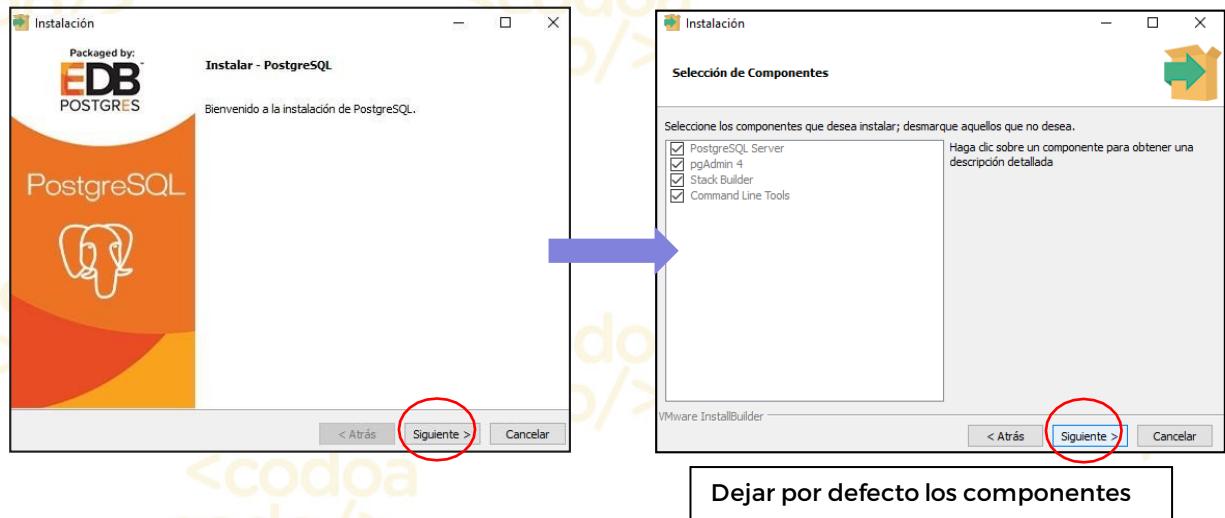
Instalar PostgreSQL

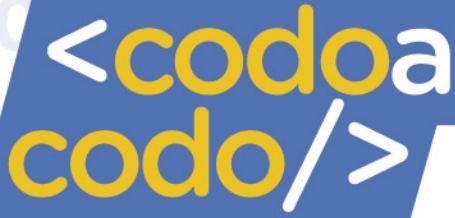
1. Descargar PostgreSQL

Descargar el instalador del programa de la página oficial <https://www.postgresql.org/download>, elegir el instalador certificado por EnterpriseDB de acuerdo a su sistema operativo. En este instructivo se instalará en Windows. Este instalador incluye el servidor PostgreSQL, pgAdmin la cual es una herramienta gráfica para administrar y desarrollar las bases de datos, y StackBuilder que es un administrador de paquetes que se puede utilizar para instalar herramientas y controladores PostgreSQL adicionales. StackBuilder incluye administración, integración y migración, replicación, geoespacial, conectores y otras herramientas.

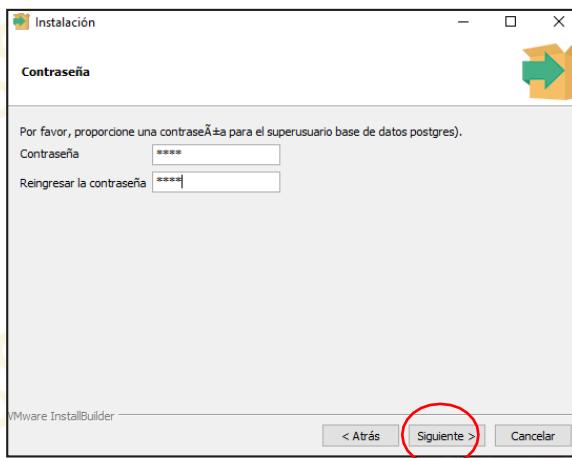
2. Instalación PostgreSQL y PgAdmin

Ejecutar el archivo descargado, y dar siguiente a las ventanas (tenga en cuenta que las mismas pueden haber variado luego de haberse creado este tutorial, pero el mismo servirá como guía para los puntos principales):

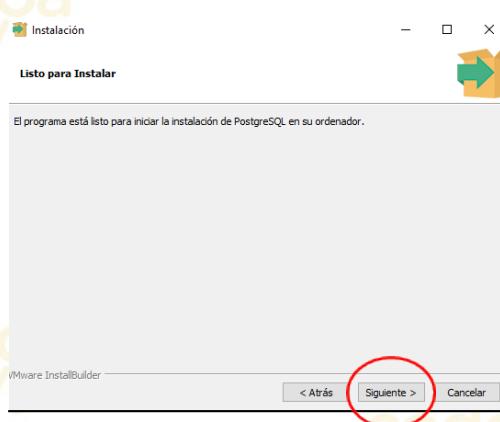
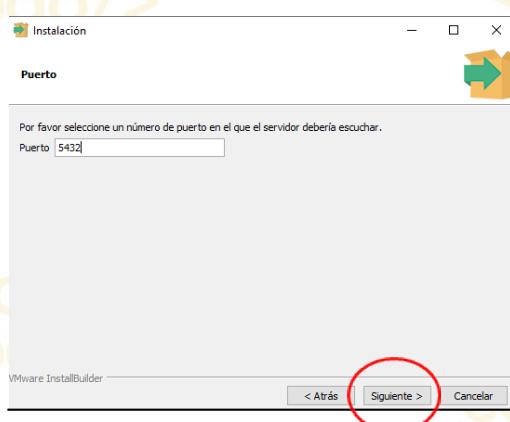




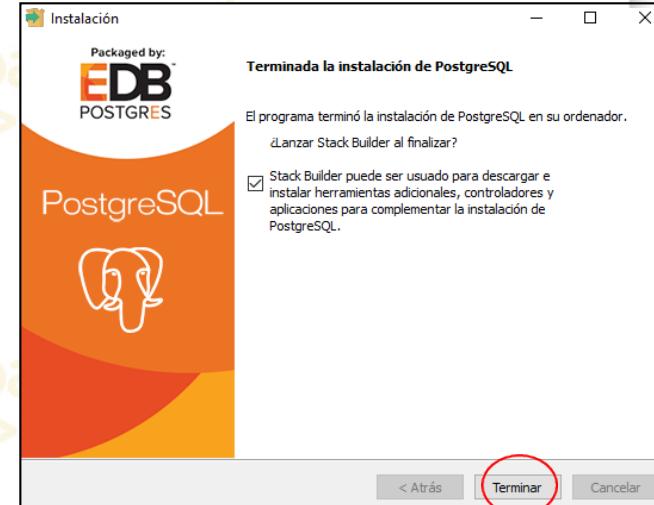
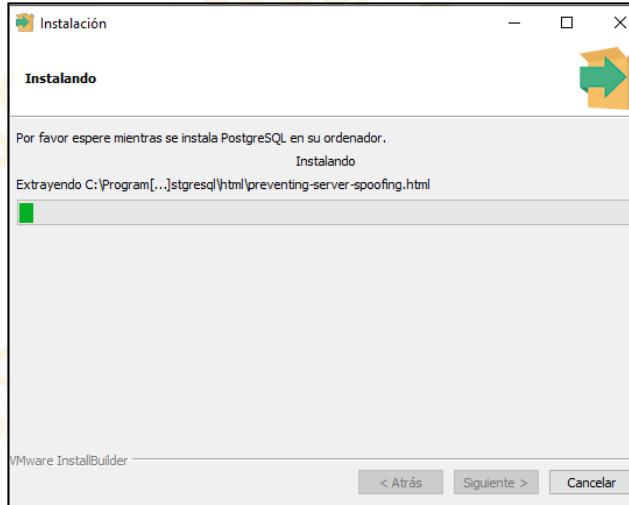
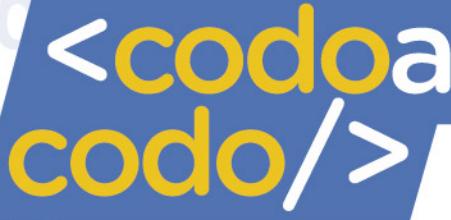
A continuación, elegir una contraseña propia para el usuario de postgres de la base de datos. Es importante porque es la que se va a usar para conectar a la Base de Datos como usuarios. Como ejemplo **1234**



Dejar por defecto el puerto de conexión 5432 y la configuración regional



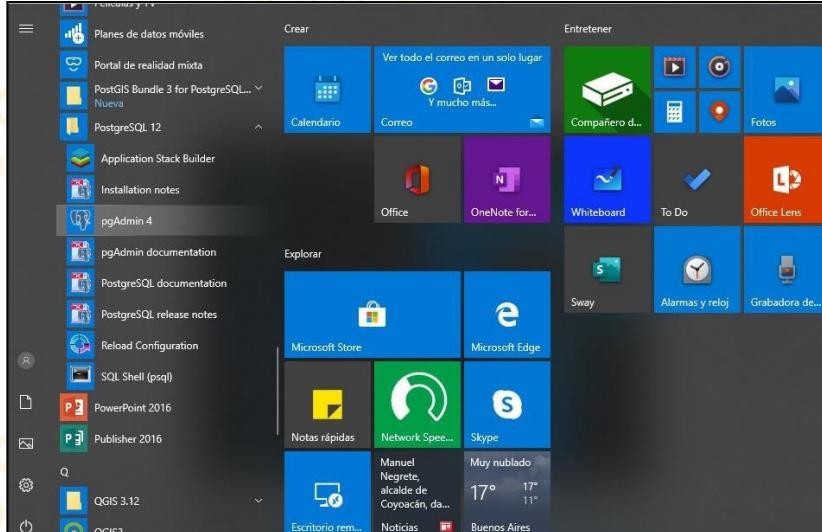
Una vez configuradas las opciones de instalación, dar siguiente y por último terminar.



3. Administración de base de datos con PgAdmin 4

La forma más conveniente de administrar la base de datos PostgreSQL es utilizar la GUI pgAdmin4. Esta herramienta se instala automáticamente durante la instalación PostgreSQL.

Se ejecuta desde la barra de programas

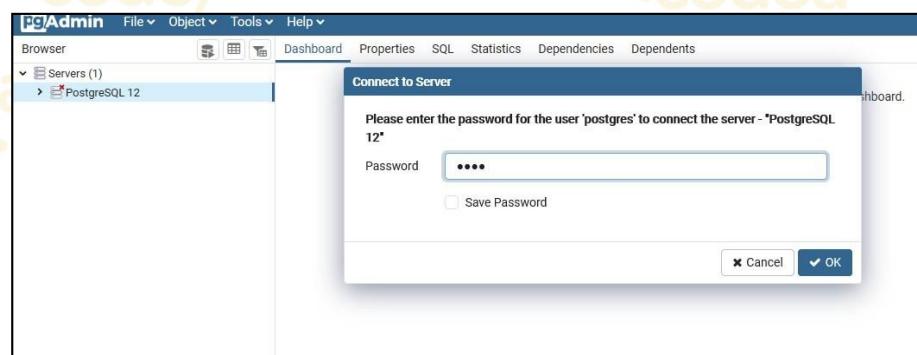




Cuando inicia la aplicación solicita una contraseña para ser usada como administrador (**1234**)

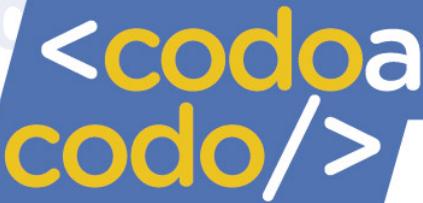


La primera pantalla permite conectarse al servidor y a la base de datos deseados. En la margen izquierda dar clic sobre Server y en PostgreSQL 12 (recuerde que esto cambiará dependiendo del momento en que esté utilizando este tutorial y la versión de postgresql que haya instalado), inmediatamente pide ingresar la contraseña para el usuario Postgres configurada en la instalación (**1234**)



En el momento de la Instalación, se crean automáticamente unas bases de datos:

- Una base de datos PostgreSQL clásicamente llamada **postgres**



Repaso SQL

1.1 Componentes del SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

1.2 Comandos

Existen dos tipos de comandos SQL:

- Los que permiten crear y definir nuevas bases de datos, campos e índices (DDL).
CREATE Utilizado para crear nuevas tablas, campos e índices
DROP Empleado para eliminar tablas e índices
ALTER Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.
- Los que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos (DML). **SELECT** Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado **INSERT** Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE Utilizado para modificar los valores de los campos y registros especificados **DELETE** Utilizado para eliminar registros de una tabla de una base de datos

1.3 Cláusulas

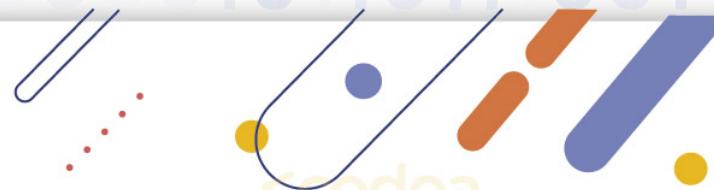
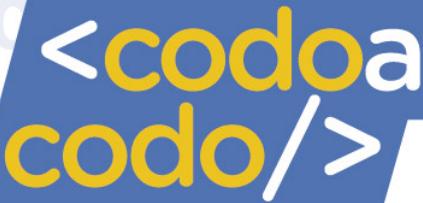
Las cláusulas son condiciones utilizadas para concretar que datos son los que se desea seleccionar o manipular.

FROM Utilizada para especificar la tabla de la cual se van a seleccionar los registros

WHERE Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar

GROUP BY Utilizada para clasificar los registros seleccionados en grupos específicos

HAVING Utilizada para expresar la condición que debe satisfacer cada grupo



ORDER BY Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

1.4 Operadores Lógicos

AND Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.

OR Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta. **NOT** Devuelve el valor contrario de la expresión.

1.5 Operadores de Comparación

< Menor que

> Mayor que

<> Distinto de

<= Menor ó Igual que

>= Mayor ó Igual que

= Igual que

BETWEEN Utilizado para especificar un intervalo de valores.

o **LIKE** Para la comparación de una cadena de texto con una expresión regular

1.6 Funciones de Agregación

Las funciones de agregación se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

AVG Utilizada para calcular el promedio de los valores de un campo determinado

COUNT Utilizada para devolver el número de registros de la selección

SUM Utilizada para devolver la suma de todos los valores de un campo determinado

MAX Utilizada para devolver el valor más alto de un campo especificado

MIN Utilizada para devolver el valor más bajo de un campo especificado