Instituto Tecnológico de Costa Rica

Unidad de Computación

“Proyecto 3 Bases de Datos II”

Marco Antonio Espinoza Salas

Allen Antonio Jiménez González

Lester Alexander Trejos Bermúdez

Sede San Carlos

18/12/2020

**Tabla de contenidos**

Contenido

[Introducción 4](#_Toc47144606)

[Análisis del problema 6](#_Toc47144607)

[Requerimiento #1: 6](#_Toc47144608)

[Requerimiento #2: 6](#_Toc47144609)

[Requerimiento #3: 6](#_Toc47144610)

[Requerimiento #4: 6](#_Toc47144611)

[Requerimiento #5: 6](#_Toc47144612)

[Requerimiento #6: 7](#_Toc47144613)

[Requerimiento #7: 7](#_Toc47144614)

[Solución del problema 8](#_Toc47144615)

[Solución al requerimiento #1: 8](#_Toc47144616)

[Solución al requerimiento #2: 9](#_Toc47144617)

[Solución al requerimiento #3: 10](#_Toc47144618)

[Solución al requerimiento #4: 10](#_Toc47144619)

[Solución al requerimiento #5: 12](#_Toc47144620)

[Solución al requerimiento #6: 13](#_Toc47144621)

[Solución al requerimiento #7: 14](#_Toc47144622)

[Análisis de resultados 15](#_Toc47144623)

[Análisis de resultados del primer requerimiento: 16](#_Toc47144624)

[Análisis de resultados del segundo requerimiento: 16](#_Toc47144625)

[Análisis de resultados del tercer requerimiento: 18](#_Toc47144626)

[Análisis de resultados del cuarto requerimiento: 18](#_Toc47144627)

[Análisis de resultados del quinto requerimiento: 20](#_Toc47144628)

[Análisis de resultados del sexto requerimiento: 21](#_Toc47144629)

[Análisis de resultados del séptimo requerimiento: 23](#_Toc47144630)

[Conclusiones 26](#_Toc47144631)

[Recomendaciones 27](#_Toc47144632)

[Referencias Bibliográficas 28](#_Toc47144633)

## 

## Introducción

El propósito de este informe es presentar las experiencias obtenidas en la realización del proyecto, que consiste en el desarrollo de consultas distribuidas , utilizando como fuente una base de datos remota o creada en el servidor local, y una base de datos central que fue suministrada por el profesor del grupo. Además, en este informe se encuentra dividido de la siguiente forma: en la sección requerimientos solicitados (Análisis del Problema) están citados punto por punto, esto para aclarar las minutas en las que se dividirá el proyecto. A su vez, la sección de las soluciones muestra la forma y el proceso en el cual se resolvió cada una de las minutas mencionadas en la sección de requerimientos solicitados. Por otra parte, en la sección de análisis, se muestran y explican los resultados obtenidos al ejecutar las respectivas estrategias planteadas en la sección de soluciones. Como antepenúltima sección, tenemos las recomendaciones, estas nos indican acciones de valiosas para un mejor desarrollo de una aplicación similar a esta en un futuro. De penúltima sección tenemos las conclusiones, las cuales dictan los puntos más significativos del proyecto, y por última sección tenemos las referencias bibliográficas, las que permite reconocer el trabajo citado a sus respectivos autores.

Este proyecto se realizó utilizando herramientas informáticas, principalmente el lenguaje de programación Python para la filtración de la información, PostgreSQL como motor de base de datos y QGIS como sistema de información geográfica.

Explicando el desarrollo del proyecto, para su creación se utilizó PostgreSQL, se comienza con la lectura de los requisitos del proyecto para su posterior desarrollo. Después de establecer los requisitos del proyecto, se comenzó con la codificación de la aplicación, al final de la codificación se realiza una revisión y corrección de los errores presentes en el programa, para que el proyecto sea fluido y no tenga complicaciones para la hora de usarlo; cabe resaltar que el usuario final puede ingresar, modificar y deshabilitar como funciones principales.

Un proyecto como este es muy necesario para la gente de hoy, ya que descentralizar la información es una forma importante de adquirir esta información. Por ejemplo: Toda la información que tiene el TEC de Cartago no es necesaria para el TEC de San Carlos, entonces, con estas bases de datos podemos tener la información pertinente de cada sede sin necesitar acceder a la de otras.

## Análisis del problema

Preliminarmente, se conoce que el proyecto consiste en elaborar unas cuantas consultas distribuidas, la cual debe cumplir una serie de requerimientos y que esta, debe hacer uso de la base de datos desarrollada en evaluaciones anteriores del curso “Bases de Datos II”.

El siguiente apartado tratará punto por punto los requerimientos solicitados para la realización de este proyecto, consiguiendo así, una explicación más detallada y amplia de lo que el proyecto consiste.

### Requerimiento #1:

Como primer requerimiento, la creación de un modelo de la base de datos del nodo local con su respectivo manejo de información descriptiva.

### Requerimiento #2:

Como segundo requerimiento, el acceso a los datos locales debe ser consumido y manipulado directamente desde la herramienta QGIS utilizando una vista geográfica.

### Requerimiento #3:

Como tercer requerimiento, la tabla nodos registra el nombre y una frase de acceso para identificar el nodo, el usuario basesII solo tienen acceso a realizar inserciones en la tabla y de consultar los nombres de los nodos.

### Requerimiento #4:

Como cuarto requerimiento, función elimina\_empresa, función crea\_empresa, función registra\_caracteristicas, vista vista\_empresas. Estas como requerimientos funcionales del sistema.

### Requerimiento #5:

Como quinto requerimiento, se deben realizar un módulo de consulta que permita establecer filtros sobre los datos de nodo local y ajustar la extensión de los resultados en el canvas de Qgis.

## Solución del problema

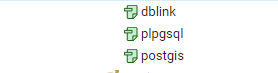
Para mantener una consistencia en el formato del presente informe, en esta sección se explicará la solución a cada requerimiento ejemplificado en el apartado anterior.

### Solución al requerimiento #1:



Imagen#1: Imagen del logo del motor de bases de datos utilizado, PostgreSQL. Fuente: https://www.muylinux.com/2018/07/09/configurar-postgresql-ruby-on-rails/

Se utilizó el motor de base de datos PostgreSQL para la realización, ya que existen extensiones para este motor para la conexión de Qgis con las bases de datos de este motor anteriormente mencionado.



Imagen#2: Extensión para la conexión con Qgis. Fuente: Elaboración propia

Para comprobar que los procedimientos creados en la base de datos que conectan con Qgis, se usó esta herramienta y se aplicaron los pasos que el profesor nos enseñó y ejecutó en los videos en clase.

Para la parte de las nuevas características como objetos que deben ser ingresados a la base de datos nodo local como parte de los atributos de estas empresas

### Solución al requerimiento #2:

Para la solución a este requerimiento, se procede con la creación de una conexión a base de datos para PostgreSQL desde Qgis con el fin de conectar las bases de datos con este sistema de información geográfica.

### Solución al requerimiento #3:

Para la solución a este requerimiento, cuando se ingresa, actualiza o eliminar una nueva empresa desde el sistema de información geográfica Qgis[[1]](#footnote-1), se levanta un trigger que ejecuta estas funciones en ambas bases de datos (nodo local y nodo central) .

### Solución al requerimiento #4:

Para solucionar este requerimiento, se crea desde base de datos los tres procedimientos y la vista, en los cuales, uno es para la generación de nuevas empresas, otro para su eliminación, para registrar sus características y la vista para ver las empresas creadas.

### Solución al requerimiento #5:

Se crea esta funcionalidad de filtro por medio del lenguaje de programación Python, y se utilizaron las librerías PyQt5 [[2]](#footnote-2)(para la creación de la interfaz gráfica de usuarios), qgis (para crear las capas para visualizar los puntos de las empresas) y psycopg2 [[3]](#footnote-3)(para hacer la conexión con la base de datos nodo local).

## Análisis de resultados

En esta sección se llevará a cabo un análisis de los resultados obtenidos a lo largo de creación del proyecto.

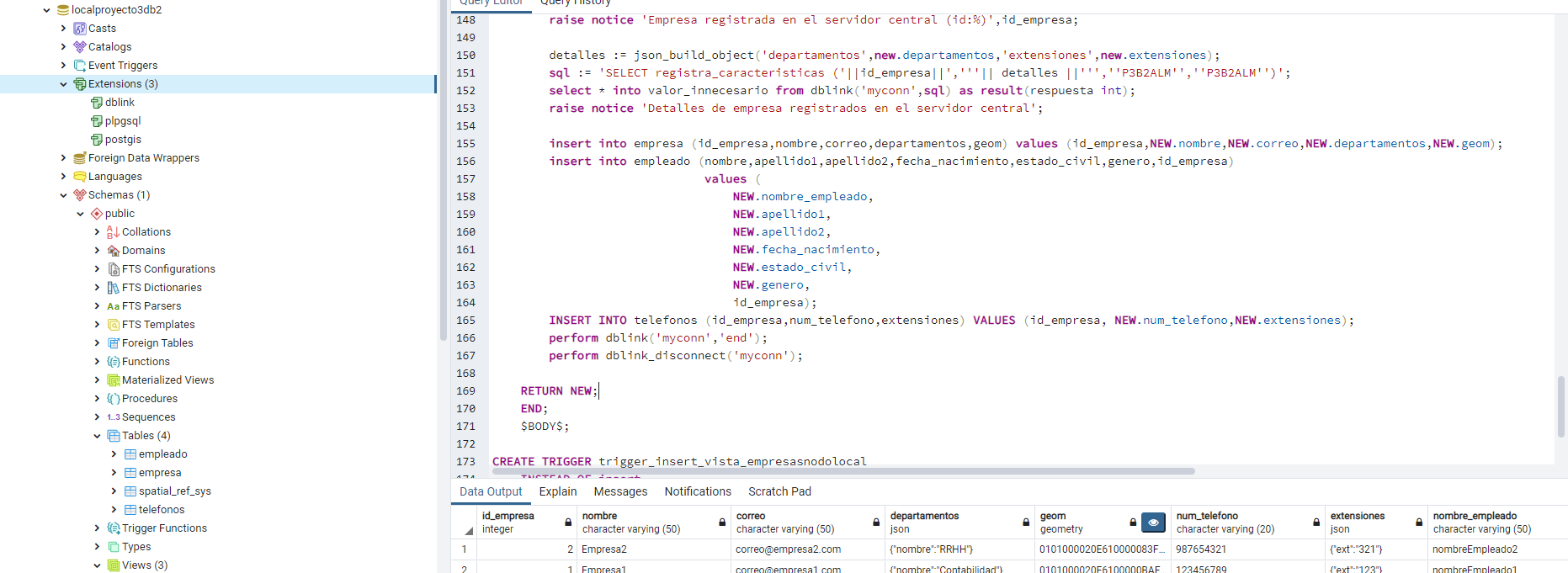
Tabla#1: Cotejo de los requerimientos, sus estados y observaciones. Fuente: Elaboración propia

|  |  |  |
| --- | --- | --- |
| Tarea/Requerimiento | Estado | Observaciones |
| Creación de un modelo de la base de datos del nodo local | Completo | Ninguna |
| El acceso a los datos locales debe ser consumido y manipulado directamente desde la herramienta QGIS utilizando una vista geográfica | Completo |  |
| La tabla nodos registra el nombre y una frase de acceso para identificar el nodo, el usuario basesII solo tienen acceso a realizar inserciones en la tabla y de consultar los nombres de los nodos. | Completo | Ninguna |
| Requerimientos funcionales del sistema (vistas y procedimientos). | Completo | Ninguna |
| Se deben realizar un módulo de consulta que permita establecer filtros sobre los datos. | Completo | Ninguna |

Con relación en la “Tabla#1” se destaca punto por punto los requerimientos del proyecto. En el siguiente apartado, se mostrará los resultados obtenidos de estos puntos para así, justificar el estado de estos requerimientos.

### Análisis de resultados del primer requerimiento:

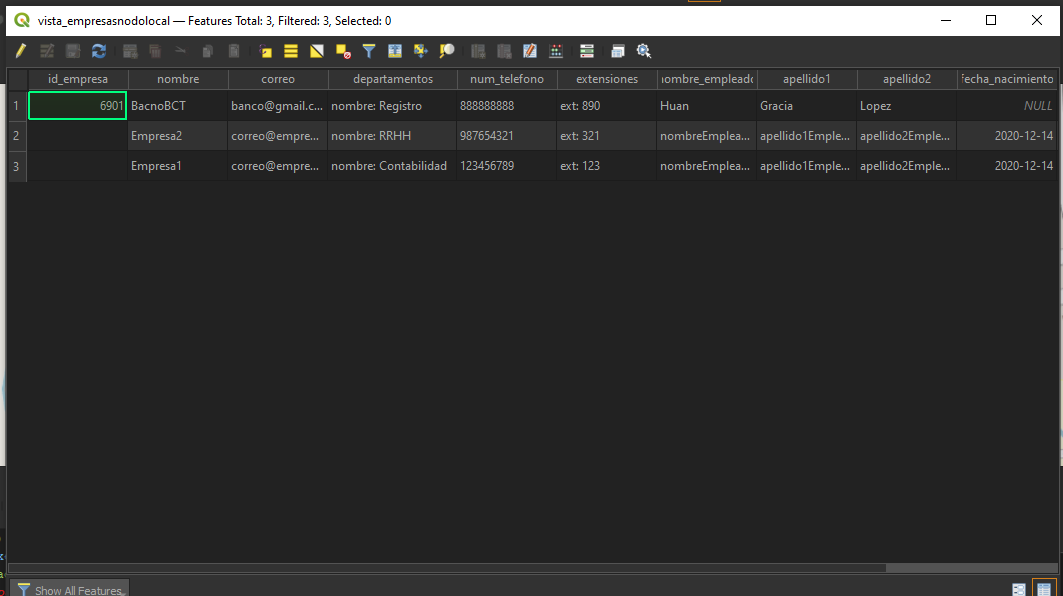
De este requerimiento se obtuvieron resultados alentadores, ya que se creó la base de datos nodo local.



Imagen#3: Base de datos local Fuente: Elaboración propia.

### Análisis de resultados del segundo requerimiento:

Como parte del consumo y acceso a los datos de la base de datos, se hizo la implementación para que funcione esta desde el sistema de información gráfica Qgis, como lo que se lee en los requerimientos del sistema suministrado en la explicación de este mismo para su respectiva implementación.



Imagen#4: Ejemplo de cómo modificar un vehículo. Fuente: Elaboración propia.

### Análisis de resultados del tercer requerimiento:

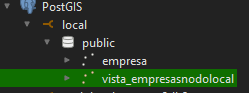
Se registra el nombre y una frase de acceso para su visualización en el nodo central cuando se hace una inserción en el nodo central.



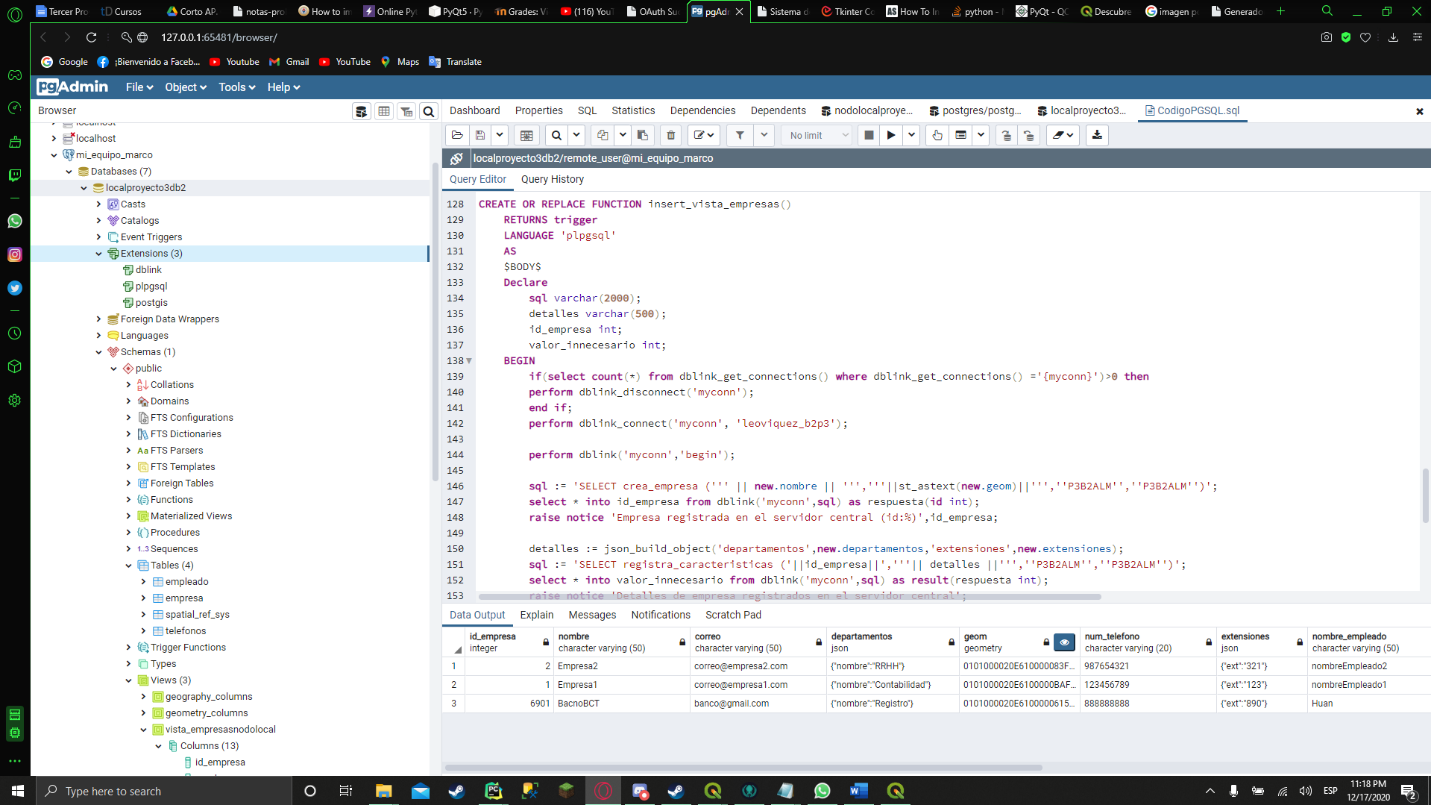
Imagen#5: Ejemplo de funcionamiento del trigger. Fuente: Elaboración propia.

### Análisis de resultados del cuarto requerimiento:

Se crean las funciones respectivas a los requerimientos funcionales de una forma correcta.



Imagen#6: Vista empresa\_local. Fuente: Elaboración propia.



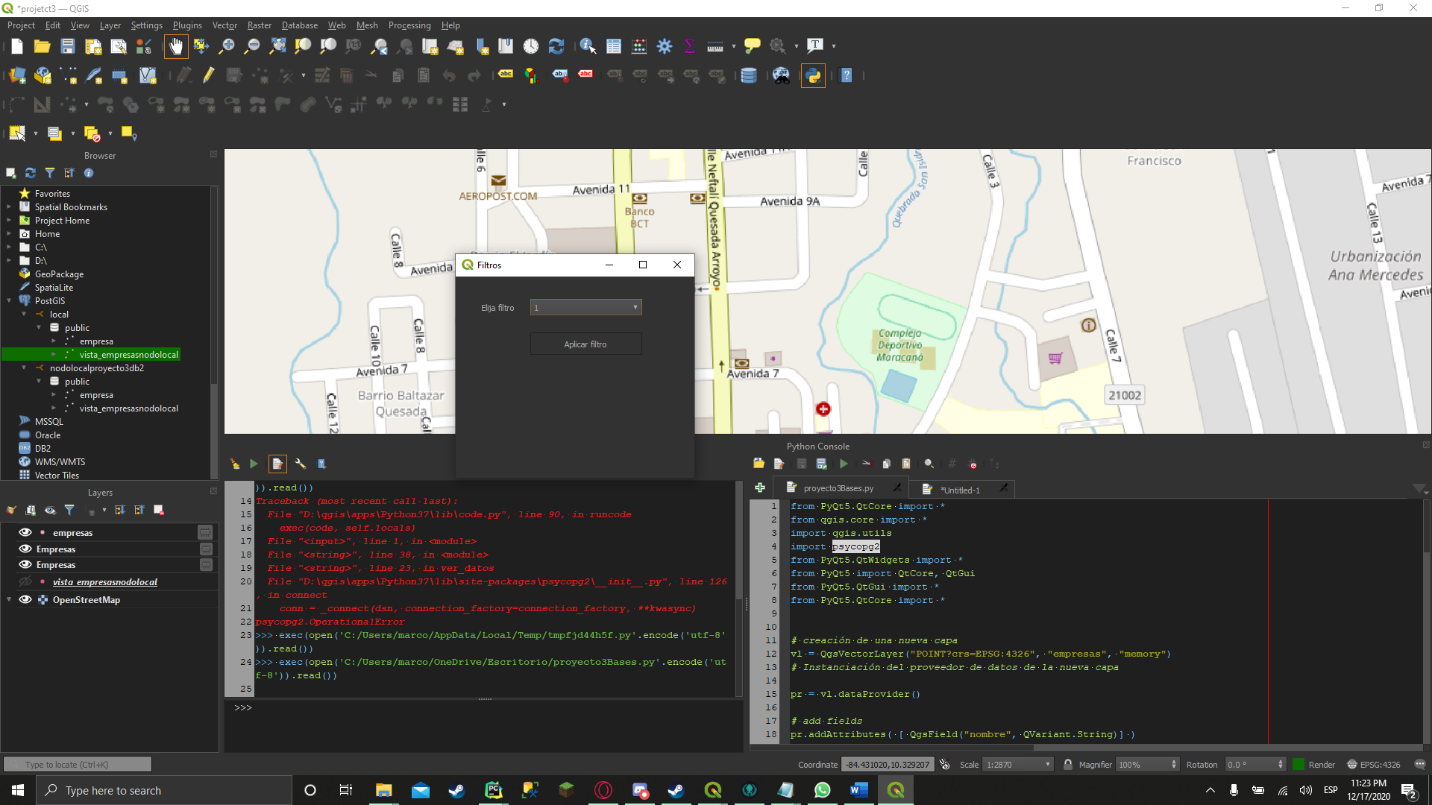
Imagen#6: Procedimiento para hacer inserciones. Fuente: Elaboración propia.

### Análisis de resultados del quinto requerimiento:

Se realizan los filtros respectivos para la consulta de datos desde Python, ya que Qgis lo permite.

Se realizan tres tipos distintos de filtros de información, los tres se eligen desde una sencilla interfaz gráfica de usuario.

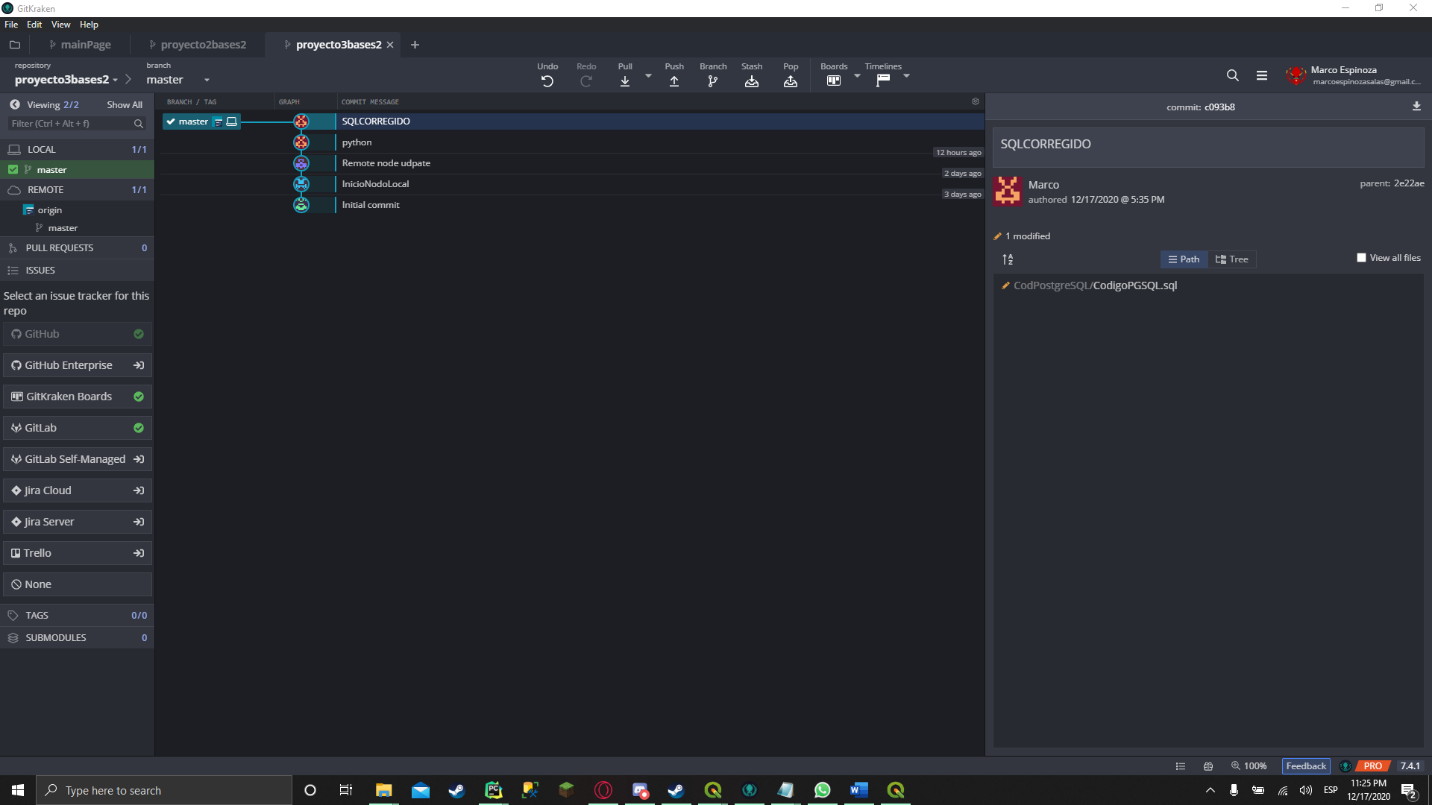
\*Cabe resaltar que, para visualizar mejor los resultados se debe reiniciar el ejecutable de Python para ver mejor los filtros



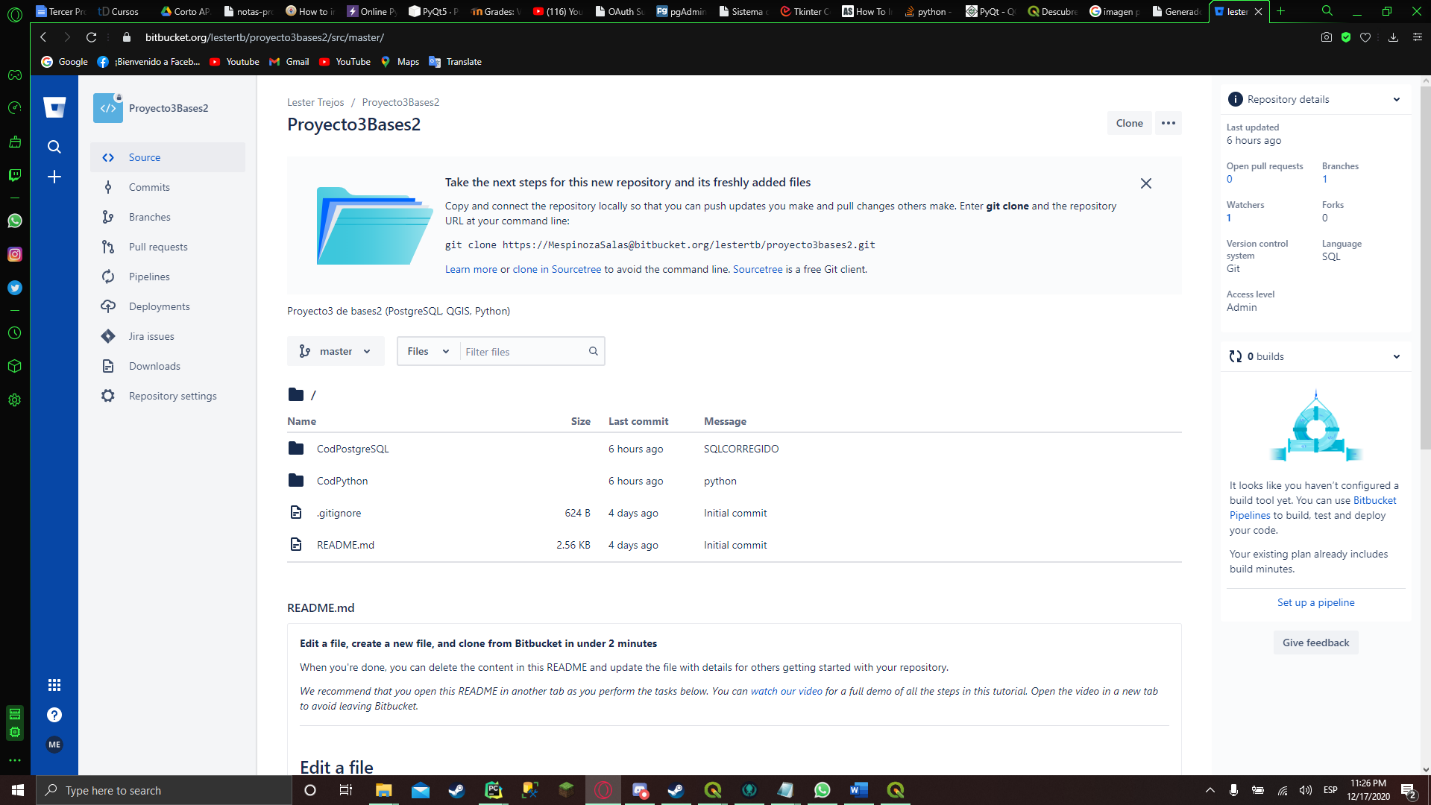
Imagen#7: Filtro desde Qgis en Python. Fuente: Elaboración propia.

Todo desarrollo de un sistema informático necesita un control de versiones refiriéndose a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración de este. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Dicho control de versiones puede hacerse de manera manual o con alguna herramienta que brinde esta función, en este proyecto se utilizaron 2 herramientas, GitKraken para la creación de repositorios locales con sus respectivos commits, permitiendo el control de versiones, además esta herramienta permite el enviar dichos repositorios a la nube, la otra herramienta es Bitbucket es la herramienta web que posee los repositorios en la nube y recibe la información enviada por GitKraken.



Imagen#8: Controlador de versiones. Fuente: Elaboración propia.



Imagen#9: Repositorio de versiones Bitbucket[[4]](#footnote-4). Fuente: Elaboración propia.

## Conclusiones

* Con base en los resultados obtenidos por medio de la solución del problema, se concluye que la creación de una base de datos distribuidas facilita el acceso de información pertinente a las personas que no requieren de toda la información completa.
* Con base en las investigaciones realizadas para el desarrollo de este proyecto, se concluye que PostgreSQL es una herramienta óptima de cara a la realización de orientadas a base de datos, ya que, cuenta con mucho soporte, y también, la información que existe en la red es muy basta para solucionar cualquier inconveniente que se presentó.
* Con base en las investigaciones realizadas para el desarrollo de este proyecto, se concluye que Qgis es una herramienta óptima de cara a las conexiones con las bases de datos ya que, cuenta con mucho soporte.

## Recomendaciones

* Dividir el trabajo en un proyecto informático es una buena práctica para su ejecución e implementación.
* Coordinar adecuadamente con los integrantes del equipo de trabajo para la adecuada distribución de tareas, estableciendo medios de retroalimentación por parte de los integrantes, para la corrección inmediata de posibles anomalías en el desarrollo del proyecto.
* Gestione su proyecto con un controlador de versiones (GitKraken[[5]](#footnote-5)), ya que compartir el código es más fácil, mejora la corrección de errores y maneja las etapas de este.

## Referencias Bibliográficas

[1]"Descubre QGIS", Qgis.org, 2020. [Online]. Available: https://qgis.org/es/site/about/index.html. [Accessed: 18- Dec- 2020].

[2]"Free Git GUI Client - Windows, Mac, Linux | GitKraken", *GitKraken.com*, 2018. [Online]. Available: https://www.gitkraken.com/. [Accessed: 18- Dec- 2020].

[3]"Github y Bitbucket servicios de Git en la nube", Hipertextual, 2014. [Online]. Available: https://hipertextual.com/archivo/2014/05/github-y-bitbucket/. [Accessed: 18- Dec- 2020].

[4]M. Laca, "PyQt 5 + Designer – Lo instalamos y examinamos código – tutorial", pythones.net. [Online]. Available: https://pythones.net/pyqt-instalacion-y-codigo-tutorial/. [Accessed: 18- Dec- 2020].

[5]"Conectarse a bases de datos PostgreSQL desde Python con psycopg2", Linuxito, 2018. [Online]. Available: https://www.linuxito.com/programacion/1059-conectarse-a-bases-de-datos-postgresql-desde-python-con-psycopg2#:~:text=Psycopg2%20es%20un%20adaptador%20PostgreSQL,librería%20oficial%20del%20cliente%20PostgreSQL. [Accessed: 18- Dec- 2020].

1. QGIS es un Sistema de Información Geográfica (SIG) de Código Abierto licenciado bajo GNU - General Public License . QGIS es un proyecto oficial de Open Source Geospatial Foundation (OSGeo). Corre sobre Linux, Unix, Mac OSX, Windows y Android y soporta numerosos formatos y funcionalidades de datos vector, datos ráster y bases de datos.[1] [↑](#footnote-ref-1)
2. **Pyqt** como dijimos es un binding de la biblioteca gráfica de QT para el lenguaje de programación Python. Nos permitirá desarrollar aplicaciones con un entorno gráfico agradable[4] [↑](#footnote-ref-2)
3. Es un adaptador PostgreSQL para el lenguaje Python implementado utilizando libpq, la librería oficial del cliente PostgreSQL.[5] [↑](#footnote-ref-3)
4. **Es un servicio que te permiten administrar tus proyectos usando Git pero en la nube** [↑](#footnote-ref-4)
5. Permite llevar el completo seguimiento de nuestros repositorios, ver ramas, tags, crear nuevos, todo el historial de nuestro trabajo, commits. [2] [↑](#footnote-ref-5)