



DOI: <https://doi.org/10.23857/dc.v11i2.4448>

Ciencias de la Educación  
Artículo de Investigación

*Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3*

*Programming Alternatives for Integrating Asterisk and PostgreSQL over IPv6 for Consulting Academic Notes in GNS3*

*Alternativas de programação para integração de Asterisk e PostgreSQL sobre IPv6 para consulta de notas académicas no GNS3*

Cristian Salcan<sup>I</sup>

[cristian.salcan@unach.du.ec](mailto:cristian.salcan@unach.du.ec)

<https://orcid.org/0009-0005-3154-374X>

Gonzalo Allauca<sup>II</sup>

[gallauca@unach.edu.ec](mailto:gallauca@unach.edu.ec)

<https://orcid.org/0000-0002-7342-8142>

Lourdes Paredes<sup>III</sup>

[lparedes@epoch.edu.ec](mailto:lparedes@epoch.edu.ec)

<https://orcid.org/0000-0002-5331-2759>

**Correspondencia:** [cristian.salcan@unach.du.ec](mailto:cristian.salcan@unach.du.ec)

\***Recibido:** 23 de abril de 2025 \***Aceptado:** 14 de mayo de 2025 \* **Publicado:** 30 de junio de 2025

- I. Universidad Nacional de Chimborazo, Ecuador.
- II. Universidad Nacional de Chimborazo, Ecuador.
- III. Escuela Superior Politécnica de Chimborazo, Ecuador.

---

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

## Resumen

La presente investigación permitió evaluar el rendimiento de tecnologías utilizadas para la integración de Asterisk y PostgreSQL sobre el protocolo IPv6 , con el objetivo de implementar un prototipo funcional de consulta automática de notas académicas mediante telefonía IP sobre el simulador GNS3 con dos escenarios de integración para la evaluación del rendimiento, el primero a través de la programación de la conexión directa de Asterisk a la base de datos PostgreSQL utilizando ODBC y el segundo a través de la programación de una capa de Webservice con el módulo AGI de Asterisk; se utiliza como usuario final de VoIP el softphone Zoiper para las pruebas funcionales de consulta de notas automáticas a través de la implementación de un IVR en Asterisk. El tipo de investigación del estudio es cuantitativa y los indicadores definidos fueron el tiempo de respuesta, la latencia y el uso de recursos para medir el rendimiento en los dos escenarios de integración; y se utiliza SIPp como herramientas de esfuerzo. Como resultado de comparar las dos opciones de integración para la consulta de notas académicas, se encontró que la conexión directa a la base de datos a través de ODBC ofrece un mejor rendimiento en términos de tiempo de respuesta y latencia, con diferencias de 9.15 ms y 1.8 ms respectivamente. Además, se observó una ligera diferencia en el uso de recursos del sistema, con un 1.74% más de uso de CPU y un mínimo aumento del 0.04% en el uso de memoria RAM en comparación con la integración a través de WebService AGI, permitiendo aplicar esta solución de integración a distintos ámbitos educativos o en su defecto adaptar a otros requerimientos de concurrencia en áreas distintas a la educativa que requieran sistemas de respuesta automatizada con telefonía IP.

**Palabras Claves:** Asterisk; PostgreSQL; IPv6; GNS3; ODBC; AGI; SIPp; IVR; Telefonía IP.

## Abstract

This research allowed to evaluate the performance of technologies used for the integration of Asterisk and PostgreSQL over the IPv6 protocol, with the objective of implementing a functional prototype

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

for automatic academic grade query via IP telephony on the GNS3 simulator with two integration scenarios for performance evaluation, the first through the programming of the direct connection from Asterisk to the PostgreSQL database using ODBC and the second through the programming of a Webservice layer with the Asterisk AGI module; the Zoiper softphone is used as the VoIP end user for functional testing of automatic grade query through the implementation of an IVR in Asterisk. The type of research of the study is quantitative and the defined indicators were response time, latency and resource usage to measure performance in the two integration scenarios; and SIPp is used as effort tools. As a result of comparing the two integration options for academic grade queries, it was found that the direct connection to the database via ODBC offers better performance in terms of response time and latency, with differences of 9.15 ms and 1.8 ms respectively. In addition, a slight difference in system resource usage was observed, with 1.74% more CPU usage and a minimal 0.04% increase in RAM usage compared to integration via AGI Webservice. This allows this integration solution to be applied to different educational settings or, failing that, to adapt to other concurrency requirements in areas other than education that require automated response systems with IP telephony.

**Keywords:** Asterisk; PostgreSQL; IPv6; GNS3; ODBC; AGI; SIPp; IVR; Telefonía IP.

## Resumo

Esta pesquisa permitiu avaliar o desempenho das tecnologias utilizadas para a integração do Asterisk e PostgreSQL sobre o protocolo IPv6, com o objetivo de implementar um protótipo funcional para consulta automática de notas acadêmicas via telefonia IP no simulador GNS3 com dois cenários de integração para avaliação de desempenho, o primeiro através da programação da ligação direta do Asterisk à base de dados PostgreSQL utilizando ODBC e o segundo através da programação de uma camada Webservice com o módulo Asterisk AGI; o softphone Zoiper é utilizado como utilizador final VoIP para testes funcionais de consulta automática de notas através da implementação de um IVR no Asterisk. O tipo de investigação do estudo é quantitativo e os indicadores definidos foram o tempo de resposta, a latência e a utilização de recursos para medir o desempenho nos dois cenários de integração; e o SIPp é utilizado como ferramentas de esforço. Como resultado da comparação das duas opções de integração para as consultas de notas acadêmicas, verificou-se que a ligação direta à

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

base de datos via ODBC ofrece un mejor desempeño en términos de tiempo de respuesta e latencia, con diferencias de 9,15 ms e 1,8 ms respectivamente. Además disso, observó-se una ligera diferencia na utilização dos recursos do sistema, com mais 1,74% de utilização da CPU e um aumento mínimo de 0,04% na utilização da RAM em comparação com a integração através do AGI Webservice. Isto permite que esta solução de integração seja aplicada a diferentes ambientes educativos ou, na sua ausência, se adapte a outros requisitos de simultaneidade em áreas para além da educação que exigem sistemas de resposta automatizados com telefonia IP.

**Palavras-chave:** Asterisk; PostgreSQL; IPv6; GNS3; ODBC; AGI; SIPp; IVR; Telefonia IP.

### Introducción

Las soluciones telemáticas eficientes y escalables para la gestión de servicios han ido evolucionando exponencialmente, es así que la transición del protocolo IPv4 a IPv6; como consecuencia del agotamiento de direcciones IP, ha permitido que tecnologías como la telefonía IP y los servicios de bases de datos distribuidas se integren de forma más eficaz (Zhang, Li, & Zhang, 2020; Khasawneh & Alarmouty, 2019). En este contexto, la presente investigación propone a través de un prototipo programar la integración de la PBX Asterisk, con el gestor de base de datos PostgreSQL; utilizando el protocolo IPv6 (Singh, 2023). El prototipo funcional se desarrolla en el simulador GNS3 que permite desplegar escenarios cliente-servidor en un entorno controlado (González, 2022). El prototipo incorpora un IVR (Respuesta de Voz Interactiva) para consultas académicas mediante llamadas telefónicas, lo cual representa una solución útil para instituciones educativas. Esta propuesta considera dos tipos de integración entre las plataformas: una conexión directa mediante ODBC y otra a través de la programación de WebServices usando el módulo AGI de Asterisk (Momtaz & Noor, 2015). La evaluación del prototipo se enfoca en medir indicadores de rendimiento como la latencia, el tiempo de respuesta y el uso de recursos del sistema. La herramienta SIPp permite ejecutar pruebas de rendimiento simulando múltiples llamadas y genera estadísticas confiables (Singh, 2023; Madsen, 2018). La integración entre plataformas se la realiza con el objetivo de permitir la consulta a una base de datos, utilizando dos escenarios de integración. Se utilizan dos tablas en la base de datos con información académica para la ejecución de las pruebas y el análisis de resultados, proyectando así el uso de IPv6 en entornos académicos o en cualquier otro entorno. El primer escenario de integración

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

de las plataformas de Voz y Base de Datos se realiza con la programación de una conexión directa desde Asterisk hacia la base de datos PostgreSQL, mientras que en el segundo escenario se programa el uso de Webservice con el módulo AGI de Asterisk. La selección del simulador de red GNS3 como plataforma para el desarrollo del prototipo permite una simulación controlada y precisa, previo su implementación en entornos de producción reales (Bustos, 2023). Esta investigación permite validar la funcionalidad de la integración propuesta, y proporcionar una comprensión de las alternativas de integración de Asterisk y PostgreSQL sobre IPv6 y su rendimiento. Los resultados obtenidos en la presente investigación no solo permitieron comparar ambos escenarios, sino también determinar la factibilidad de aplicar este tipo de soluciones en entornos reales. La incorporación de IPv6 no solo garantiza sostenibilidad a largo plazo, sino características de escalabilidad y adaptabilidad a otras aplicaciones tecnológicas (NIC México, 2024). La revisión de casos de estudio permitió establecer una perspectiva práctica de cómo otras organizaciones han abordado la integración de tecnologías utilizando el protocolo IPv6. El primer caso de estudio: La Universidad de California, Berkeley, implementó Asterisk sobre IPv6 en 2016 para mejorar la conectividad y la seguridad de sus sistemas de telefonía IP. La implementación se realizó en dos fases: la primera consistió en la migración de los servidores Asterisk existentes a IPv6, y la segunda en la implementación de nuevas funciones de seguridad IPv6. Los beneficios de la implementación incluyeron mejor rendimiento, reducción de latencia y tiempo de respuesta, mayor seguridad al proporcionar una mayor protección contra ataques, y simplificación de la administración al eliminar la necesidad de NAT (UC Berkeley, 2016). El segundo caso de estudio refiere al Monitoreo Vehicular, el mismo que planteó el diseño e implementación de una aplicación que pueda acceder a una base de datos que posee las calles de la ciudad de Quito, vinculadas a las coordenadas de monitoreo GPS, del cual se pueden recibir y generar llamadas para otorgar información de la posición actual del vehículo con una base de alarmas. Dentro de este trabajo se analizaron los códecs de telefonía IP que relacionan a Asterisk, así como las tecnologías de TTS. Se implementó la comunicación hacia la base de datos María DB mediante el uso de AGI PHP. En esto se consiguió ejecutar con éxito el plan de marcado diseñado con los respectivos scripts de PHP para la lectura de la base de datos y así se obtuvo el sistema final con un costo de implementación representativo, sin embargo, se analizó que es representativo tomando en cuenta los valores que poseen los softwares especializados (Lema & Ruiz, 2014). El tercer caso de

---

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

estudio: en una empresa denominada Innova Networks, a través de un IVR los usuarios pueden ejecutar consultas por medio de scripts programados en PHP, que conjuntamente con AGI de Asterisk permite ejecutar instrucciones dentro del dialplan para consulta de información de productos, mostrando la información solicitada por el usuario, las alertas de desabastecimiento oportunamente y permitiendo a los encargados la toma de decisiones adecuada para mejorar el control de los productos (Rojas, 2013).

### 3.- Objetivos

#### General

Programar Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

#### Objetivos Específicos

- Investigar las tecnologías que permitan programar la integración de Asterisk y PostgreSQL a través del IPv6.
- Desarrollar el prototipo para la integración de Asterisk y PostgreSQL con IPv6 sobre GNS3.
- Evaluar el rendimiento de las tecnologías utilizadas para la integración de Asterisk y PostgreSQL sobre IPv6.

### 4.- Metodología

La metodología aplicada en la presente investigación fue de tipo experimental, con un enfoque cuantitativo orientado a evaluar el rendimiento y la funcionalidad del prototipo en términos de tiempo de respuesta, la latencia y el uso de recursos al integrar Asterisk y PostgreSQL sobre el protocolo IPV6 utilizando la herramienta de esfuerzo para telefonía IP SIPp y desplegando sobre GNS3 dos escenarios funcionales. A continuación, se describen detalladamente las fases del proceso metodológico:

#### 4.1. Diseño del prototipo y definición del entorno de simulación

Se utilizó el simulador de red GNS3 como plataforma de simulación para modelar una red local con direccionamiento IPv6. Se definieron tres nodos principales: un servidor Asterisk, un servidor PostgreSQL y un cliente softphone. Las máquinas virtuales se configuraron con sistemas operativos

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

compatibles con IPv6 y se establecieron túneles de comunicación para la transmisión de datos y voz sobre la red simulada, como se muestra en la arquitectura de la figura 1

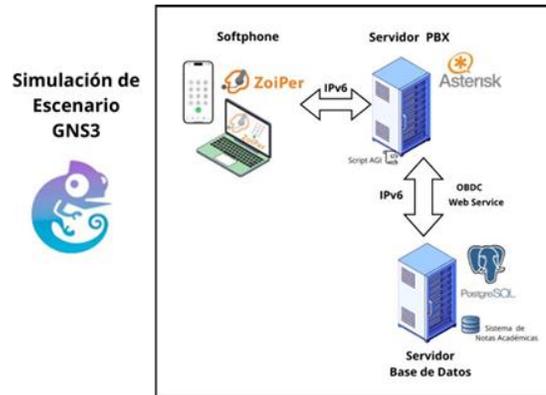


Figura 1. Arquitectura funcional del prototipo utilizando GNS3

### 4.2. Desarrollo de la arquitectura del sistema

Para la evaluación del rendimiento de integración de tecnologías ODBC y Webservices, se implementa una arquitectura de red que permita desplegar los dos siguientes escenarios:

- **Escenario I:** Integración directa mediante el uso de ODBC, que permite al servidor Asterisk realizar consultas SQL directamente sobre la base de datos, como se muestra en la figura 2.

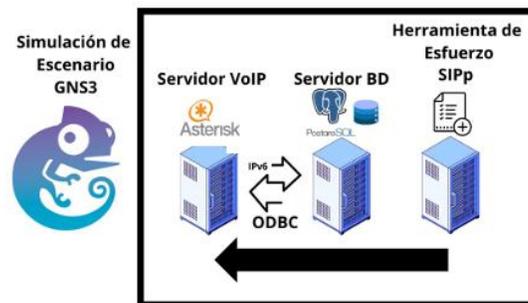


Figura 2. Escenario I de Integración utilizando ODBC

- **Escenario II:** Integración mediante la programación de WebService y scripts en PHP conectados a Asterisk mediante el módulo AGI (Asterisk Gateway Interface). Este enfoque emula una arquitectura más desacoplada, segura e interoperable con distintas plataformas y tecnologías, como se muestra en la figura 3.

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

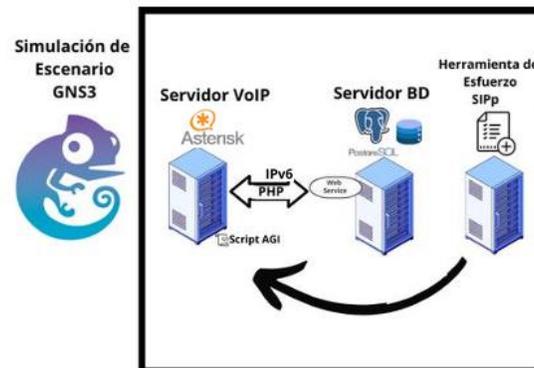


Figura 3. Escenario II de integración utilizando Webservice y AGI

### 4.3. Configuración de componentes de red y servicios

En ambos escenarios se utilizaron los siguientes componentes: Sistema operativo Linux Ubuntu Server 20.04 implementado sobre una máquina virtual creada en VirtualBox, PostgreSQL 10.23 como sistema gestor de bases de datos con tablas previamente cargadas con registros simulados de notas de estudiantes en distintas asignaturas, Asterisk versión 13.38 con soporte para IPv6, Softphone Zoiper instalados en clientes para realizar las llamadas y comprobar la funcionalidad de las consultas automáticas receptadas por el IVR implementado en Asterisk. Las direcciones IPv6 se asignaron estáticamente tanto en las interfaces de los servidores como de las interfaces de los routers del escenario desplegado en GNS3 para garantizar una comunicación estable durante las pruebas. El plan de marcado de Asterisk fue personalizado para que cada escenario pueda operar de forma independiente, se utiliza además el lenguaje de programación PHP para programar la conexión directa a PostgreSQL así como la capa de Webservice que accede las notas académicas registradas.

### 4.4. Pruebas de funcionamiento

Se utilizó el comando ping6 para comprobar conectividad entre los distintos servidores desplegados a través de los dos escenarios propuestos en GNS3. Se verificó el funcionamiento de las llamadas desde los softphones y la respuesta automática a través del IVR y obtener los datos de las notas académicas desde la base de datos PostgreSQL, utilizando la programación de conexión tanto a través del ODBC como del Webservice.

### 4.5. Medición del rendimiento.

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

Las pruebas de rendimiento se realizaron sobre las alternativas programadas de acceso directo a PostgreSQL y el acceso indirecto a través de Webservice al gestor de base de datos, estas pruebas se ejecutaron mediante la herramienta de esfuerzo SIPp la misma que permite simular llamadas y generar tráfico SIP, entregando como resultado el rendimiento (load testing) de Asterisk como respuesta a las llamadas ejecutadas; estas llamadas pueden ser parametrizadas dependiendo los indicadores que se desea medir. SIPp utiliza primitivas o comandos SIP como INVITE, TRYING, RINGING, OK, ACK, BYE, entre otras, como se muestra en la figura 4.

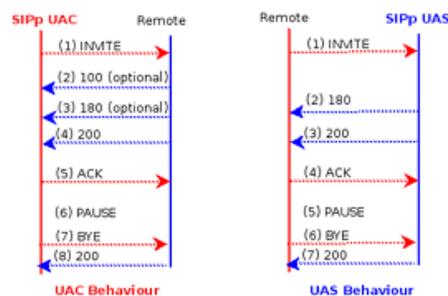


Figura 4. Primitivas de Comunicación de SIPp

Las pruebas de rendimiento se realizaron simulando 1000 llamadas telefónicas en cada uno de los escenarios y se evaluaron los siguientes indicadores:

- Latencia (tiempo de ida y vuelta de la petición y respuesta).
- Tiempo de respuesta total desde el inicio de la llamada hasta la respuesta verbal del sistema.
- Porcentaje de uso de CPU y memoria RAM durante la carga.

Se realizaron tres iteraciones de cada prueba para validar la consistencia de los resultados. Además, se empleó la herramienta htop para el monitoreo del uso de recursos en tiempo real.

### 4.6 Procedimiento de implementación

4.6.1 Investigar las tecnologías de integración de Asterisk y PostgreSQL sobre protocolo IPv6 y realizar un análisis comparativo para seleccionar las mejores alternativas de integración a implementarse sobre un escenario simulado utilizando GNS3.

4.6.2 Implementación del Prototipo de Integración de Asterisk y PostgreSQL sobre IPv6 utilizando GNS3.

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

#### 4.6.2.1 Instalación y Configuración de GNS3 sobre el Sistema Operativo Windows

4.6.2.2 Creación de 4 Máquinas Virtuales utilizando VirtualBox versión 7.0, para la instalación de:

- Windows 7 64 bits. - Donde se instala el Softphone Zoiper para realizar las llamadas y consultar las notas del estudiante.
- Linux Sangoma 7.0. – Sistema operativo donde se encuentra preinstalado la PBX Asterisk versión 13.38.3 donde se configura los canales SIP, plan de marcado, IVR y se configura las alternativas de integración con PostgreSQL utilizando ODBC y Webservice con AGI PHP.
- Ubuntu Desktop versión 18.04.- Donde se instala la base de datos PostgreSQL con soporte del protocolo IPv6 y se implementa además la capa de WebService a ser consumida desde AGI de Asterisk.
- Ubuntu Desktop versión 22.04.- Donde se instala la herramienta de esfuerzo SIPp para realizar las pruebas de rendimiento del sistema.

#### 4.6.2.3 Diseño de los escenarios de red para las alternativas de integración y pruebas de rendimiento

- Establecimiento del direccionamiento de red
- Diseño Lógico y despliegue físico de máquinas virtuales y componentes activos de red para cada escenario de integración.

#### 4.6.2.4 Configuración de Asterisk para la integración con ODBC y Webservice.

- Configuración de canales SIP y plan de marcado en la PBX Asterisk
- Configuración de `/etc/asterisk/manager.conf` para la detección de Softphone por IPv6 en la línea `bindaddr`.
- Configuración de la extensión 1888 para registrar el Softphone y probar las dos alternativas de integración con el sistema académico.
- Comprobación de conexión del Softphone con el servidor de Asterisk.
- Configuración del IVR a través del número 2001 para conexión directa a mediante el uso de ODBC con la base de datos de PostgreSQL.

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

- Configuración del IVR a través del número 2002 para conexión mediante Webservice a la base de datos de PostgreSQL. En la figura 6 se muestra la configuración para establecer comunicación entre los servidores de Asterisk y PostgreSQL utilizando un socket ODBC. Para ello, se instaló el driver ejecutando el comando "sudo yum install odbc-postgresql". En el archivo /etc/odbc.ini se configuró la conexión con la base de datos como: usuario, contraseña, puerto, dirección IPv6 y nombre de la base de datos.

```
[SISTEMA_ACADEMICO_PSQL]
Description      = PostgreSQL connection to Alumnos
Driver           = PostgreSQL
Database        = Sistema_Academico
Servername       = 2003:db8:c::10
Username         = admin_
Password        = 1234_
Port            = 5432
```

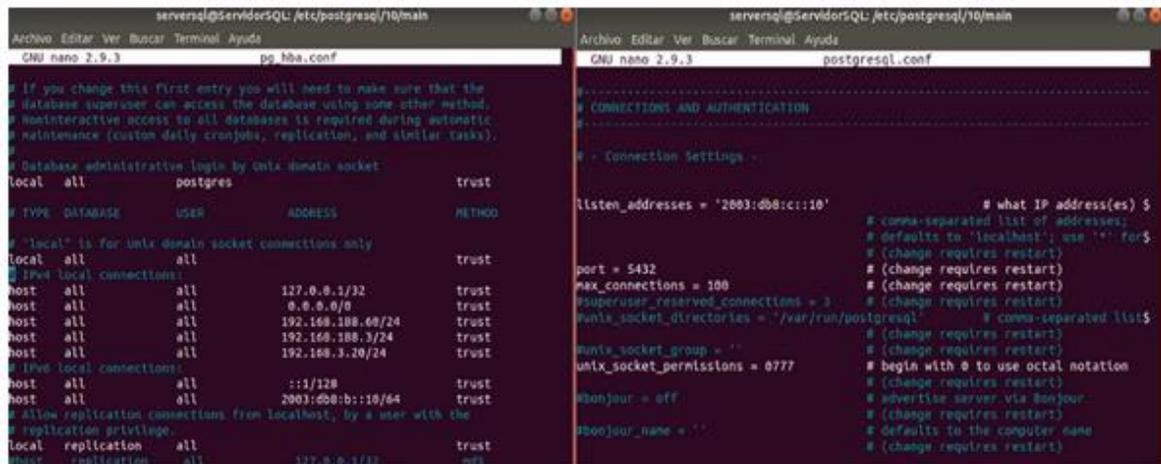
Figura 5. Conexión de PostgreSQL con Asterisk

### 4.6.2.5 Implementación de Base de Datos mediante PostgreSQL

- Instalación de PostgreSQL 10.23 dentro de la máquina virtual de Ubuntu Desktop versión 18.04.
- Configuración de archivo para escuchar las direcciones IPv6 configuradas dentro del archivo /etc/postgresql/10/main/ pg.hba.conf.
- Configuración de archivo /etc/postgresql/10/main/ postgresql.conf. para la selección del puerto y permisos necesarios para la comunicación.
- Creación de la Base de Datos Sistema\_Academico dentro de PostgreSQL.
- Creación de la tabla Notas\_Estudiantes dentro de base de datos Sistema\_Academico con las columnas ID, nombre, apellido, CI, Nota1, Nota2, Nota3, Nota4, Nota5.
- Creación de la tabla Materias dentro de la tabla Notas\_Estudiantes con las columnas ID, código, materia, asignatura.
- Registro de las asignaturas dentro de la tabla Materias.
- Registro de los alumnos y notas dentro de la tabla Notas\_Estudiantes.

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

- En la ifgura 6 se observa al archivo pg.hba.conf, configurado el método y la dirección IPv6 del servidor de Asterisk para que exista interacción entre estos dos servidores. Mientras que en el archivo postgresql.conf se configuró el puerto, permisos para el socket de dominio Unix y la dirección IPV6 del servidor de PostgreSQL.



```

serversql@ServidorSQL:/etc/postgresql/10/main
GNU nano 2.9.3 pg_hba.conf
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local all postgres trust

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
host all all 0.0.0.0/0 trust
host all all 192.168.188.60/24 trust
host all all 192.168.188.3/24 trust
host all all 192.168.3.20/24 trust
# IPv6 local connections:
host all all ::1/128 trust
host all all 2003::db8:b::10/64 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all trust
host replication all 127.0.0.1/32 md5

serversql@ServidorSQL:/etc/postgresql/10/main
GNU nano 2.9.3 postgresql.conf
# CONNECTIONS AND AUTHENTICATION
#-----
# Connection Settings
#-----
listen_addresses = '2003::db8:c::10' # what IP address(es) $
# comma-separated list of addresses;
# defaults to 'localhost'; use '*' for $
# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list$
# (change requires restart)
unix_socket_group = '' # (change requires restart)
unix_socket_permissions = 0777 # begin with 0 to use octal notation
# (change requires restart)
# advertise server via Bonjour
# (change requires restart)
# defaults to the computer name
# (change requires restart)

```

Figura 6. Configuraciones de PostgreSQL para integración con Asterisk.

### 4.6.2.5 Implementación de servidor festival TTS

- Instalación de archivos del sistema Festival mediante el comando `sudo yum install festival`.
- Configuración de Puerto e IP dentro del archivo `/etc/Asterisk/festival.conf`, para la ejecución dentro de Asterisk

### 4.6.2.6 Configuración de integración con ODBC

- Instalación del softphone zoiper dentro de la máquina virtual de Windows 7 64bits.
- Colocar las SIP CREDENTIALS con la dirección IP del servidor, el usuario o extensión y el password.

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

- Registrar la extensión y verificar conexión.
- Integración del sistema mediante ODBC (Escenario 1)
- Instalación del socket de conexión sudo yum install odbc-postgresql.
- Comprobación del Driver ODBC dentro del archivo /etc/odbcinst.ini
- Crear la instancia SISTEMA\_ACADEMICO\_PSQL para el llamado de las funciones ODBC dentro del archivo /etc/odbc.ini.
- Configurar la instancia con usuario: admin, contraseña: 1234, puerto de comunicación: 5432, Dirección IP del servidor: 2003:db8: c::10 y nombre de la base de datos: Sistema\_Academico.
- Comprobar la Conexión con la Base de Datos mediante el comando isql -v Sistema\_Academico\_PSQL.
- Configurar el archivo res\_odbc\_custom.conf para habilitar la instancia SISTEMA\_ACADEMICO\_PSQL.
- Configurar el archivo func\_odbc\_conf con los QUERY necesarios para la búsqueda dentro de PostgreSQL.
- Configurar el archivo extensions\_custom.conf, para determinar el Dialplan de la extensión 2001 para la implementación del IVR de consulta de notas.
- En la Figura 7 se muestra la configuración del Dialplan en el archivo extensions\_custom.conf, que define el flujo y las acciones a tomar durante una llamada. Se ha creado dos planes de llamada [consulta\_bd]

```

GNU nano 2.3.1 File: /etc/asterisk/extensions_custom.conf

[ivr_academico]
include => consulta_bd
include => consulta_webservice

[consulta_bd]
exten=> 2001,1,Answer()
exten=> 2001,2,Playback(Bienvenida)
exten=> 2001,3,Playback(Cedula)
exten=> 2001,4,Read(CI)
exten=> 2001,5,Set(nombre=${ODBC_nombre(${CI}})
exten=> 2001,6,GotoIf($[${nombre}=""?]fcal:cca1)
exten=> 2001,7(cca1),Set(apellido=${ODBC_apellido(${CI}})
exten=> 2001,8(cca1),Festival(ESTUDIANTE)
exten=> 2001,9(cca1),Festival(${nombre})
exten=> 2001,10(cca1),Festival(${apellido})
exten=> 2001,11(cca1),Playback(Opcion1)
exten=> 2001,12(cca1),Playback(Opcion2)
exten=> 2001,13(cca1),WaitExten(3)
exten=> 2001,14(cca1),Goto(2001,11)

;CEDULA INCORRECTA
exten=> 2001,n(fcal),Festival(Numero de Cedula Incorrecto)
exten=> 2001,n(fcal),Festival(Saliendo del Sistema)
exten=> 2001,n(fcal),Hangup

;SI SELECCIONA 1 ENTRA A ESTE CODIGO
exten=> 1,1,Answer()
exten=> 1,n,Festival(INGRESE EL CODIGO DE ASIGNATURA)
exten=> 1,n,Read(codigo)
exten=> 1,n,Festival(UN MOMENTO PORFAVOR)
exten=> 1,n,Set(materia=${ODBC_codigo_asignatura(${CI},${codigo}})
exten=> 1,n,GotoIf($[${materia}=""?]nmat:smat)
exten=> 1,n(smat),Festival(BUSCANDO INFORMACION)
exten=> 1,n(smat),Set(asignatura=${ODBC_asignatura(${CI},${codigo}})

```

Figura 7.- Plan de Marcado de Asterisk con ODBC

#### 4.6.2.7 Integración del sistema mediante Webservice AGI PHP

- Crear el archivo prueba.php para el consumo desde Asterisk incluyendo la librería phpagi.
- Configurar los QUERY que permitan la lectura de los datos y de los alumnos en la base de datos.
- Agregar la dirección url `http://[2003:db8:c::10]:80/pruebacon.php` para que exista conexión con la Base de Datos.
- Programar el código para que realice la búsqueda y consulta de la nota tanto por todas las materias como por asignatura.
- Configurar el archivo `extensions_custom.conf`, para determinar el Dialplan de la extensión 2002 para la implementación del IVR de consulta de notas por Webservice.
- Colocar dentro del dialplan la dirección `/var/lib/Asterisk/agi-bin/prueba.php`, para la lectura mediante AGI.
- En la figura 8 se muestra el plan de marcado de Asterisk para conectarse a través de Webservice a la base de datos del sistema académico y la consulta de notas.
- En la figura 9 se muestra la programación del archivo denominado como `prueba.php` en `var/lib/asterisk/agi-bin`. Este archivo ejecutará la función AGI y contiene el algoritmo para obtener y extraer información, así como la dirección URL del Webservice a través del cual se accederá a la base de datos.

```

[consulta_webservice]
exten => 2002,1,Answer()
exten => 2002,2,Playback(Bienvenida)
exten => 2002,3,Playback(Cedula)
exten => 2002,4,Read(CI)
exten => 2002,5,Festival(Buscando Informacion, Un momento Porfavor)
exten => 2002,6,AgI(/var/lib/asterisk/agi-bin/prueba.php,${CI})
exten => 2002,7,GotoIf("${${validacion}"="No existe"}?cal1:cal2)
exten => 2002,8(cal2),Festival('Estudiante ${nombre} ${apellido}')
exten => 2002,9(cal2),Festival('Presione tres si desea conocer la nota por codigo')
exten => 2002,10(cal2),Festival('Presione cuatros si de desea conocer todas las notas')
exten => 2002,11(cal2),WaitExten(3);
exten => 2002,12(cal2),Goto(2002,9);
;CEDULA INCORRECTA
exten => 2002,n(cal1), Festival(Numero de cedula incorrecta)
exten => 2002,n(cal1), Festival(Saliendo del Sistema)
exten => 2002,(cal1), Hangup()

```

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

Figura 8. Programación del Plan de Mercado de Asterisk con Webservice.

```
GNU nano 2.3.1 File: /var/lib/asterisk/agi-bin/prueba.php
#!/usr/bin/php -q
<?php
require_once(__DIR__.'phpagi/phpagi.php');
include('http://[2003:db8:c::101:80/pruebacon.php');
// TimeZone
date_default_timezone_set('America/Bogota');
ob_implicit_flush(true);
// Execution Timeout
set_time_limit(30);
// New AGI Object
$agi = new AGI();
$CI=$SERVER['argv'][1];
$agi->verbose($CI);
$query="SELECT * FROM NOTAS1 WHERE ci='$CI'";
$result=pg_query($query);
$row=pg_fetch_array($result);
if ($row!=NULL)
{
    $nombre=$row[1];
    $apellido=$row[2];
    $nota1=$row[4];
    $nota2=$row[5];
}
else
{
    $validacion="No existe";
}
$agi->set_variable("validacion",$validacion);
$agi->set_variable("nombre",$nombre);
$agi->set_variable("apellido",$apellido);
$agi->set_variable("nota1",$nota1);
$agi->set_variable("nota2",$nota2);
```

Figura 9. Programación archivo AGI de Asterisk con Webservice.

#### 4.6.3 Pruebas de esfuerzo del prototipo del sistema.

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3
 

---

- Instalar el Software SIPp dentro de la máquina virtual Ubuntu 22.04.
- Crear el archivo XML con los parámetros para las pruebas de esfuerzo.
- Ejecutar el archivo XML con SIPp
- Verificar los archivos LOG para verificar los resultados obtenidos en cada uno de los escenarios.
- Determinar los parámetros de uso de recursos, latencia y tiempo de respuesta.

### 5.- Resultados

En la Tabla 1 se observan las diferentes tecnologías que permiten la integración de servicios entre la base de datos PostgreSQL y Asterisk con IPV6. Para el primer escenario (Ver figura 2) existen opciones de driver de conexión ODBC y JDBC, siendo el más adecuado para el prototipo la primera opción, debido a que únicamente requería la instalación del driver correspondiente a la base de datos PostgreSQL, mientras que el otro requería de una serie de componentes pertenecientes a JAVA que dificultarían la implementación por compatibilidad. Se observó que para el escenario 2 (Ver figura 3) con WebService se identificaron igualmente dos alternativas para la integración entre los servicios. El primero AGI, como parte nativa de las librerías de Asterisk y que permitía la lectura de código PHP para la lectura de los datos dentro de PSQL, siendo esta la más adecuada para el proyecto. Por otro lado, la opción no nativa FastAGI con comandos diferentes a la primera opción y requería la instalación y descarga de archivos que permitieran la vinculación de esta interfaz con Asterisk.

**Tabla 1.** Comparativa de Tecnologías de Integración

Escenario	Tecnologías de Integración	Observaciones
Conexión directa a base de datos	Driver ODBC	Requiere un Driver específico para cada base de datos
	Driver JDBC	Requiere de Servicios y Aplicaciones compatibles con JAVA
	AGI	Interfaz más utilizada para la integración con otros servicios que puede utilizar PHP, Python.

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

<b>Conexión mediante WebService</b>	FAST AGI	Requiere instalación de librerías adicionales y los comandos son diferentes con respecto a AGI.
-------------------------------------	----------	---

Como resultado de ejecutar las pruebas del escenario 1 (Ver figura 2) se marcó desde el Softphone Zoiper al número 2001, obteniendo la respuesta automática del IVR programado en el Dial Plan de Asterisk, el mismo que solicita el número de cédula del estudiante registrado en la base de datos y se devuelve la nota de la materia seleccionada, comprobando así el correcto funcionamiento del sistema mediante ODBC o conexión directa. Para el caso del escenario 2 o WebService, de la misma forma se marca al número 2002 verificando los comandos y ejecución del script externo implementado mediante PHP por medio de AGI. Los resultados obtenidos fueron organizados en tablas comparativas y representados gráficamente para facilitar su interpretación. Se utilizó estadística descriptiva para determinar el promedio, mínimo y máximo de cada métrica. Posteriormente, se realizó una comparación entre los escenarios para determinar cuál ofrecía mejor rendimiento en condiciones similares.

Para lograr estimar el valor de latencia, se analizan los paquetes recibidos y paquetes transmitidos por parte del servidor (Minda & Pacheco, 2019). Mediante el uso de los archivos de respaldo de SIPp se puede determinar el tiempo de ida y vuelta de los paquetes SIP que son devueltos del servidor de Asterisk mediante la siguiente fórmula:  $l = T_{envío} - T_{recepción}$

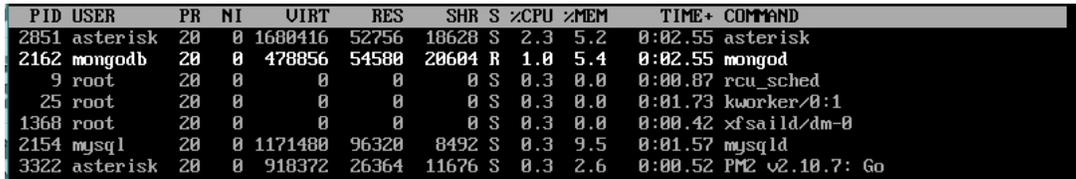
Por otro lado, para poder determinar el tiempo de respuesta utilizando SIPp se identifica el parámetro CPS que indica la cantidad de llamadas por segundo que puede ejecutar el prototipo. Con esto en mente, dentro del documento se indica que para determinar el tiempo de respuesta entre llamadas se debe utilizar el promedio de CPS y estimar el tiempo con respecto a una llamada (Voznak & Rozhon, 2010).

### 5.1 Resultados Escenario I

En la figura 10 se observa el resultado de establecer una cantidad de 1000 llamadas por prueba hasta evidenciar problemas de comunicación o llamadas que no puedan ser recibidas. Una vez establecido el protocolo de pruebas planteado se procede a la verificación del estado de recursos antes de realizar

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

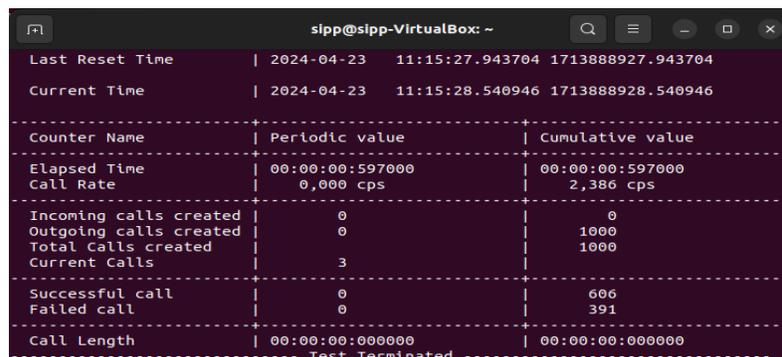
el test de estrés, mediante el uso del comando TOP. Se muestra que el servidor de Asterisk tiene un consumo del 2.3% del CPU y un 5.2% de la memoria.



PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2851	asterisk	20	0	1680416	52756	18628	S	2.3	5.2	0:02.55	asterisk
2162	mongodb	20	0	478856	54580	20604	R	1.0	5.4	0:02.55	mongod
9	root	20	0	0	0	0	S	0.3	0.0	0:00.07	rcu_sched
25	root	20	0	0	0	0	S	0.3	0.0	0:01.73	kworker/0:1
1368	root	20	0	0	0	0	S	0.3	0.0	0:00.42	xfsaidd/dm-0
2154	mysql	20	0	1171480	96320	8492	S	0.3	9.5	0:01.57	mysqld
3322	asterisk	20	0	918372	26364	11676	S	0.3	2.6	0:00.52	PM2 v2.10.7: Go

Figura 10. Uso de recursos pretest del Escenario 1

Para obtener los resultados de la prueba de estrés para 1000 llamadas se utilizó el siguiente comando dentro de la terminal del servidor de SIPp: `sipp -sf prueba4.xml -s 2001 -m 1000 -r 40 -l 80 -i [2003:db8:d::10] [2003:db8:b::10] -trace_msg`. Dentro de este comando se especifica el archivo XML asignado para todo el protocolo de pruebas, se selecciona la extensión del servidor a la que se va a realizar la llamada, la cantidad de llamadas a realizarse (-m 1000), el número de llamadas por segundo (-r 40) y las direcciones de IP tanto del servidor como del cliente. En la figura 11 se puede observar la respuesta del software SIPp con una cantidad de 1000 llamadas en dónde se muestra que se realizaron 606 llamadas efectivas, con un promedio de 2.386 llamadas por segundo. Con este valor podemos obtener el tiempo de respuesta entre cada llamada, ya que una se quiere conocer cuánto tiempo es el intervalo entre llamadas que tarda en responder el servidor, generando un valor de 41.910 ms entre cada llamada.



Counter Name	Periodic value	Cumulative value
Last Reset Time	2024-04-23 11:15:27.943704	1713888927.943704
Current Time	2024-04-23 11:15:28.540946	1713888928.540946
Elapsed Time	00:00:00:597000	00:00:00:597000
Call Rate	0,000 cps	2,386 cps
Incoming calls created	0	0
Outgoing calls created	0	1000
Total Calls created	0	1000
Current Calls	3	
Successful call	0	606
Failed call	0	391
Call Length	00:00:00:000000	00:00:00:000000

Figura 11. Respuesta software SIPp en 1000 llamadas escenario 1

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

Se observa los archivos de logs que genera la simulación, en donde se muestran los encabezados de los paquetes transmitidos y recibidos entre el simulador con el servidor de Asterisk. Esta información de tiempo que genera estos paquetes permite el cálculo de la latencia existente en el sistema. En la figura 12 muestra las pruebas de latencia que se puede obtener restando la hora del paquete recibido menos la hora del paquete enviado que para este caso es de 2.4955 ms.

```

94
95 ----- 2024-04-23 11:08:29.617354
96 UDP message sent (380 bytes):
97
98 INVITE sip:2001@[2003:db8:b::10]:5060 SIP/2.0
99 Via: SIP/2.0/UDP [2003:db8:d::10]:5060
100 From: 1888 <sip:1888@[2003:db8:d::10]:5060>;tag=4386SIPpTag004
101 To: 2001 <sip:2001@[2003:db8:b::10]:5060>
102 Call-ID: 4-4386@2003:db8:d::10
103 CSeq: 1 INVITE
104 Contact: sip:1888@[2003:db8:d::10]:5060
105 Max-Forwards: 70
106 Subject: Performance Test
107 Content-Type: application/sdp
108 Content-Length: 0
109
110
111 ----- 2024-04-23 11:08:29.620622
112 UDP message received [449] bytes :
113
114 SIP/2.0 401 Unauthorized
115 Via: SIP/2.0/UDP [2003:db8:d::10]:5060;rport=5060;received=2003:db8:d::10
116 Call-ID: 2-4386@2003:db8:d::10
117 From: "1888" <sip:1888@[2003:db8:d::10]>;tag=4386SIPpTag002
118 To: "2001" <sip:2001@[2003:db8:b::10]>
119 CSeq: 1 INVITE
120 WWW-Authenticate: Digest realm="asterisk",nonce="1713888511/
d55b7096945c21abcf185a0c841b420",opaque="6cedf38913bea3ae",algorithm=md5,qop="auth"
121 Server: FPRX-14.0.17(13.38.3)
    
```

Figura 12. Respuesta de log de SIPp para calcular la latencia del escenario 1

La figura 13 muestra finalmente que se requiere evaluar el comportamiento de los recursos utilizados por el servidor mediante el uso del comando top. que el porcentaje del uso del CPU aumentó a 7.7% y de la Memoria a un 6.2%.

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2149	mysql	20	0	1171480	102332	8492	S	7.7	10.1	0:02.88	mysqld
2884	asterisk	20	0	1680816	62620	19056	S	7.7	6.2	0:03.32	asterisk
3	root	20	0	0	0	0	S	1.0	0.0	0:01.86	ksoftirqd/0
2157	mongodb	20	0	478860	56708	20624	S	1.0	5.6	0:02.22	mongod
3190	asterisk	20	0	1075780	44048	14432	S	1.0	4.3	0:01.19	node /var/www/h

Figura 13. Prueba de uso de recursos con 1000 llamadas en Escenario 1

La tabla 2 muestra los resultados del indicador Tiempo de Respuesta, Uso de Datos y Latencia para cada una de las diferentes cantidades de llamadas, en donde se aumentó de 200 en 200 hasta generar problemas dentro del servidor. Se recopiló toda la información de las pruebas obtenidas identificando que para una cantidad de 1000 llamadas el sistema ya genera problemas con únicamente 606 llamadas efectuadas, es decir se tiene un 39.4% de errores.

Tabla 2. Recopilación de resultados obtenidos del escenario 1 hasta 1000 llamadas.

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

## ESCENARIO DE PRUEBA 1 - CONEXIÓN DIRECTA

N°	Canti	Llam	%	Llam	Tiempo	Paq	Paq	Late	CPU	RAM
Pru	dad	adas	Llam	adas	de	uete	uete	ncia	%	%
eba	de	Corre	adas	por	Respuest	1	2	(ms)		
	Llam	ctas	Corre	Segu	a					
	adas		ctas	ndo	Llamada					
					s (ms)					
1	200	195	97,5	5	20	41,4	41,4	1,73	7,65	6,56%
						575	402		%	
2	400	313	78,25	5,347	18,7020	9,25	9,24	1,01	9,60	6,60%
						69	68	18	%	
3	600	428	71,33	5,242	19,0766	51,7	51,7	0,47	9,30	6,60%
			33			612	565	27	%	
4	800	532	66,5	2,339	42,7533	1,10	1,05	4,72	8,00	6,60%
						09	374	08	%	
5	1000	606	60,6	2,386	41,9111	29,5	29,5	2,49	7,70	6,70%
						655	405	55	%	

## 5.2 Resultados Escenario II

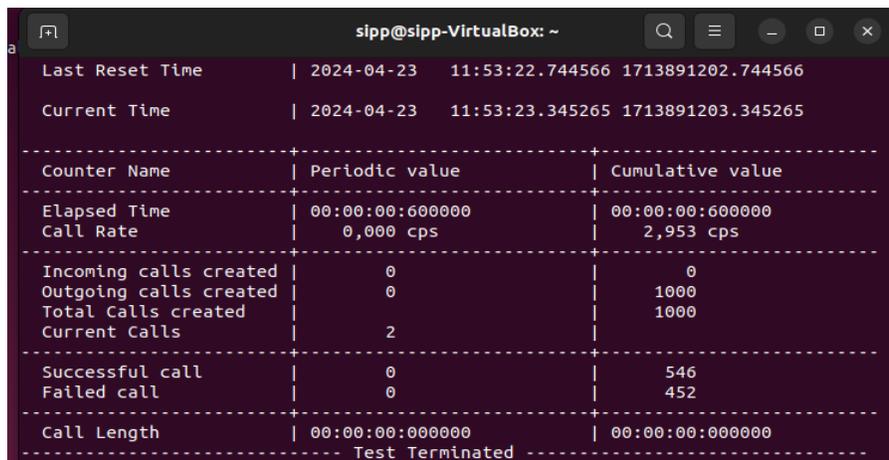
La Figura 14 muestra los resultados del uso de recursos del servidor previo a realizar el test de estrés, mediante el uso del comando TOP. Se muestra que el servidor de Asterisk tiene un consumo del 7.7% del CPU y un 6.7% de la memoria. Posteriormente se procedió a establecer una cantidad de 1000 llamadas por prueba que exista problemas en la recepción de llamadas o existan algunas que no se efectúen.

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2884	asterisk	20	0	1680816	67500	19108	S	7.7	6.7	0:11.81	asterisk
2149	mysql	20	0	1171400	103428	8496	S	6.7	10.2	0:09.93	mysqld
3190	asterisk	20	0	1076004	58068	14436	S	1.3	5.7	0:02.47	node /var/www/h
2157	mongod	20	0	478860	56708	20624	S	1.0	5.6	0:03.55	mongod
3	root	20	0	0	0	0	S	0.7	0.0	0:02.63	ksoftirqd/0
93	root	20	0	0	0	0	S	0.7	0.0	0:02.78	kworker/0:3
9	root	20	0	0	0	0	S	0.3	0.0	0:01.08	rcu_sched

Figura 14. Estado de recursos del servidor pretest escenario 2

En la figura 15 se visualiza los resultados obtenidos por el simulador una vez ejecutado y culminada la prueba de estrés para 1000 llamadas en el escenario 2. Mostrándose qué se tiene un promedio de 2,953 de llamadas por minuto, lo que permite estimar el tiempo de respuesta para cada una de las llamadas, en este caso, corresponde a 33,86 ms.



Counter Name	Periodic value	Cumulative value
Last Reset Time	2024-04-23 11:53:22.744566	1713891202.744566
Current Time	2024-04-23 11:53:23.345265	1713891203.345265
Elapsed Time	00:00:00:600000	00:00:00:600000
Call Rate	0,000 cps	2,953 cps
Incoming calls created	0	0
Outgoing calls created	0	1000
Total Calls created		1000
Current Calls	2	
Successful call	0	546
Failed call	0	452
Call Length	00:00:00:000000	00:00:00:000000

Figura 15. Resultado de prueba estrés 100 llamadas en escenario 2

La figura 16 muestra los resultados de la latencia existente en el servicio, con estas pruebas de estrés se visualizan los archivos logs generados por el simulador, en donde se toma la hora del paquete recibido y se resta de la hora del paquete transmitido obteniendo un valor de latencia de 0.2661 ms.

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

```

41072 Content-Length: 0
41073
41074
41075 ----- 2024-04-23 11:48:01.275311
41076 UDP message sent (384 bytes):
41077
41078 INVITE sip:2002@[2003:db8:b::10]:5060 SIP/2.0
41079 Via: SIP/2.0/UDP [2003:db8:d::10]:5060
41080 From: 1888 <sip:1888@[2003:db8:d::10]:5060>;tag=5068SIPpTag00133
41081 To: 2002 <sip:2002@[2003:db8:b::10]:5060>
41082 Call-ID: 133-5068@2003:db8:d::10
41083 CSeq: 1 INVITE
41084 Contact: sip:1888@[2003:db8:d::10]:5060
41085 Max-Forwards: 70
41086 Subject: Performance Test
41087 Content-Type: application/sdp
41088 Content-Length: 0
41089
41090
41091 ----- 2024-04-23 11:48:01.277972
41092 UDP message received [961] bytes :
41093
41094 SIP/2.0 200 OK
41095 Via: SIP/2.0/UDP [2003:db8:d::10]:5060;rport=5060;received=2003:db8:d::10
41096 Call-ID: 103-5068@2003:db8:d::10
41097 From: "1888" <sip:1888@[2003:db8:d::10]>;tag=5068SIPpTag00103
41098 To: "2002" <sip:2002@[2003:db8:b::10]>;tag=115a152b-5639-412b-9039-b19e03133b10
41099 CSeq: 2 INVITE
41100 Server: FPBX-14.0.17(13.38.3)
    
```

Figura 16. Respuesta de log de SIPp para calcular la latencia del escenario 2

En la figura 17 se observa un porcentaje del 6.3% para el uso del CPU y un 6.0% en uso de memoria

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2892	asterisk	20	0	1681060	61076	19096	S	6.3	6.0	0:03.85	asterisk
2153	mysql	20	0	1171468	100944	8616	S	4.7	9.9	0:02.83	mysqld
2159	mongodb	20	0	478860	53912	17492	R	1.0	5.3	0:02.49	mongod
3232	asterisk	20	0	1075776	44776	14424	S	1.0	4.4	0:01.14	node /var/www/h
25	root	20	0	0	0	0	S	0.7	0.0	0:01.52	kworker/0:1
3	root	20	0	0	0	0	S	0.3	0.0	0:02.17	ksoftirqd/0
1867	redis	20	0	143060	7916	1520	S	0.3	0.8	0:00.26	redis-server
3176	asterisk	20	0	917388	34212	11684	S	0.3	3.4	0:00.66	PM2 v2.10.7: Go

Figura 17. Uso de recursos - Servidor Asterisk durante la prueba de estrés del Escenario 2

En la Tabla 3 se registra la recopilación de todos los datos obtenidos con el incremento de 200 llamadas para cada prueba. Se puede observar los resultados recopilados en cada uno de estos escenarios de prueba, donde se identifica que, al llegar a 600 llamadas, el Webservice comienza a experimentar problemas de comunicación y genera una alta cantidad de llamadas rechazadas, equivalentes en este caso al 55.16% del total de llamadas generadas. Además, se observa un promedio de tiempo de respuesta de 27.225 ms y una latencia de 0.017 ms."

## Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

Tabla 3. Recopilación de resultados obtenidos del escenario 2 hasta 1000 llamadas

ESCENARIO DE PRUEBA 2 – WEBSERVICE										
N° Prue ba	Cantida d de Llamad as	Llamad as Correct as	% Llamad as Fallidas	Llamad as por Segund o	Tiempo					
					de Respu sta Llamad as (ms)	Paque te 1	Paque te 2	Latenc ia (ms)	CP U %	RA M %
1	200	194	97	1,509	66,2690	10,53 93	10,539 1	0,01 8	6,30 %	6,00 %
2	400	317	79,25	5,115	19,5503	25,48 44	25,478 3	0,61 5	7,00 %	6,60 %
3	600	269	44,8333	3,673	27,2257	59,50 55	59,505 3	0,01 7	6,60 %	6,60 %
4	800	401	50,125	2,424	41,2541	21,91 22	21,905 58	0,66 6	6,60 %	6,60 %
5	1000	546	54,6	2,953	33,8638	1,277 97	1,2753 11	0,26 6	6,60 %	6,60 %

En resumen, los resultados obtenidos de los dos escenarios se describen a continuación:

- Latencia promedio: 84.2 ms (ODBC) vs 86 ms (WebService).
- Tiempo de respuesta: 112.6 ms (ODBC) vs 121.7 ms (WebService).
- Uso de CPU: 53.21% (ODBC) vs 54.95% (WebService).

## 6. Discusión

---

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

Las pruebas de estrés evidenciaron la diferencia de rendimiento obtenidas con las pruebas de esfuerzo entre cada uno de los escenarios propuestos e implementados. En el caso del escenario con conexión directa de la base de datos, se evidencia una capacidad de 606 llamadas concurrentes sin inconvenientes puesto que de las 1000 llamadas establecidas en SIPp el 33.5% presentan fallos. En el caso del escenario con Webservice, se evidencia una capacidad de 546 llamadas concurrentes sin inconvenientes considerando que de las 1000 llamadas establecidas en SIPp el 54.6% presentan fallos. Otro de los parámetros considerados a medir en la presente investigación es el tiempo de respuesta, evidenciando en el escenario con conexión directa a la base de datos un promedio de 28.48 ms mientras que para conexión con Webservice se obtienen en promedio un 37.63 ms, lo que representa una diferencia de 9.15 ms entre ambos escenarios.

Para el caso de la latencia se evidencia para el escenario con conexión directa a la base de datos se obtiene una latencia promedio de 2.08 ms, mientras que para el escenario con Webservice se obtiene una latencia de 0.31 ms, lo que representa una diferencia mínima de 1.8 ms entre ambos escenarios. Según el trabajo realizado por Cedeño, Zambrano y Zambrano (2021), indican que para percibir una calidad de voz de excelencia se requiere poseer una latencia entre 0 a 150 ms, lo que indica que el sistema cumple con este estándar para ambos escenarios. Finalmente considerando el uso de recursos la capacidad del CPU y memoria RAM, se han obtenido resultados muy cercanos para ambos escenarios. Para el caso del escenario con conexión directa a la base de datos se ha obtenido un promedio de 8.37% de consumo de CPU y de memoria RAM un 6.59%, mientras que para el escenario con Webservice se obtuvo un porcentaje de uso del CPU de 6.63% y de memoria RAM un 6.45%. Existiendo una diferencia de 1.74% en el uso del CPU mientras que en el uso de memoria esto se reduce al 0.04%.

## 7.- Conclusiones

Se identificó varias opciones que permiten la integración entre Asterisk y la base de datos PostgreSQL con protocolo IPv6, determinando dos alternativas e integración para una conexión directa a la base de datos ODBC y JDBC, escogiendo ODBC por razones técnicas de compatibilidad. Por otro lado, para la integración por Webservice se determina como mejor alternativa para la integración AGI en lugar de FastAGI, ya que permite ejecutar comandos dentro del Dialplan implementado con códigos

---

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

externos como PHP, Python entre otros. El desarrollo del prototipo en GNS3 permitió integrar un Servidor Asterisk con la base de datos PostgreSQL con protocolo IPv6, a través del cual el Softphone Zoiper con dirección IPv6 puede realizar satisfactoriamente la consulta de notas académicas, a través de una interacción con un IVR implementado en Asterisk; que habilita dos opciones; la consulta por materia o la consulta de todas las materias. Para la funcionalidad de la integración con conexión directa a la base de datos se habilita la extensión 2001 y para la funcionalidad con integración por Webservice se habilita la extensión 2002. El rendimiento de la consulta de notas académicas utilizando la primera opción de integración con conexión directa a la base de datos utilizando ODBC, respecto a la segunda opción de integración con Webservice utilizando AGI, referente a tiempo de respuesta es mejor con una diferencia de 9.15 ms, respecto a latencia el primer escenario es mejor con una diferencia de 1.8 ms y respecto al uso de CPU y memoria RAM existe una diferencia de 1.74% en el uso del CPU mientras que en el uso de memoria esto se reduce al 0.04%. Los resultados de rendimiento de las dos alternativas de integración utilizadas en esta investigación, permiten dar una visión técnica para la implementación de esta solución en otros campos de aplicación que no sean la consulta de notas, teniendo una visión clara de la concurrencia que cada una de las tecnologías puede soportar y la cantidad de usuarios finales que organizaciones o instituciones tengan y requieran.

## Referencias

- Salcan Chisaguano, C. F. (2024). Prototipo de un Sistema en GNS3 con la Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas. Universidad Nacional de Chimborazo.
- Rojas, L. (2013). Implementación de un sistema IVR para la consulta remota de inventario en la empresa Innova Networks mediante Asterisk y AGI-PHP.
- Allauca, L. G., et al. (2016). Alternativas de Text To Speech para la Implementación de Servicios de VoIP. TICAL.
- Ramírez, J., & Torres, M. (2023). Evaluación del rendimiento de llamadas VoIP mediante simulación SIP con SIPp. *Revista Latinoamericana de Tecnologías de la Información*, 15(2), 45–56.
- Digium Inc. (2023). *Asterisk: The Definitive Guide*.
- PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*.
- GNS3. (2024). *GNS3 User Documentation*. <https://www.gns3.com>
- Aalam, Z., Kumar, V., & Gour, S. (2021). A review paper on hypervisor and virtual machine security. *Journal of Physics*, 19(5), 1-9. doi:10.1088/1742- 6596/1950/1/012027
- Aguirre, V. (2023). Diagnóstico y perspectivas de la implementación de IPv6 en el Ecuador. [Tesis de pregrado, Escuela Politécnica Nacional], Repositorio Institucional epn. Obtenido de <https://bibdigital.epn.edu.ec/handle/15000/23618>
- AJPD soft. (2020). Primer proyecto de laboratorio de red virtual con GNS3 en Windows e instalación de router Cisco 7200. Obtenido de [https://proyectoa.com/primer-proyecto-de-laboratorio-de-red-virtual-con-gns3-e-instalacion-de-router-cisco-7200/#google\\_vignette](https://proyectoa.com/primer-proyecto-de-laboratorio-de-red-virtual-con-gns3-e-instalacion-de-router-cisco-7200/#google_vignette)
- Arranz, U. (2016). Propuesta de infraestructura de integración de aplicaciones basado en herramientas para la Universidad de las Ciencias Informáticas. Tesis de Grado, Repositorio UCI.

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

- Bjarnason, E., Lang, F., & Mjöberg, A. (2023). An empirically based model of software prototyping: a mapping study and a multi-case study. *Empirical Software Engineering*, 28(115), 1-47. Obtenido de <https://link.springer.com/article/10.1007/s10664-023-10331-w>
- Bustos, S. (2023). Desarrollo de un plan de migración de una red de área extensa de un proveedor de servicios de internet a una red definida por software SD-WAN. [Tesis de pregrado, Universidad Técnica del Norte], Repositorio Institucional utn. Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/15328/2/04%20RED%20367%20TRABAJO%20GRADO.pdf>
- Castro, D. (2022). Control de un hogar domotizado a través de Asterisk. [Tesis de maestría, Universidad de Oviedo], Repositorio Institucional uniovi. Obtenido de <https://digibuo.uniovi.es/dspace/handle/10651/64298?locale-attribute=en>
- Cedeño, S., Zambrano, D., & Zambrano, W. (2021). Revisión sistemática de Comunicaciones Unificadas de VoIP en redes CAN. *Revista de Tecnologías de la Informática y las Telecomunicaciones*, 5(1), 17-34.  
doi:<https://doi.org/10.33936/isrtic.v5i1.3569>
- Chen, Y. (2020). Research and Implementation of Remote Sensing Image Database Technology Based on Oracle9i. *Journal of Physics*, 1(3), 1-5. doi:10.1088/1742-6596/1533/4/042002
- Deka, A., & Kumari, B. (2019). A DTMF based Railway Enquiry System. *Computing for Sustainable Global Development*, 1(2), 51-54. Obtenido de <https://ieeexplore.ieee.org/document/8991235>
- Fernández, Y. (2023). IPv6: qué es, para qué sirve y qué ventajas tiene . Obtenido de <https://www.xataka.com>: <https://www.xataka.com/basics/ipv6-que-sirve-que-ventajas-tiene>
- formaciongcc.com. (2020). Centralita Asterisk. Solución VoIP para call center. Obtenido de <https://www.formaciongcc.com>: <https://www.formaciongcc.com/centralita-asterisk/>
- GNS3. (2024). The software that empowers network professionals. Obtenido de <https://www.gns3.com/>

Programación de Alternativas de Integración de Asterisk y PostgreSQL sobre IPv6 para Consulta de Notas Académicas en GNS3

---

Guerrero, J., & Guerrero, J. E. (2010). Definición de un Procedimiento de Pruebas para Definir la Capacidad, Disponibilidad y QoS de un Servidor de Asterisk. [Tesis de Grado], Repositorio ESPOL.

©2025 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).