**Universidad Técnica Nacional**

Gestor de Base de Datos



Andrés Alberto Ávila Brenes

Sebastián Campos Salas

María Fernanda Murillo Alfaro

José Fabio Valerio Acuña

Abril 2018.

Universidad Técnica Nacional

Ingeniería del software

Aplicación de bases de datos

**Tabla de Contenidos**

[Resumen Ejecutivo 1](#_Toc512342703)

[Objetivo General 2](#_Toc512342704)

[Objetivos Específicos 2](#_Toc512342705)

[Introducción 3](#_Toc512342706)

[Descripción del Problema 4](#_Toc512342707)

[Desarrollo 5](#_Toc512342708)

[Historia 5](#_Toc512342709)

[Características 6](#_Toc512342710)

[Ventajas 6](#_Toc512342711)

[Desventajas 7](#_Toc512342712)

[¿Cuál es su funcionalidad? 8](#_Toc512342713)

[Recomendaciones 8](#_Toc512342714)

[Conclusiones 8](#_Toc512342715)

[Lista de referencias 10](#_Toc512342716)

# Resumen Ejecutivo

Se realiza un administrador de base de datos, utilizando Visual Studio como IDE, C# como lenguaje principal y PgAdmin4. La idea principal es generar una versión propia de PgAdmin4, la cual deberá tener las mismas funciones que el software original, esta versión de PgAdmin4 deberá soportar consultas grandes, realizar triggers, funciones y demás.

Al realizar este proyecto se desarrolla más lógica, se aprende a trabajar desde lo más básico para crear una base de datos, hablamos de diccionarios de datos, una vez que se aprende a utilizarlos es fácil manejar las bases de datos.

# Objetivo General

1. Desarrollar un gestor de base de datos, que se molde a las necesidades comerciales y que de abasto a base de datos robustas.

# Objetivos Específicos

1. Generar consultas desde la interfaz gráfica con resultados óptimos.
2. Mostrar las bases de datos que se tienen de PostgreSQL y mostrarlas en la interfaz gráfica.
3. Crear, modificar y eliminar bases de datos.

# Introducción

Primeramente, el modelo de base de datos a seguir es PgAdmin4, un modelo muy estable y seguro, en cuanto a seguridad no es recomendado. Esta versión propia de PgAdmin4 es muy similar al software original, un poco más amigable y accesible para cualquier tipo de usuario.

Al desarrollar este tipo de proyecto se obtiene mucho conocimiento, ya que se tuvo que investigar un 90% de este, la curva de aprendizaje es muy significativa.

Con este gestor de base de datos, se desea resultados óptimos, que soporte base de datos robustas y realice cualquier consulta al instante.

# Descripción del Problema

Se debe desarrollar una propuesta de administrador de base de datos, el cual soporte consultas y bases de datos robustas, donde se esperan respuestas optimas, en este caso el proyecto está inspirado en PostgreSQL

# Desarrollo

## Historia

PostgreSQL ha tenido una larga evolución, la cual se inicia en 1982 con el proyecto [Ingres](https://es.wikipedia.org/w/index.php?title=Ingres_(Base_de_Datos)&action=edit&redlink=1) en la [Universidad de Berkeley](https://es.wikipedia.org/wiki/Universidad_de_California,_Berkeley). Este proyecto, liderado por [Michael Stonebraker](https://es.wikipedia.org/wiki/Michael_Stonebraker), fue uno de los primeros intentos en implementar un motor de base de datos relacional. Después de haber trabajado un largo tiempo en *Ingres* y de haber tenido una experiencia comercial con el mismo, Michael decidió volver a la Universidad en 1985 para trabajar en un nuevo proyecto sobre la experiencia de Ingres, dicho proyecto fue llamado post-ingres o simplemente POSTGRES.

El proyecto post-ingres pretendía resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los [años 1980](https://es.wikipedia.org/wiki/A%C3%B1os_1980). El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados [objetos](https://es.wikipedia.org/wiki/Objeto_(programaci%C3%B3n)). Se esforzaron en introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos. Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones - las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario.

## Ventajas

* Ahorros considerables de costos de operación: PostgreSQL ha sido diseñado para tener un mantenimiento y ajuste menor que los productos de proveedores comerciales, conservando todas las características, estabilidad y rendimiento.
* Estabilidad y confiabilidad: No se han presentado caídas de la base de datos.
* Extensible: El código fuente está disponible de forma gratuita, para que quien necesite extender o personalizar el programa pueda hacerlo sin costes.
* Multiplataforma: Está disponible en casi cualquier Unix, con 34 plataformas en la última versión estable, además de una versión nativa de Windows en estado de prueba.
* Diseñado para ambientes de alto volumen: Utilizando una estrategia de almacenamiento de filas llamada MVCC, consigue mejor respuesta en grandes volúmenes. Además, MVCC permite a los accesos de solo lectura continuar leyendo datos consistentes durante la actualización de registros, permitiendo copias de seguridad en caliente
* Herramientas gráficas de diseño y administración de bases de datos.
* Soporta los tipos de datos, cláusulas, funciones y comandos de tipo estándar SQL92/SQL99 y extendidos propios de PostgreSQL.
* Puede operar sobre distintas plataformas, incluyendo Linux, Windows, Unix, Solaris y MacOS X.
* Buen sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas.
* Gran capacidad de almacenamiento.
* Buena escalabilidad ya que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima, soportando una mayor cantidad de peticiones simultáneas a la base de datos de forma correcta.

## Desventajas

* En comparación con MySQL es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene MySQL.
* Soporte en línea: Hay foros oficiales, pero no hay una ayuda obligatoria.
* Consume más recursos que MySQL.
* La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.

# Recomendaciones

Mejorar la interfaz grafica para una mejor interacion con los usuarios

# Conclusiones

Este proyecto fue de gran provecho en cuanto a la obtención de conocimiento, ya que se tuvo que investigar bastante durante su desarrollo. Con este proyecto se aprendió a manejar diccionarios de datos, también se aprendió cómo funcionan las bases de datos desde otra perspectiva.

Al realizar una versión propia de PgAdmin4 ha sido una experiencia tediosa y a la vez maravillosa, se debió programar cada una de las consultas y saber manejarlas en código para ejecutarlas en querys y que a la vez extrajera los datos desde PostgreSQL.

# Lista de referencias

.