



PostgreSQL

The world's most advanced
open source database.

PREVIEWED

05/2016

EXE

EXE

www.exe.cl

Lo básico





¿Qué es postgres?

- **PostgreSQL** es un sistema de gestión de bases de datos objeto-relacional (¿Documetal, hstore, etc..?) distribuido bajo licencia BSD y con su código fuente disponible libremente (Open Source)
- **Ingres 1977-1985** - "El comienzo". Michael Stonebraker.
- **En Michel Stonebraker regresa en 1985 con nuevas ideas.**
 - Proyecto Postgres (después de Ingres) patrocinado por la Defense Advanced Research Projects Agency (DARPA), la Army Research Office (ARO), la National Science Foundation (NSF), y ESL, Inc.
- **Postgres95 1994-1995** - Nueva vida en el mundo opensource
 - En 1994, dos estudiantes de Berkeley, Andrew Yu y Jolly Chen, empezaron a trabajar con el código de Postgres (versión 4.2) y llamaron al proyecto Postgres95. Hicieron una limpieza general del código, arreglaron errores en el mismo, e implementaron otras mejoras
- **PostgreSQL 1996-actualidad** - Proyecto PostgreSQL



Características generales

- Es una base de datos 100% ACID
- Integridad referencial
- Tablespace
- Nested transactions (savepoints)
- Replicación asincrónica/sincrónica / Streaming replication - Hot Standby
- Two-phase commit
- PITR - point in time recovery
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)
- Múltiples métodos de autenticación
- Acceso encriptado via SSL
- Actualización in-situ integrada (pg_upgrade)
- Completa documentación
- Licencia BSD
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit



Instalación



- Las distribuciones de postgres se encuentran en
 - <http://www.postgresql.org/download/>
- Siempre es posible compilar los fuentes
 - No es buena idea
- Para Windows:
 - Existe un instalador gráfico
 - Incluye:
 - PostgreSQL server
 - pgAdmin III una herramienta gráfica de administración
 - StackBuilder un administrador de paquetes
- Para OSX
 - Existe un instalador gráfico
 - Se puede instalar con Brew
 - Para desarrollo lo más fácil es Postgres.app (Distribución completa incluye javascript, postgist, etc.)
 - <http://postgresapp.com/>



Distribuciones y Forks notables



- **PostgreSQL Live CD. Basado en Fedora**
- **2ndQuadrant Unified Data Analytics (2UDA)** es una suite de data analytics que une bases de datos, planillas, minería de datos y visualización.
- **La licencia de PostgreSQL es del estilo MIT/BSD, lo cual lo hace un gran candidato para forks**
 - Netezza inicialmente, Amazon Redshift (Fork de un fork)
- **Propietarios**
 - ENTERPRISE DB, fork con sintaxis Oracle
- **Open Source**
 - BigSQL, provee un paquete de Hadoop con Postgres,
 - Postgres-XL. Es base datos para operaciones masivas en paralelo (MPP) construida sobre PostgreSQL 9.5. Escalamiento lineal (Casi)
 - tPosgres
 - PostgresXC



Versiones Actuales

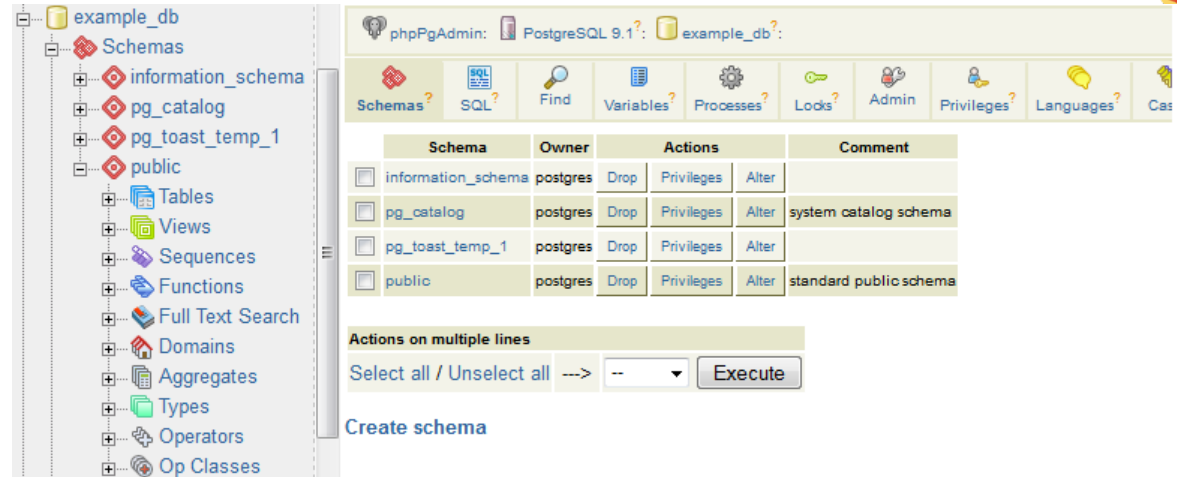


- **9.5.3**
 - 9.6beta1
- **Postgis 2.2.2**
- **Postgrest 0.3.1.1**



Herramientas de administración

- PSQL
- pgAdmin
- phpPgAdmin
- Adminer
 - Es una opción para administrar múltiples bases de datos





Objetos de la base de datos

- Service**

- PostgreSQL se instala como un servicio (daemon) a nivel de S.O. Es factible ejecutar más de un servicio, en diferentes puertos con un almacenamiento de datos distintos. Habitualmente no se distingue entre servidor y servicio.

- Database**

- Cada servicio PostgreSQL contiene muchas bases de datos.

- Schema**

- Los schemas (Esquemas) son parte del estándar SQL. Son el siguiente nivel de organización al interior de una base de datos (Como un estado dentro de un país). La mayoría de los objetos de la base de datos pertenecen a un esquema.
 - PostgreSQL crea automáticamente un esquema llamado public al crear una base de datos. Por omisión (Default) todo lo que se crea queda en el esquema public (Más adelante veremos como cambiar esto).
 - Si las tablas no son muchas no es necesario crear más esquemas, pero si hablamos de miles de tablas, entonces es necesario crear diferentes esquemas y agrupar lógicamente tablas en esos esquemas.



Objetos de la base de datos

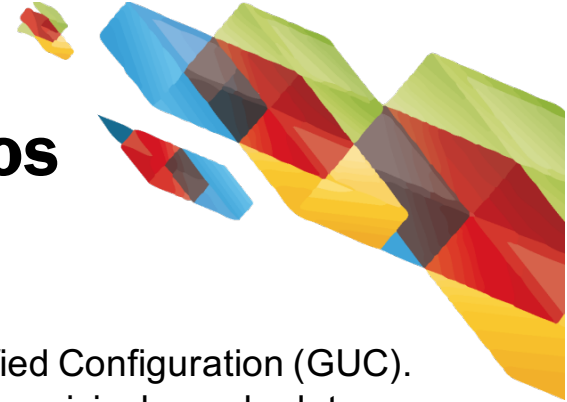


•Catalog

- Los catálogos son esquemas del sistema donde PostgreSQL almacena las funciones que ya vienen incorporadas y la metadata.
- Al crear una base de datos existen dos catálogos: **pg_catalog**, que contiene todas las funciones las tablas, las vistas del sistemas y los tipos que vienen preempaquetados con PostgreSQL e **information_schema**, que contiene las vistas que forman parte del estándar ANSI para tener acceso a la metainformación de PostgreSQL en el formato que establece el dicho estándar.
- Todos los parámetros de ajuste y afinamiento del servidor están en tablas del sistema, en el catálogo **pg_catalog** las cuales se pueden consultar y modificar (Requiere privilegios de super usuario). Al modifica evidentemente es posible alterar el funcionamiento normal de postgres.
- En el catálogo **information_schema** se encuentra la misma información que se encuentra en otros servidores como MySQL o SQL Server. Las vistas más usadas son columns, que contiene todas las columnas de una base de datos, tables que contiene todas las tablas y las vistas (Y su SQL asociado). A las columnas estándares PostgreSQL agrega algunas columnas adicionales como columns.udt_name, para describir los tipos personalizados de datos. Aún cuando las columnas, tablas, y vistas son todas vistas de PostgreSQL, pgAdmin las muestra en la rama information_schema→Catalog Objects.



Objetos de la base de datos



- variable

- Las variables son parte de lo que PostgreSQL llama Grand Unified Configuration (GUC).
- Las variables son opciones que pueden ser seteadas a nivel de servicio, base de datos y otros niveles. Una de las opciones (variable) más utilizadas es **search_path**, que establece los nombres de los esquemas por omisión y que ahorra el uso del nombre del esquema como prefijo al acceder a los objetos.



Objetos de la base de datos



•extension

- Las extensiones se agregaron a partir de PostgreSQL 9.1 y permiten a los desarrolladores empaquetar funciones, tipos de datos, índices personalizados, tablas y GUCs.
- Las extensiones se instalan y desinstalan como una unidad.
- Son similares a las extensiones de Oracle y permiten distribuir add-ons.
- Las extensiones se instalan en el servidor. Esto involucra instalar binarios y ejecutar scripts.
- Las extensiones se habilitan para cada base de datos. Por ejemplo si se requiere de búsqueda avanzada de textos se debe habilitar la extensión **fuzzystr match** para la base de datos donde se requiere.
- Es factible al momento de agregar una extensión escoger el esquema en el cual se instalarán. Por omisión son instaladas en el esquema **public**.
- Se recomienda instalar las extensiones en un esquema aparte o en su propio esquema, en el caso de que sean muy grandes, para evitar mezclar los objetos de las extensiones con los objetos propios. Se recomienda incluir esos esquemas en la variable **search_path** de la base de datos, de tal forma de poder usar las funciones sin necesidad de indicar el esquema.
- Existen extensiones en que no es posible cambiar el esquema de instalación y demandan ser instaladas en un esquema en particular. Por ejemplo las extensiones de lenguaje, como **plv8**, se instalan en el **pg_catalog**



Objetos de la base de datos



- table**

- Las tablas son la esencia de una base de datos. En PostgreSQL, las tablas son ciudadanos de primera clase en sus esquemas.
- Las tablas en Postgres reconocen sus padres y sus hijos (Son jerárquicas)
- Estas jerarquías permiten implementar herencia
- Al momento de crear una tabla, se crea automáticamente un tipo de dato personalizado con la estructura completa de la tabla. Es decir, es factible definir una columna que tenga la estructura de una tabla que ya existe en la base de datos.

- foreign table y foreign data wrapper**

- Las tablas foráneas fueron lanzadas en la versión 9.1.
- Son tablas virtuales que están enlazadas a datos fuera de la base de datos. Una vez hecho el enlace se comportan como tablas. Una tabla foránea puede estar enlazada a un archivo CSV o a una tabla que se encuentra en otro servidor PostgreSQL o incluso a una tabla en un servidor SQL Server u Oracle, a una base de datos NoSQL como Redis, o un web service provisto por Twitter o Salesforce.
- La configuración de las tablas foráneas se hace a través de foreign data wrappers (FDWs). FDWs contienen el enlace mágico entre PostgreSQL y la fuente de datos externa.



Objetos de la base de datos

- **foreign table and foreign data wrapper**

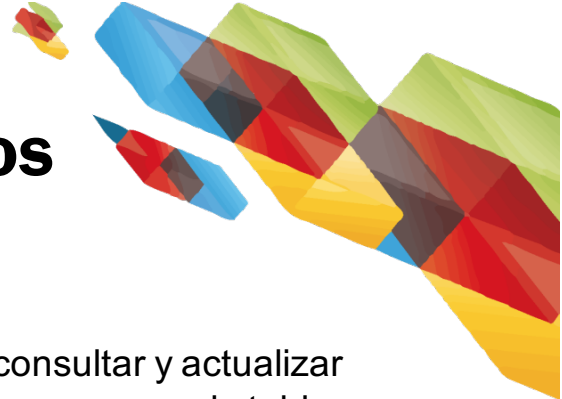
- Su implementación sigue el estándar establecido por SQL/Management of External Data (MED).
- Existen muchos FDWs para fuentes de datos muy usadas o populares y que son de libre uso.
- Los FDWs se instalan como extensiones
- Una vez instaladas pgAdmin las muestra en el nodo llamado Foreign Data Wrappers.

- **tablespace**

- Un espacio de tablas es la ubicación física donde se almacenan los datos
- Esto permite mover bases de datos, o tablas de ubicación física sin mayores problemas



Objetos de la base de datos



- view

- Las vistas son una manera de generar abstracciones lógicas para consultar y actualizar datos. En el caso de vistas "**autoupdatable**", estas deben involucrar a una sola tabla (Característica disponible a partir de Postgres 9.3)
- En el caso de que la vista involucre a más de una tabla, se hace necesario implementar triggers para hacerla actualizable.
- A partir de la versión 9.3, se cuenta con **materialized views**, que permiten hacer caché con la data más usada en consultas.

- function

- Las funciones en PostgreSQL pueden retornar un valor o un conjunto de valores. También es factible escribir funciones que manipulen los datos. Cuando una función realiza esta última tarea se le suele llamar procedimiento almacenado.

- language

- Las funciones son creadas en lenguajes para procedimientos, procedural languages (PLs). Out of the box soporta tres: SQL, PL/pgSQL, y C. Es factible instalar lenguajes adicionales utilizando los comandos CREATE EXTENSION o CREATE PROCEDURAL LANGUAGE.
- Los lenguajes más utilizados actualmente son Python, JavaScript, Perl, y R.



Objetos de la base de datos



•cast

- Los casts permiten convertir valores de un tipo a valores de otro tipo. Las conversiones son realizadas por funciones.
- Una característica especial de postgres, es que es factible crear conversiones.
- Por ejemplo es factible convertir un entero a un zip code (5 dígitos de largo) agregando un cero a los números entre el 1000 y el 9999.
- Las conversiones pueden ser implícitas o explícitas. Las implícitas van desde tipos específicos a tipos más generales.
- En otras situaciones se requieren conversiones explícitas.

•sequence

- Una secuencia controla el autoincremento de un tipo de dato serial.
- PostgreSQL automáticamente crea secuencias al definir una columna serial
- Es factible cambiar el valor inicial, el incremento y el siguiente valor de una secuencia.
- Las secuencias son objetos y pueden ser usados en más de una tabla (Id único, por ejemplo)
- SQL Server y Oracle tienen secuencias pero deben ser creadas manualmente.



Objetos de la base de datos



- row o record

- Para efectos de esta clase los términos fila o registro son lo mismo.
- En PostgreSQL, las filas pueden ser tratadas o manejadas de manera independiente a la tabla. Esto es útil a la hora de escribir funciones o utilizar el constructor de filas en SQL.

- trigger

- Los gatillos o triggers son un elemento común en las bases de datos corporativas y lo que hacen es reaccionar frente a un evento asociado a un cambio en los datos.
- Cuando PostgreSQL lanza un evento, el gatillo se puede activar antes o después de que el evento ocurra
- A partir de la versión 9.0 PostgreSQL es posible evaluar una condición WHEN para ver si debe activar el gatillo frente a un evento.
- La versión 9.0 también introdujo la cláusula UPDATE OF clause, que permite monitorear los cambios de ciertas columnas
- En la versión 9.1, un cambio de datos en una vista puede activar un gatillo.
- Cambios en el DDL, a partir de la versión 9.3 activan gatillos.
- En la versión 9.4 cambios en tablas foráneas también activan gatillos.



Objetos de la base de datos

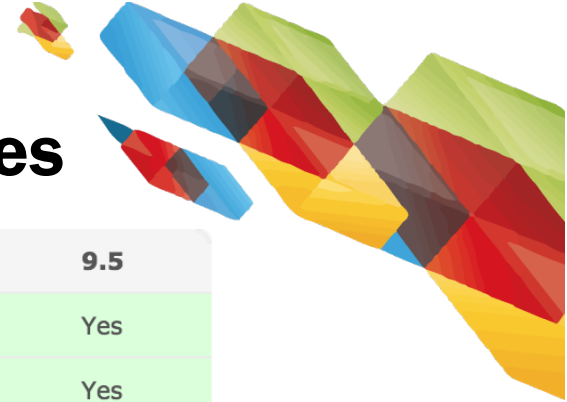


•Rule

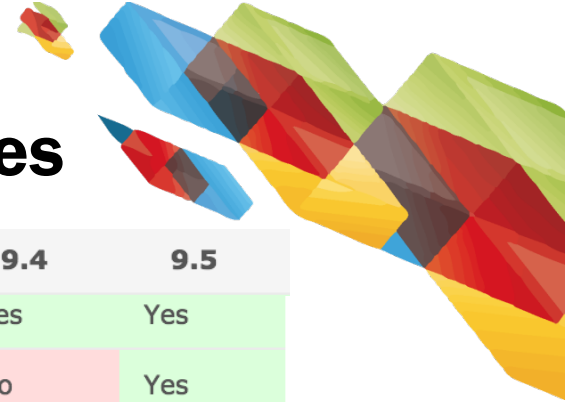
- Las reglas son instrucciones que permiten sustituir una acción por otra.
- PostgreSQL utiliza reglas internamente para definir vistas.
- Ejemplo: `CREATE VIEW vw_alumnos AS SELECT * FROM alumnos WHERE activo;`
Internamente, PostgreSQL agrega una regla `INSTEAD OF SELECT` que establece que cuando trates de hacer un select from una tabla llamada `vw_alumnos`, solo se devolverán filas de la tabla `alumnos` cuyo valor para el campo `activo` sea verdadero.
- Una regla puede ser útil para trigger, en el caso de eventos simples asociados a `update/insert/delete`. La regla puede insertar comandos SQL adicionales antes del original. Eso evita el overhead de tocar cada registro por separado.
- Las reglas son difíciles de depurar por eso su uso no es tan extendido como los PLs.



Diferencias en las versiones



Backend	9.1	9.2	9.3	9.4	9.5
64-bit large objects	No	No	Yes	Yes	Yes
ALTER object IF EXISTS	No	Yes	Yes	Yes	Yes
ALTER SYSTEM	No	No	No	Yes	Yes
ALTER TABLE ... SET LOGGED / UNLOGGED	No	No	No	No	Yes
Cascading streaming replication	No	Yes	Yes	Yes	Yes
Checksum on data pages	No	No	Yes	Yes	Yes
COPY from/to STDIN/STDOUT	No	No	Yes	Yes	Yes
CREATE TABLE ... (LIKE) with foreign tables, views and composite types	No	Yes	Yes	Yes	Yes
Custom background workers	No	No	Yes	Yes	Yes
Dynamic Background Workers	No	No	No	Yes	Yes
Foreign table inheritance	No	No	No	No	Yes
GROUPING SETS, CUBE and ROLLUP support	No	No	No	No	Yes
IMPORT FOREIGN SCHEMA	No	No	No	No	Yes

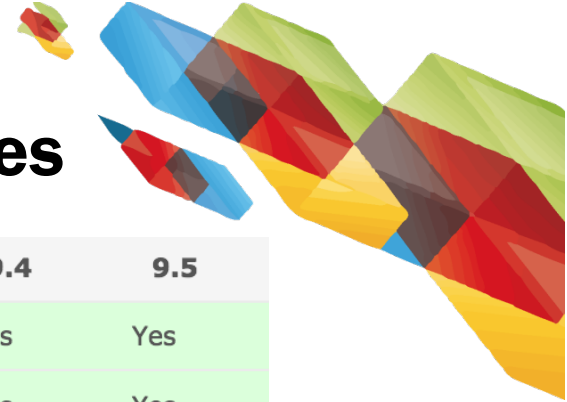


Diferencias en las versiones

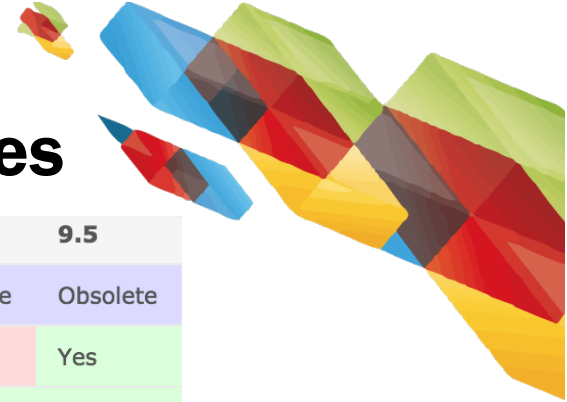
Backend	9.1	9.2	9.3	9.4	9.5
Improved set of JSON functions and operators	No	No	Yes	Yes	Yes
INSERT ... ON CONFLICT DO NOTHING/UPDATE	No	No	No	No	Yes
JSONB-modifying operators and functions	No	No	No	No	Yes
Materialized views with concurrent refresh	No	No	No	Yes	Yes
min_wal_size / max_wal_size	No	No	No	No	Yes
Parallel pg_dump	No	No	Yes	Yes	Yes
pg_receivexlog tool	No	Yes	Yes	Yes	Yes
Replication Slots	No	No	No	Yes	Yes
SELECT FOR NO KEY UPDATE/SELECT FOR KEY SHARE lock modes	No	No	Yes	Yes	Yes
SP-GiST indexes for range types	No	No	Yes	Yes	Yes
Streaming-only cascading replication	No	No	Yes	Yes	Yes



Diferencias en las versiones

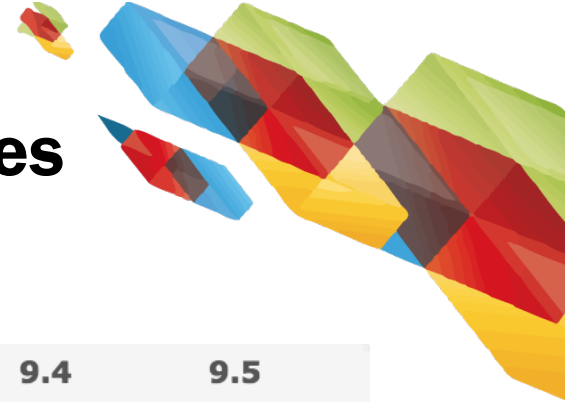


Backend	9.1	9.2	9.3	9.4	9.5
Support for anonymous shared memory	No	No	Yes	Yes	Yes
Time-delayed Standbys	No	No	No	Yes	Yes
Updatable views	No	No	Yes	Yes	Yes
WITH CHECK clause	No	No	No	Yes	Yes
WITHIN GROUP clause	No	No	No	Yes	Yes
WITH ORDINALITY clause	No	No	No	Yes	Yes
Writable Foreign Data Wrappers	No	No	Yes	Yes	Yes



Diferencias en las versiones

Security	9.1	9.2	9.3	9.4	9.5
krb5 authentication (without gssapi)	Yes	Yes	Yes	Obsolete	Obsolete
Row-Level Security	No	No	No	No	Yes
security_barrier option on views	No	Yes	Yes	Yes	Yes
Network	9.1	9.2	9.3	9.4	9.5
Internationalisation	9.1	9.2	9.3	9.4	9.5
Platforms	9.1	9.2	9.3	9.4	9.5
Datatypes	9.1	9.2	9.3	9.4	9.5
JSONB data type	No	No	No	Yes	Yes
JSON data type	No	Yes	Yes	Yes	Yes
Range types	No	Yes	Yes	Yes	Yes
smallserial type	No	Yes	Yes	Yes	Yes
Functions and triggers	9.1	9.2	9.3	9.4	9.5
Event triggers	No	No	Yes	Yes	Yes
FILTER clause for aggregate functions	No	No	No	Yes	Yes



Diferencias en las versiones

Procedural Languages	9.1	9.2	9.3	9.4	9.5
CREATE TRANSFORM	No	No	No	No	Yes
contrib modules	9.1	9.2	9.3	9.4	9.5
contrib/pg_rewind	No	No	No	No	Yes
contrib/pg_stat_statements improvements	No	Yes	Yes	Yes	Yes
contrib/pg_trgm regular expressions indexing	No	No	Yes	Yes	Yes
contrib/pg_xlogdump	No	No	Yes	Yes	Yes
contrib/postgres_fdw	No	No	Yes	Yes	Yes
contrib/tcn	No	Yes	Yes	Yes	Yes



Ciclo de vida y soporte

Versión	Versión menor	Soportada	Lanzamiento	Soporte
9.2	9.2.0	Si	Sep 2012	Sep 2017
9.1	9.1.5	Si	Sep 2011	Sep 2016
9.0	9.0.9	Si	Sep 2010	Sep 2015
8.4	8.4.13	Si	Jul 2009	Jul 2014
8.3	8.3.20	Si	Feb 2008	Feb 2013
8.2	8.2.23	No	Dic 2006	Dic 2011
8.1	8.1.23	No	Nov 2005	Nov 2010
8.0	8.0.26	No	Ene 2005	Oct 2010
7.4	7.4.30	No	Nov 2003	Oct 2010
7.3	7.3.21	No	Nov 2002	Nov 2007
7.2	7.2.8	No	Feb 2002	Feb 2007
7.1	7.1.3	No	Abr 2001	Abr 2006
7.0	7.0.3	No	May 2000	May 2005
6.5	6.5.3	No	Jun 1999	Jun 2004
6.4	6.4.2	No	Oct 1998	Oct 2003
6.3	6.3.2	No	Mar 1998	Mar 2003



Drivers



- **PHP:** Dos drivers el antiguo, pgsql driver y el nuevo pdo_pgsql. Se habilitan en el php.ini, usualmente ya están instalados
- Los desarrollos en JAVA, cuentan con drivers JDBC's que se descargan de la página de Postgres
- Para .NET (Microsoft o Mono), se usa el driver Npgsql. Fuentes y binarios están disponibles para .NET Framework 3.5 y posteriores, Microsoft Entity Framework, y Mono.NET.
- Para conexión desde Microsoft Access, Office o vía Open Database Connectivity (ODBC), los drivers de 32 y 64 bits están disponibles en la página de PostgreSQL.
- LibreOffice 3.5 (y posteriores) tienen incorporado un driver nativo de PostgreSQL . Para versiones más antiguas de Libre Office y para OpenOffice se utiliza el driver JDBC.
- Python brinda soporte para PostgreSQL a través de varios drivers; psycopg es el más popular. También existe soporte para Postgres en Django
- En Ruby, se utiliza rubygems pg.
- La conectividad en Perl para PostgreSQL es a través de los drivers DBI y DBD::Pg. Existe un driver 100% Perl, DBD::PgPP.
- En Node.js existen 3 drivers Node Postgres, Node Postgres Pure y Node-DBI.
- En scala está disponible un Driver asíncrono, escrito utilizando Netty, **postgresql-async**.



EXPLORAR una Base de Datos