**UNIVERSIDAD TECNOLÓGICA DE PARRAL**

****

LINEAMIENTOS PARA ELABORAR EL REPORTE TÉCNICO DE

ESTADÍAS

****

REPORTE DE ESTADÍA

“SERVIDOR DE TELECOMUNICACIONES VOIP ASTERISK, INTEGRACIÓN A BASE DE DATOS RIS”

QUE COMO REQUISITO PARA OBTENER EL TÍTULO DE:

T.S.U

TÉCNICO SUPERIOR UNIVERSITARIO EN:

TECNOLOGIAS DE LA INFORMACION AREA INFRAESTRUCTURA DE REDES DIGITALES

PRESENTA:

JUAN LUIS QUEZADA RASCON

ASESOR EMPRESARIAL:

ING. JUAN EMMANUEL RAMÍREZ MONJE

ASESOR ACADÉMICO:

LUIS IVAN GARCIA GONZALEZ

TUTOR ACADÉMICO:

JOSE IRVING MARTINEZ SAENZ

Mayo, 2022

**“SERVIDOR DE TELECOMUNICACIONES VOIP ASTERISK, INTEGRACIÓN A BASE DE DATOS RIS”**

Reporte Técnico de Estadías realizada por **Juan Luis Quezada Rascón** bajo la dirección del comité revisor indicado, aprobada por el mismo y aceptada como requisito parcial para obtener el título de:

**TÉCNICO SUPERIOR UNIVERSITARIO EN:**

TECNOLOGIAS DE LA INFORMACION AREA INFRAESTRUCTURA DE REDES DIGITALES

COMITÉ REVISOR:

Ing. Juan Emmanuel Ramírez Monje

Asesor Empresarial

Luis Iván García González José Irving Martínez Sáenz

Asesor Académico Tutor Académico

**CONSTANCIA DE TERMINACIÓN DE ESTADÍA**

**Hgo. del Parral, Chih. a \_\_\_\_ de \_\_\_\_\_\_\_\_\_\_\_\_ de 20\_\_**

**ING. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**JEFE DE LA CARRERA DE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PRESENTE. -**

En atención al convenio y/o acuerdo celebrado con la Universidad Tecnológica de Parral, hago constar que el(la) C. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, alumno(a) de la carrera de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, con número de matrícula \_\_\_\_\_\_\_\_\_\_\_\_; culminó con su programa de Estadías, en el que realizó el proyecto titulado **“\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”** en nuestras instalaciones, durante el periodo cuatrimestral comprendido del \_\_\_ de \_\_\_\_\_\_\_\_ al \_\_\_ de \_\_\_\_\_\_\_\_\_ del año en curso, cubriendo un total de 525 horas, de acuerdo al programa y plan de estudio de la carrera correspondiente, siendo su tutor por parte de esta empresa, el C. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Sin otro particular por el momento y en espera de que la información anterior cumpla con los requisitos de la Universidad Tecnológica de Parral, me despido de usted no sin antes ponerme a sus órdenes para cualquier aclaración.

**A T E N T A M E N T E**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(NOMBRE, FIRMA Y PUESTO DE PERSONA AUTORIZADA

POR LA EMPRESA PARA ASIGNAR LA LIBERACIÓN

DEL ALUMNO)

c.c.p. Departamento de Vinculación: Constancia de Servicio Soci

**DEDICATORIA.**

**AGRADECIMIENTOS.**

**RESUMEN.**

**INDICE:**

[INTRODUCCION 1](#_Toc104718395)

[JUSTIFICACION 1](#_Toc104718396)

[OBJETIVOS DEL PROYECTO 1](#_Toc104718397)

[MARCO TEÓRICO 1](#_Toc104718398)

[DESARROLLO DE LA ESTADIA 1](#_Toc104718399)

[ANALISIS DE LOS RESULTADOS 1](#_Toc104718400)

[REFERENCIA BIBLIOGRAFICA 1](#_Toc104718401)

[APÉNDICE 1](#_Toc104718402)

**INDICE DE FIGURAS.**

# INTRODUCCION:

Es un gabinete radiológico digital, el mejor equipado al sur de chihuahua, cuenta con la más alta calidad y personal altamente calificado, cuenta con áreas como lo son:

* Resonancia Magnética
* Tomografía Computada
* Ultrasonido
* Rayos X Digital
* Densitometría ósea
* Mastografía Digital con Tomosíntesis
* Espirometría
* Audiometría
* Ortopantomografía Digital
* Tomografía Dental
* RX Especiales

# JUSTIFICACION:

## Delimitación del proyecto.

Actualmente la empresa Diagnocons se encuentra en remodelación y uno de sus objetivos es mejorar su infraestructura de telecomunicaciones que ayude a mejorar sus comunicaciones y operaciones para ofrecer un mejor servicio.

# OBJETIVOS DEL PROYECTO:

## Objetivo general.

Desarrollar un Servidor de telecomunicaciones VOIP ASTERISK y que tenga una integración a una base de datos RIS.

## Objetivos específicos.

* Diseño de red.
* Instalación del servidor
* Configuración del servidor
* Pruebas.

# MARCO TEÓRICO:

## Fundamentos de VOIP

VoIP (Voice Over Internet Protocol) o Voz sobre IP es transmitir la voz sobre una red de datos, la comunicación de voz es analógica por lo que lo primero que hay que hacer es convertir esa voz analógica en digital, después esa señal digital se va a convertir en un paquete IP mediante un proceso de compresión o denominado como un códec o codificación para que pueda viajar a través de la red. Luego en el otro extremo llega el paquete y hace el proceso de descompresión y luego lo convierte a digital y por último lo convierte a analógico. El dispositivo que se encarga de transformar de analógico a la señal digital se llama terminal puede ser ya sea un micrófono o un teléfono.

**Conceptos**

* Protocolo de señalización: mensajes necesarios de coordinación previa al establecimiento del audio, es decir, que se van a ejecutar durante la llamada y los mensajes que se ejecutan al finalizar la llamada.
  + Protocolo SIP: Protocolo estándar y moderno.
    - Es un protocolo desarrollado por la DTF que fue publicado en RFC 32 6:1.
    - Viaja sobre UDP O TCP comúnmente en el puerto 50/60 para realizar toda la señalización.
    - Protocolo simple, fácil de implementar y es de los más usados.
    - Necesita el protocolo RTP (Real Time Protocol) que sirva para transportar el audio.
  + Protocolo H323: Protocolo estándar y antiguo.
    - Viaja sobre TCP comúnmente en el puerto 1720.
    - Desarrollado por la ITU-T.
    - Protocolo complejo, no muy usado hoy en día, sin embargo, útil para comunicarse con centrales propietarias antiguas.
    - Necesita el protocolo RTP (Real Time Protocol) que sirva para transportar el audio.
  + Protocolo IAX2: Protocolo nuevo, propio de Asterisk
    - Viaja sobre UDP comúnmente en el puerto 4569.
    - Aparte de señalización es protocolo de Trafico por lo que es útil en troncales cuando es fuera de su red.
* Protocolo de tráfico: transporta el audio, es decir el flujo de conversación, la data del audio que es donde ya intervienen los usuarios.
* Protocolo RTP (Real Time Protocol):
  + Definido por el IETF, usa el estándar RFC 3550.
  + Usa UDP, los puertos en el rango de 10000 a 20000.
  + Es el encargado de llevar el flujo de audio o video.
  + Adicionalmente lo controla el protocolo RTCP (Real Time Control Protocol), el cual monitorea estadísticas de calidad de servicio como perdida de paquetes, así como información de los usuarios.
* Códecs: también denominado compresor es aquel que va a convertir nuestra información inicial en otro formato que lo pueda comprimir nuestra información inicial para el tema de ahorro de ancho de banda. El códec está directamente relacionado a dos puntos importantes:
  + Define la calidad de la llamada, por que a mayor compresión menor calidad de la llamada.
  + Define el ancho de banda que va a usar.
  + Códecs más comunes:
    - G711, trabaja a 64 kbps, es conocido en Asterisk como ulaw y/o alaw, este códec no comprime prácticamente nada por lo que brinda la mejor calidad de la llamada, pero consume un mayor ancho de banda.
    - GSM, trabaja a 13kbps.
    - G729, trabaja a 8kbps, es protocolo propietario, es decir no es un códec libre, hay que pagar por él, consume poco ancho de banda, pero te da una calidad de llamada aceptable.
    - G721, trabaja a 32kbps.

Nota: las centrales telefónicas deben soportar el códec de extremo a extremo.

## ¿Qué es un sistema operativo?

Es el software que coordina y dirige todos los servicios y aplicaciones que utiliza el usuario en una computadora, por eso es el más importante y fundamental.

Los sistemas operativos, también llamados núcleos o kernels, suelen ejecutarse de manera privilegiada respecto al resto del software, sin permitir que un programa cualquiera realice cambios de importancia sobre él que puedan comprometer su funcionamiento.

https://concepto.de/sistema-operativo/

## ¿Qué es Debian?

Es una distribución del Sistema operativo GNU/Linux compuesto por más de 40000 paquetes de software precompilados, mayormente software libre y de código abierto bajo la Licencia Pública General GPL que incluye a Linux como principal núcleo del sistema y a otros como Hurd.

Debian soporta en mayor o menor medida varios tipos de procesadores y puede ser utilizado como sistema operativo para computadoras personales, así como para servidores. Se enfoca en la estabilidad y seguridad y es usada como base para muchas otras distribuciones de Linux.

<https://www.ecured.cu/Debian>

## ¿Qué es Rufus?

Es una utilidad que le ayuda a formatear y crear soportes USB de arranque, como «pendrives», tarjetas de memoria, etcétera.

Es especialmente útil en casos donde:

* necesite crear medios de instalación USB a partir de ISOs arrancables (Windows, Linux, UEFI, etc.)
* necesite trabajar en un equipo que no tenga un sistema operativo instalado
* necesite actualizar el firmware o BIOS de un ordenador desde DOS
* quiera ejecutar una utilidad de bajo nivel

<https://rufus.ie/es/>

## ¿Qué es Putty?

Es una terminal de simulación open source que fue desarrollado para actuar como cliente de conexiones seguras a través de protocolos raw TCP, Telnet, rlogin y portal serial. Por lo tanto, este software se indica para establecer conexiones seguras de acceso remoto a servidores a través de Shell Seguro (SSH y para construir canales encriptados entre servidores. Este nos servirá para controlar el servidor desde otro terminal.

<https://www.hostgator.mx/blog/putty-en-programacion-aprende-que-es/>

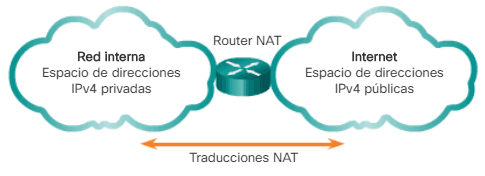
## ¿Qué es packet tracer?

Packet Tracer es la herramienta de aprendizaje y simulación de redes interactiva para los instructores y alumnos de Cisco CCNA. Esta herramienta les permite a los usuarios crear topologías de red, configurar dispositivos, insertar paquetes y simular una red con múltiples representaciones visuales. Packet Tracer se enfoca en apoyar mejor los protocolos de redes que se enseñan en el currículum de CCNA.

<https://erickosvaldovg.wordpress.com/2014/09/30/que-es-packet-tracer/#:~:text=Que%20es%20Packet%20Tracer%3F%20Packet%20Tracer%20es%20la,y%20simular%20una%20red%20con%20m%C3%BAltiples%20representaciones%20visuales>.

## NAT

NAT proporciona la traducción de direcciones privadas a direcciones públicas. Esto permite que un dispositivo con una dirección IPv4 privada acceda a recursos fuera de su red privada, como los que se encuentran en Internet. NAT tiene el beneficio adicional de proporcionar cierto grado de privacidad y seguridad adicional a una red, ya que oculta las direcciones IPv4 internas de las redes externas.



Según la terminología de NAT, la red interna es el conjunto de redes sujetas a traducción. La red externa se refiere a todas las otras redes.

Al utilizar NAT, las direcciones IPv4 se designan de distinto modo, según si están en la red privada o en la red pública (Internet), y si el tráfico es entrante o saliente.

**Términos interna, externa, global y local**

NAT incluye cuatro tipos de direcciones:

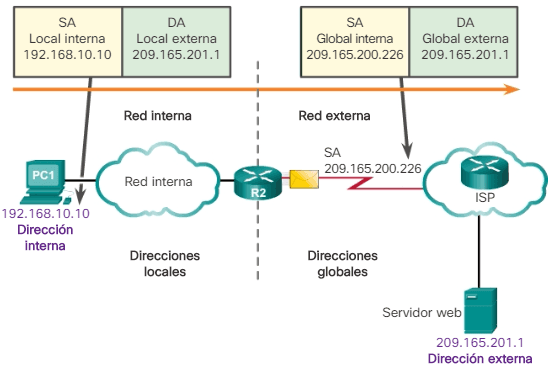
* Dirección local interna
* Dirección global interna
* Dirección local externa
* Dirección global externa

Al determinar qué tipo de dirección se utiliza, es importante recordar que la terminología de NAT siempre se aplica desde la perspectiva del dispositivo con la dirección traducida:

* Dirección interna: la dirección del dispositivo que se traduce por medio de NAT.
* Dirección externa: la dirección del dispositivo de destino.

NAT usa los conceptos de local o global con relación a las direcciones:

* Dirección local: cualquier dirección que aparece en la porción interna de la red.
* Dirección global: cualquier dirección que aparece en la porción externa de la red.

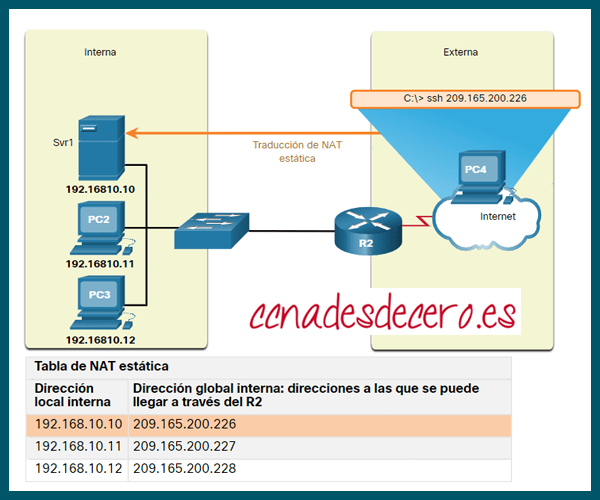


* **Dirección local interna:** la dirección de origen vista desde el interior de la red. En la ilustración, la dirección IPv4 192.168.10.10 se asignó a la PC1. Esta es la dirección local interna de la PC1.
* **Dirección global interna:** la dirección de origen vista desde la red externa. En la ilustración, cuando se envía el tráfico de la PC1 al servidor web en 209.165.201.1, el R2 traduce la dirección local interna a una dirección global interna. En este caso, el R2 cambia la dirección IPv4 de origen de 192.168.10.10 a 209.165.200.226. De acuerdo con la terminología de NAT, la dirección local interna 192.168.10.10 se traduce a la dirección global interna 209.165.200.226.
* **Dirección global externa:** la dirección del destino vista desde la red externa. Es una dirección IPv4 enrutable globalmente y asignada a un host en Internet. Por ejemplo, se puede llegar al servidor web en la dirección IPv4 209.165.201.1. Por lo general, las direcciones externas globales y locales son iguales.
* **Dirección local externa:** la dirección del destino vista desde la red interna. En este ejemplo, la PC1 envía tráfico al servidor web en la dirección IPv4 209.165.201.1. Si bien es poco frecuente, esta dirección podría ser diferente de la dirección globalmente enrutable del destino.

<https://ccnadesdecero.es/nat-network-address-translation/>

**TIPOS DE NAT**

**NAT Estático**



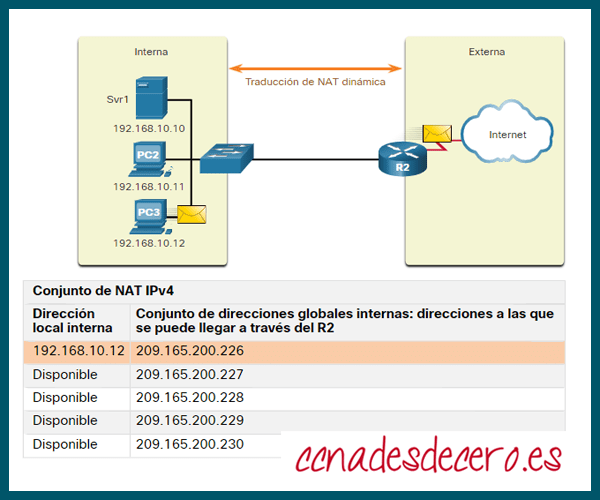
La NAT estática consiste en una asignación uno a uno entre direcciones locales y globales. Estas asignaciones son configuradas por el administrador de red y se mantienen constantes. En la ilustración, el R2 se configuró con las asignaciones estáticas para las direcciones locales internas del Svr1, la PC2 y la PC3. Cuando estos dispositivos envían tráfico a Internet, sus direcciones locales internas se traducen a las direcciones globales internas configuradas. Para redes externas, estos dispositivos parecen tener direcciones IPv4 públicas. En la ilustración, el R2 se configuró con las asignaciones estáticas para las direcciones locales internas del Svr1, la PC2 y la PC3. Cuando estos dispositivos envían tráfico a Internet, sus direcciones locales internas se traducen a las direcciones globales internas configuradas. Para redes externas, estos dispositivos parecen tener direcciones IPv4 públicas.

Por ejemplo, un administrador de red de PC4 puede usar SSH para obtener acceso a la dirección global interna de Svr1 (209.165.200.226). R2 traduce esta dirección global interna a la dirección local interna 192.168.10.10 y conecta la sesión a Svr1.

La NAT estática requiere que haya suficientes direcciones públicas disponibles para satisfacer la cantidad total de sesiones de usuario simultáneas.

**NAT Dinámica**

La NAT dinámica utiliza un pool de direcciones públicas y las asigna según el orden de llegada. Cuando un dispositivo interno solicita acceso a una red externa, la NAT dinámica asigna una dirección IPv4 pública disponible del pool.

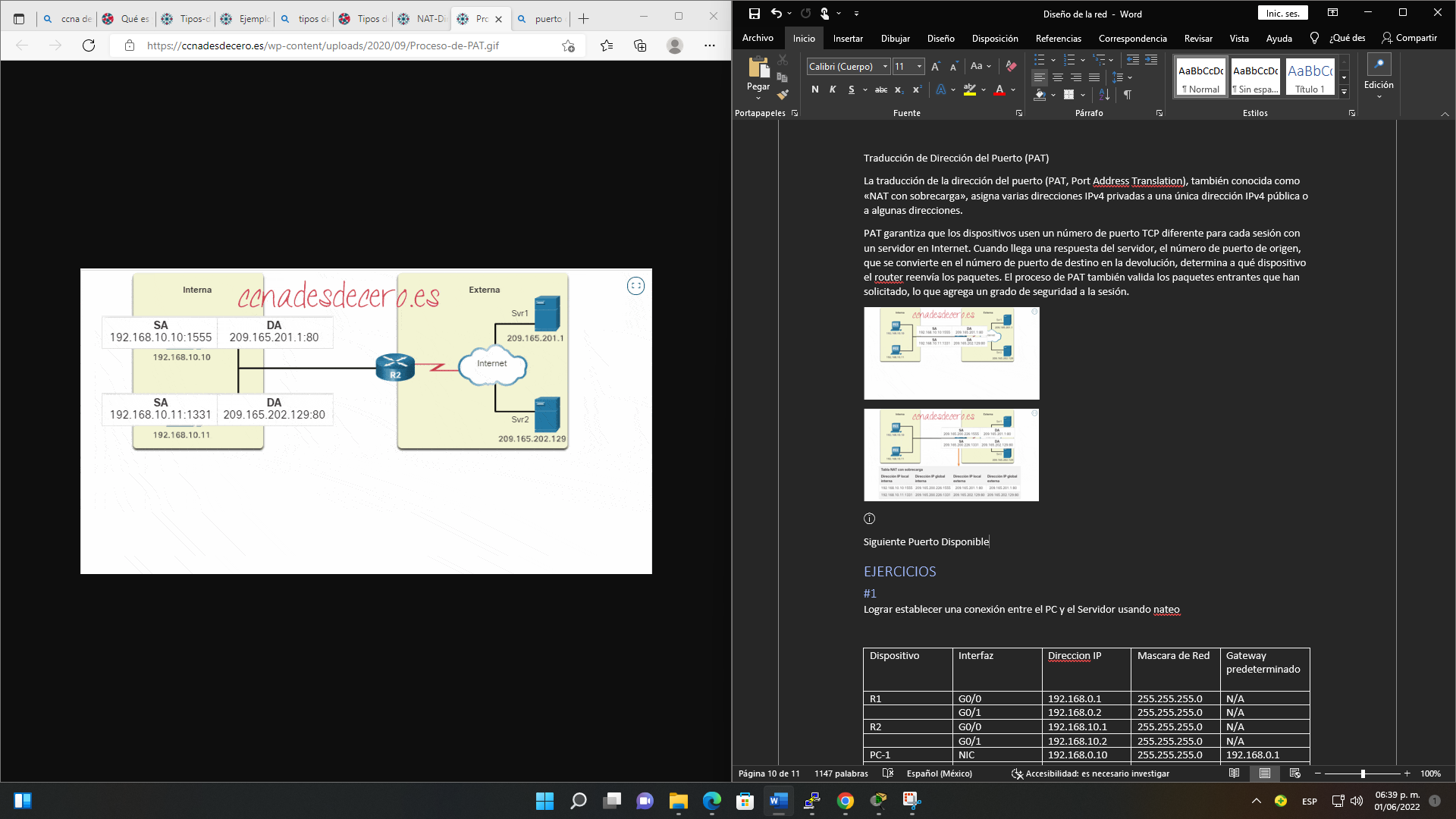


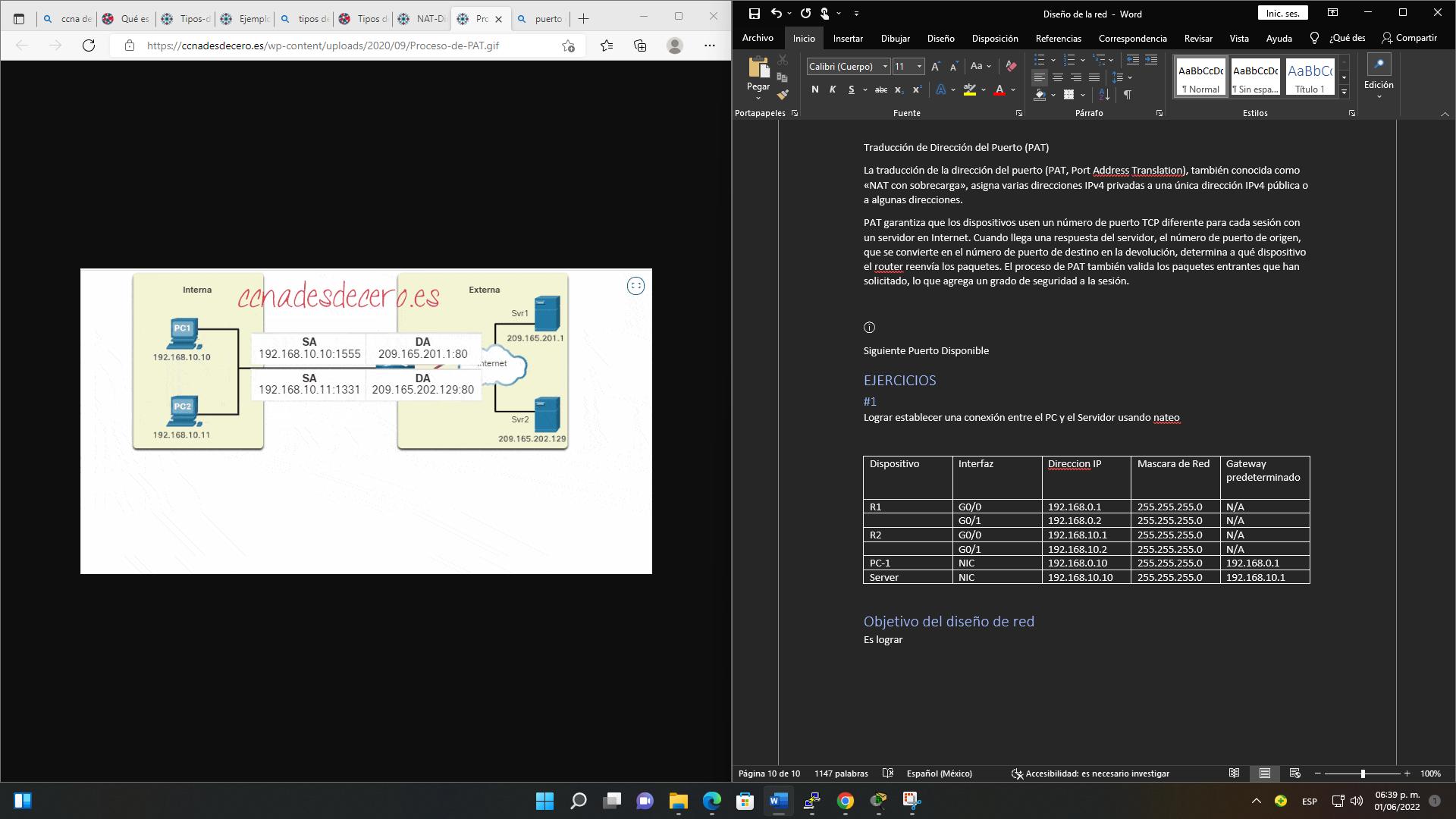
En la figura, PC3 ha accedido a Internet utilizando la primera dirección disponible en el pool NAT dinámico. Las demás direcciones siguen disponibles para utilizarlas. Al igual que la NAT estática, la NAT dinámica requiere que haya suficientes direcciones públicas disponibles para satisfacer la cantidad total de sesiones de usuario simultáneas.

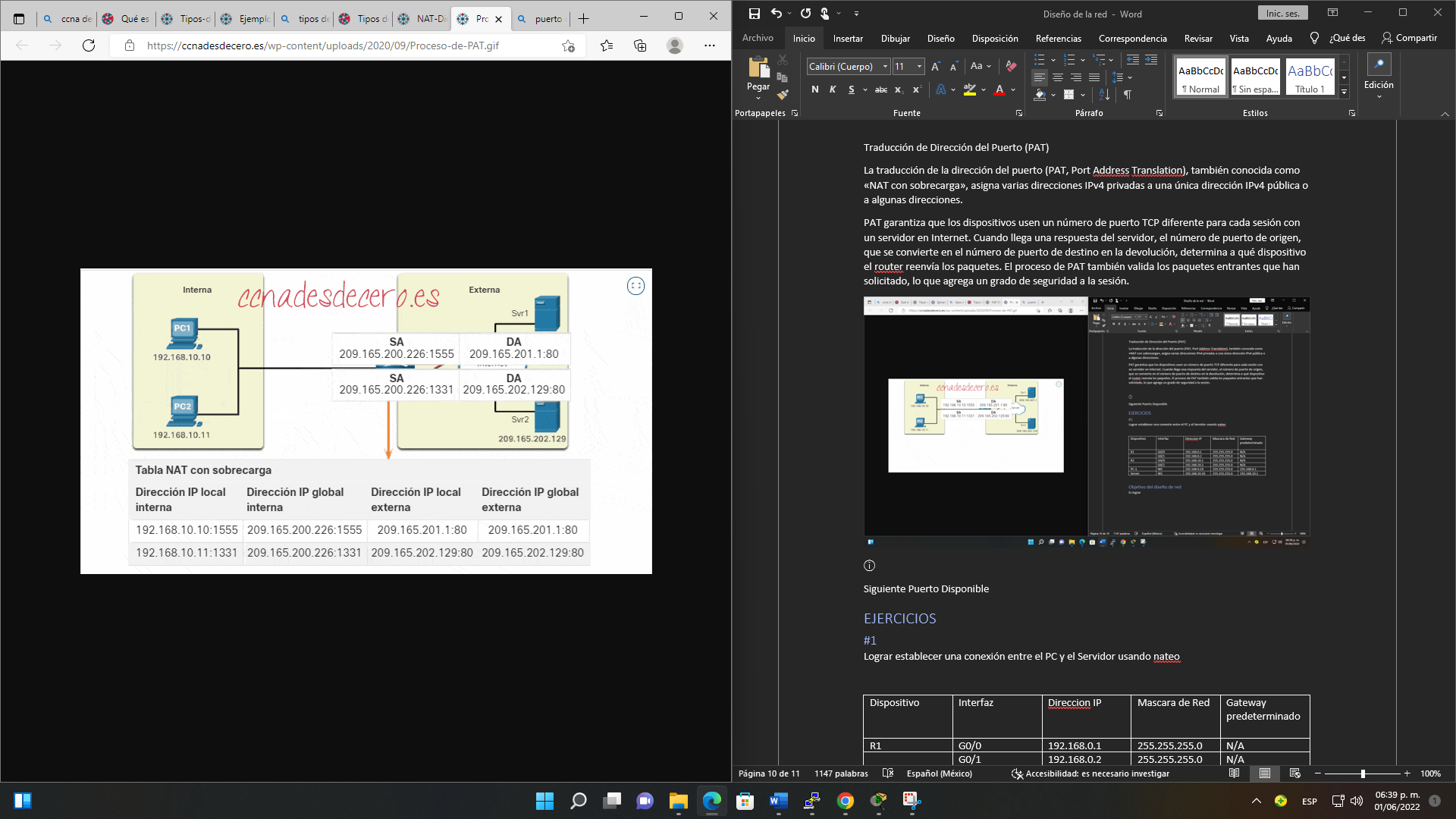
**Traducción de Dirección del Puerto (PAT)**

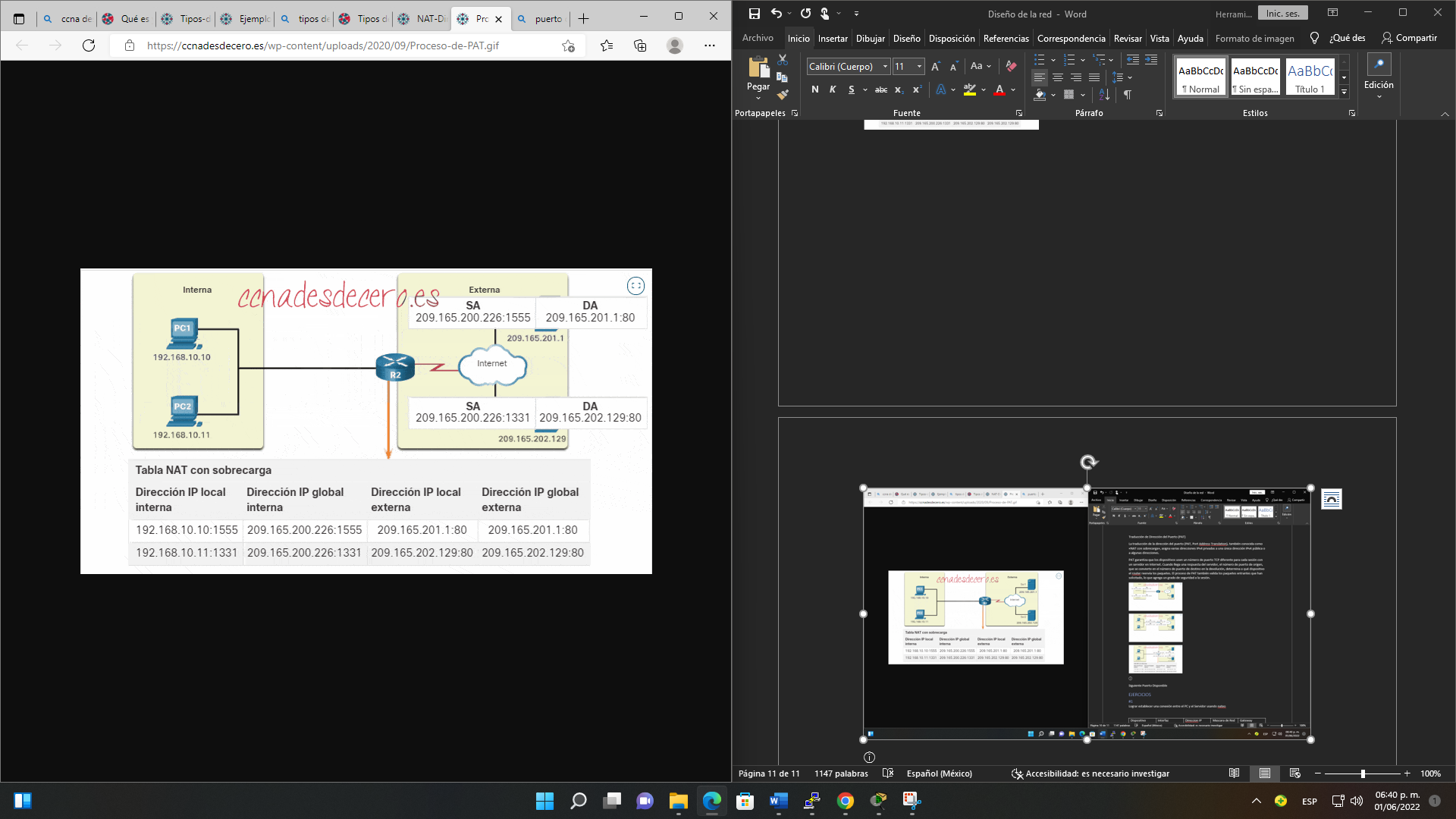
La traducción de la dirección del puerto (PAT, Port Address Translation), también conocida como «NAT con sobrecarga», asigna varias direcciones IPv4 privadas a una única dirección IPv4 pública o a algunas direcciones.

PAT garantiza que los dispositivos usen un número de puerto TCP diferente para cada sesión con un servidor en Internet. Cuando llega una respuesta del servidor, el número de puerto de origen, que se convierte en el número de puerto de destino en la devolución, determina a qué dispositivo el router reenvía los paquetes. El proceso de PAT también valida los paquetes entrantes que han solicitado, lo que agrega un grado de seguridad a la sesión.









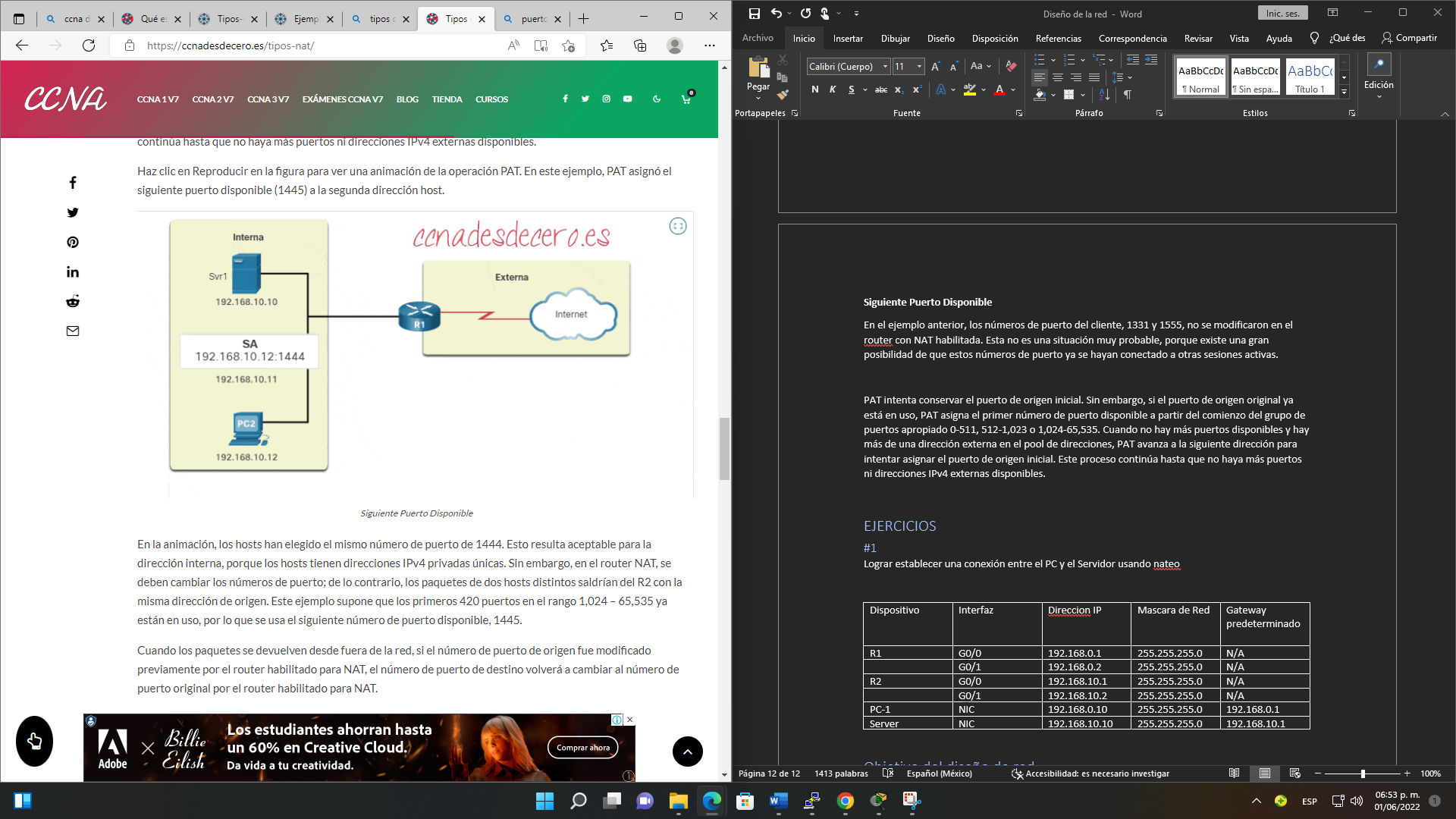
A medida que el R2 procesa cada paquete, utiliza un número de puerto (1331 y 1555, en este ejemplo) para identificar el dispositivo en el que se originó el paquete. La dirección de origen (SA, source address) es la dirección local interna con el número de puerto asignado TCP / UDP agregado. La dirección de destino (DA, destination address) es la dirección global externa con el número de puerto de servicio agregado. En este ejemplo, el puerto de servicio es 80, que es HTTP.

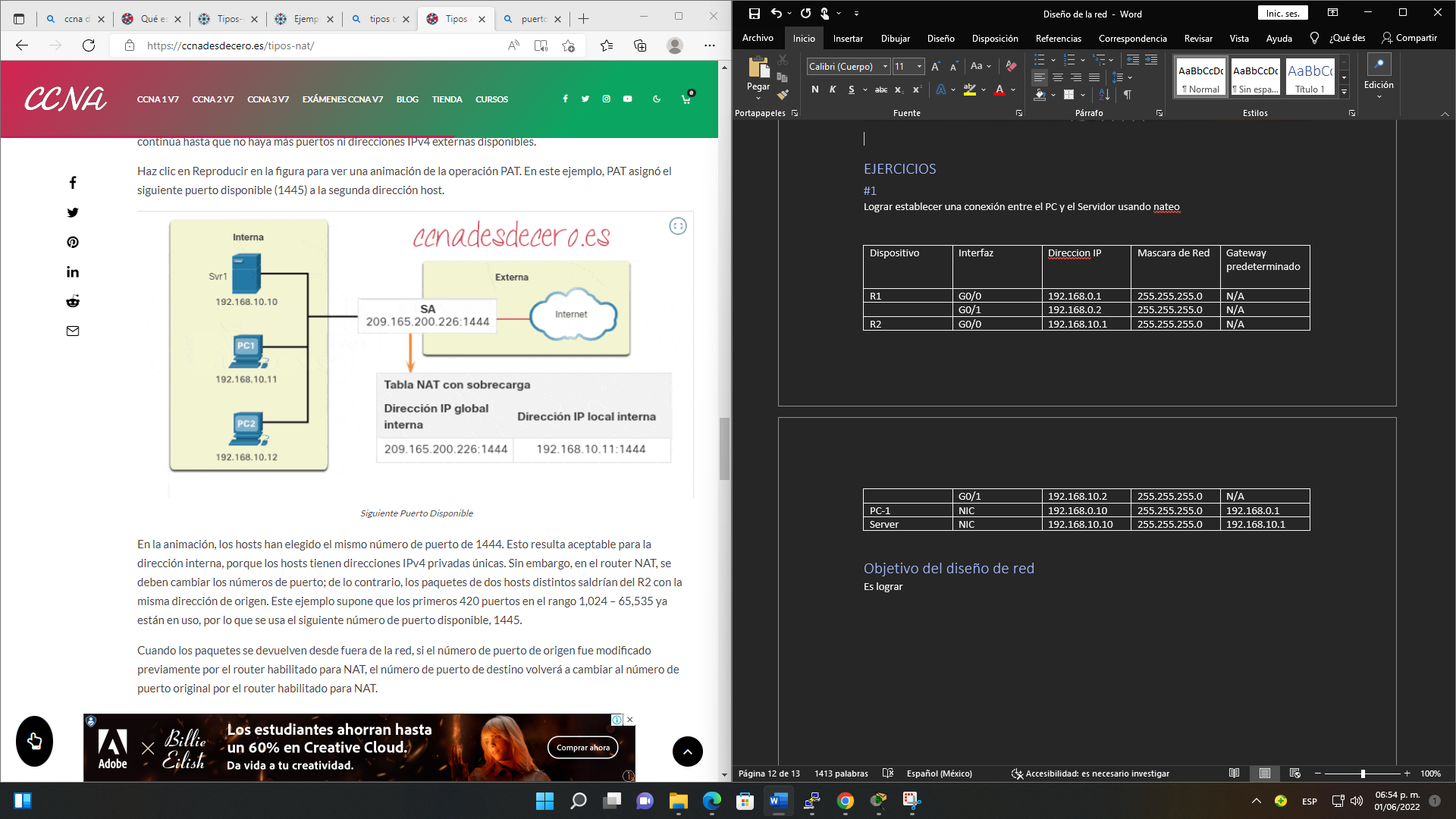
Para la dirección de origen, el R2 traduce la dirección local interna a una dirección global interna con el número de puerto agregado. La dirección de destino no cambia, pero ahora se conoce como la dirección IPv4 global externa. Cuando el servidor web responde, se invierte la ruta.

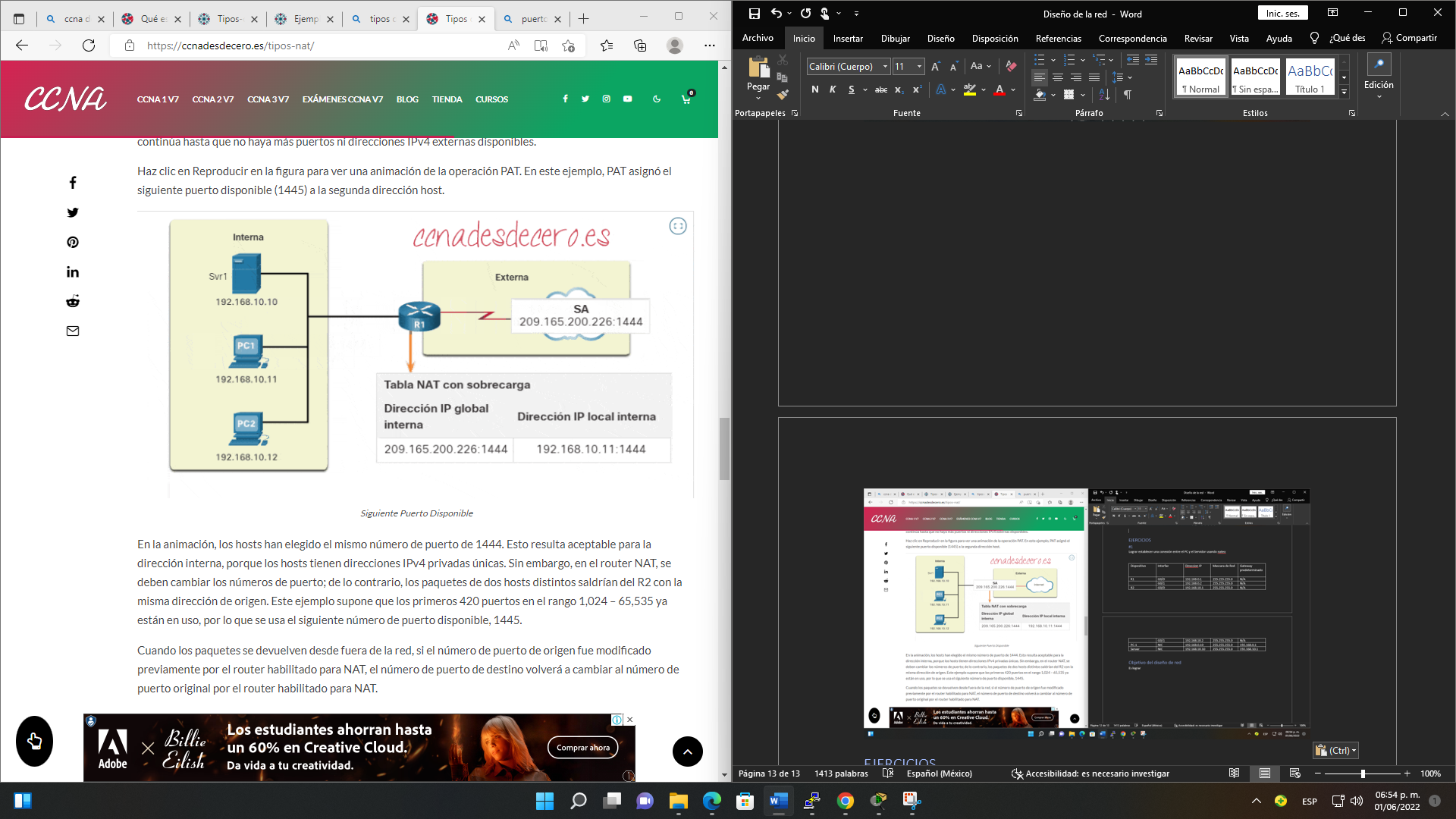
**Siguiente Puerto Disponible**

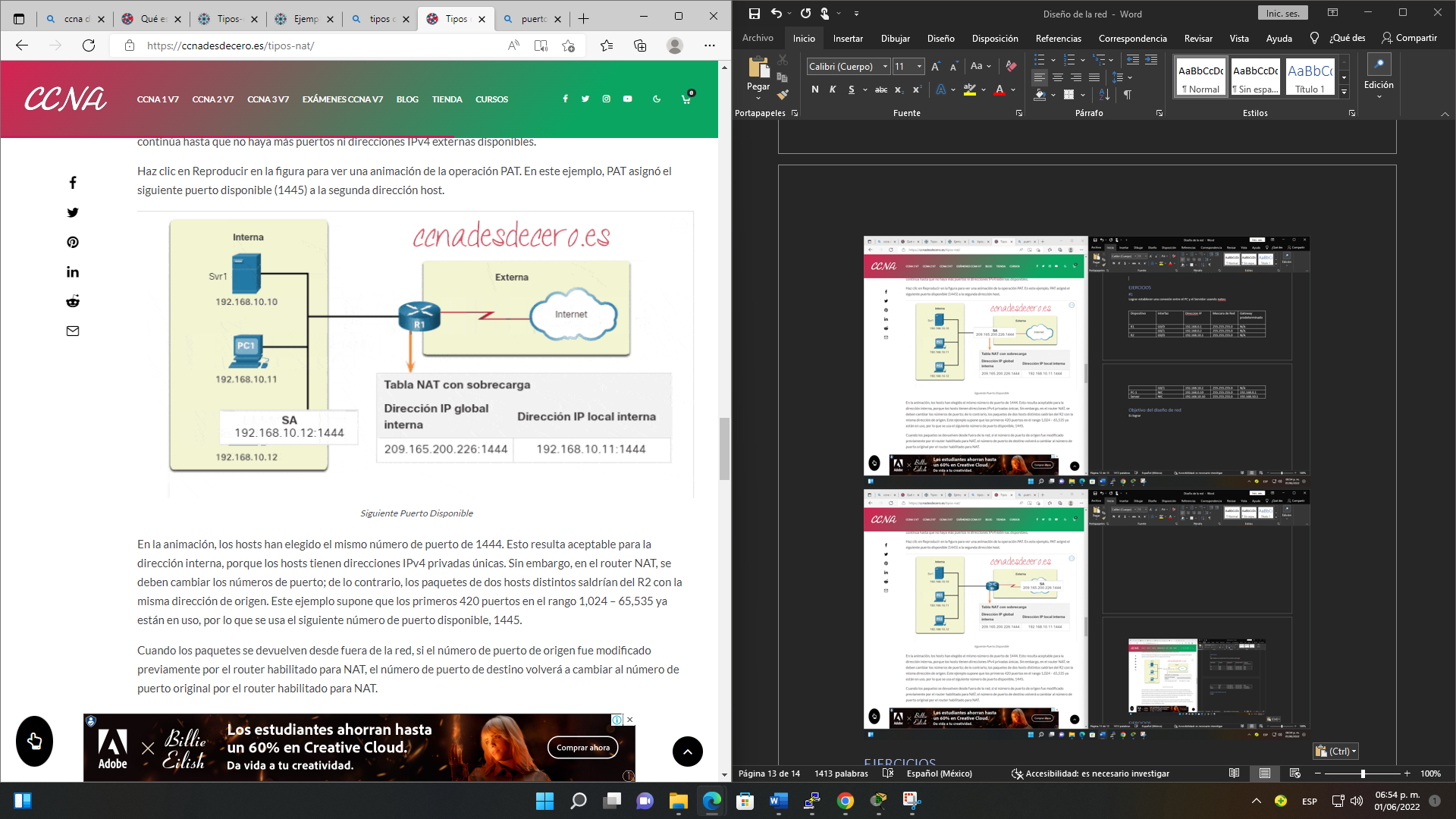
En el ejemplo anterior, los números de puerto del cliente, 1331 y 1555, no se modificaron en el router con NAT habilitada. Esta no es una situación muy probable, porque existe una gran posibilidad de que estos números de puerto ya se hayan conectado a otras sesiones activas.

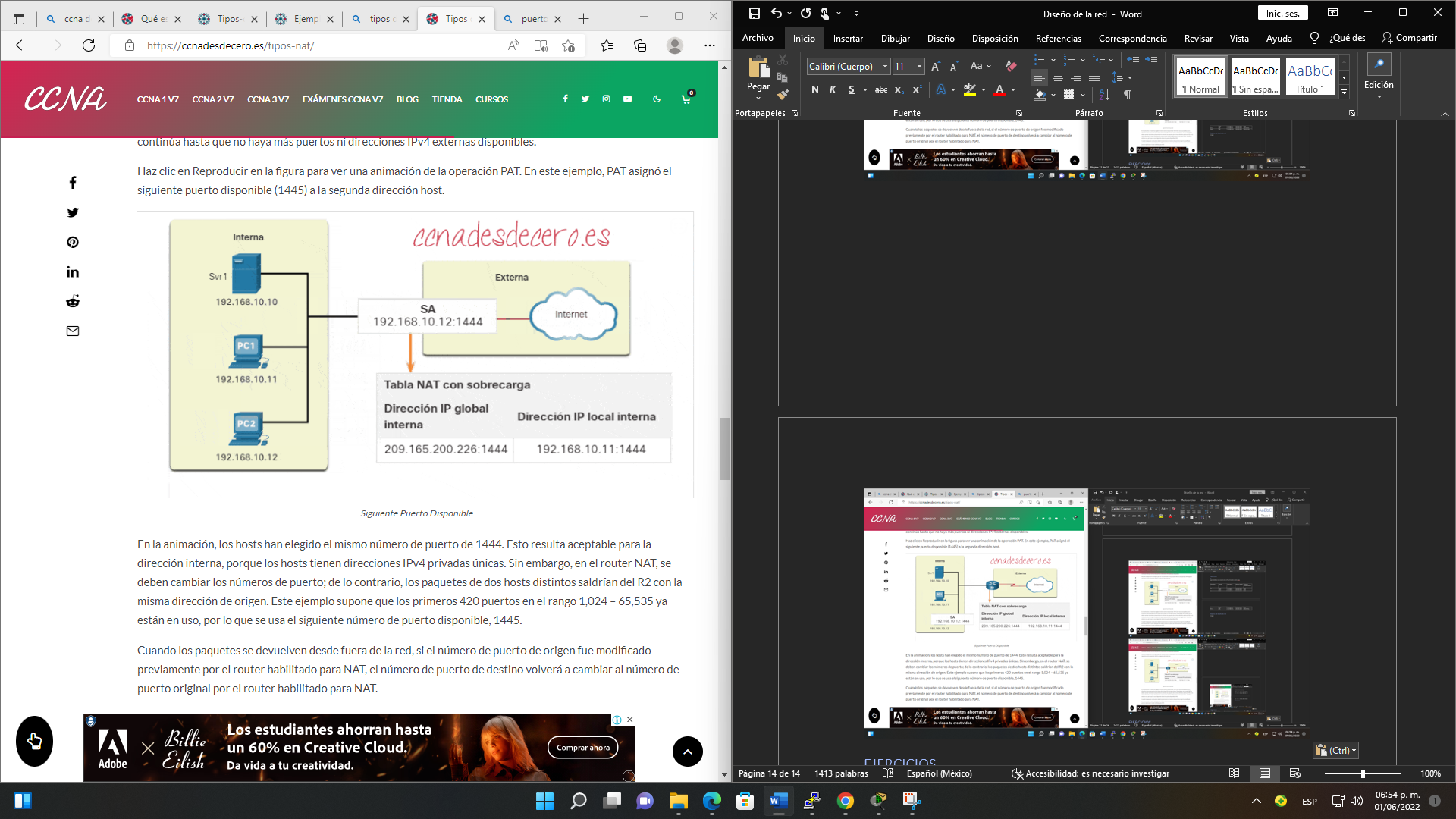
PAT intenta conservar el puerto de origen inicial. Sin embargo, si el puerto de origen original ya está en uso, PAT asigna el primer número de puerto disponible a partir del comienzo del grupo de puertos apropiado 0-511, 512-1,023 o 1,024-65,535. Cuando no hay más puertos disponibles y hay más de una dirección externa en el pool de direcciones, PAT avanza a la siguiente dirección para intentar asignar el puerto de origen inicial. Este proceso continúa hasta que no haya más puertos ni direcciones IPv4 externas disponibles.

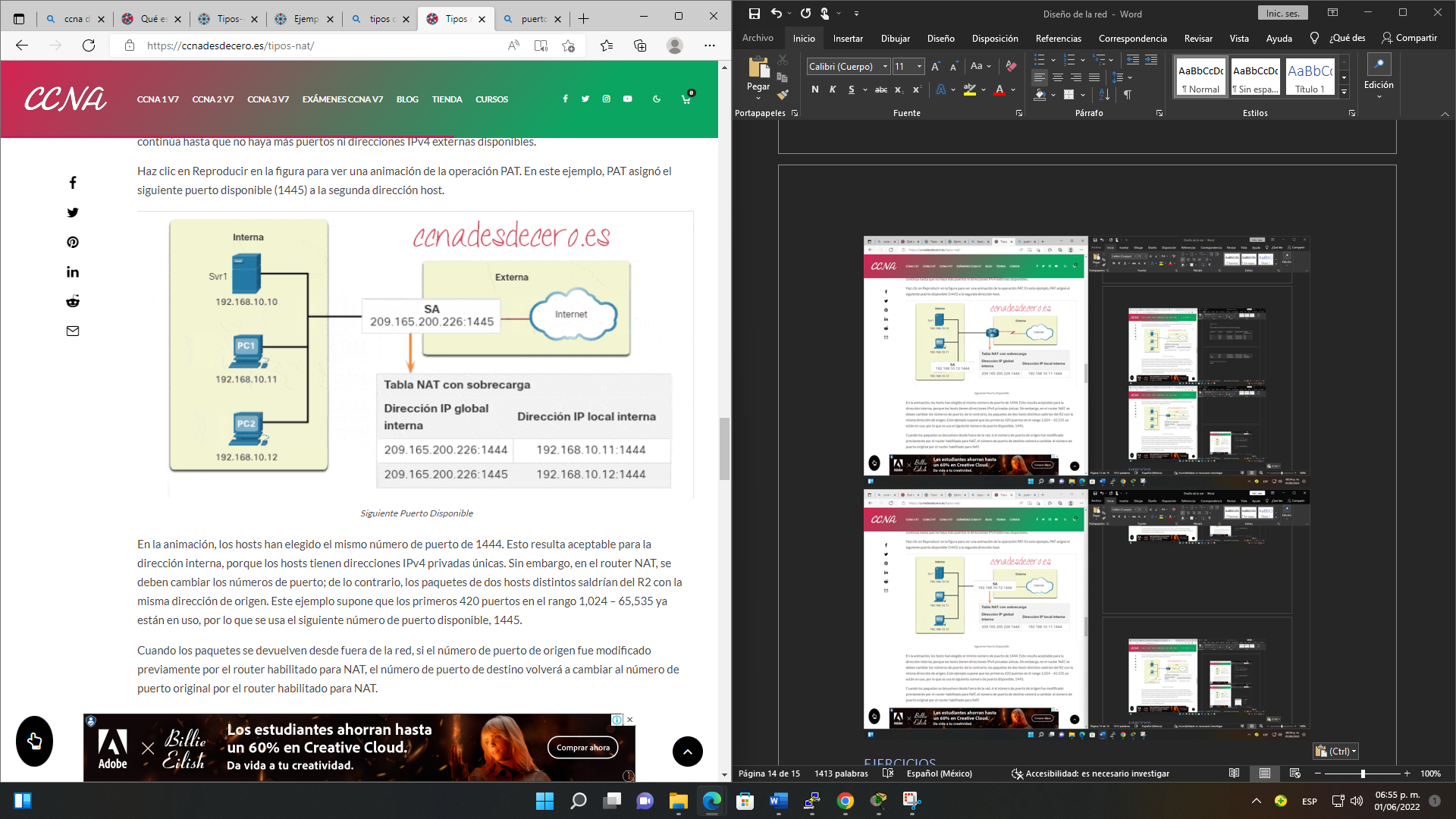


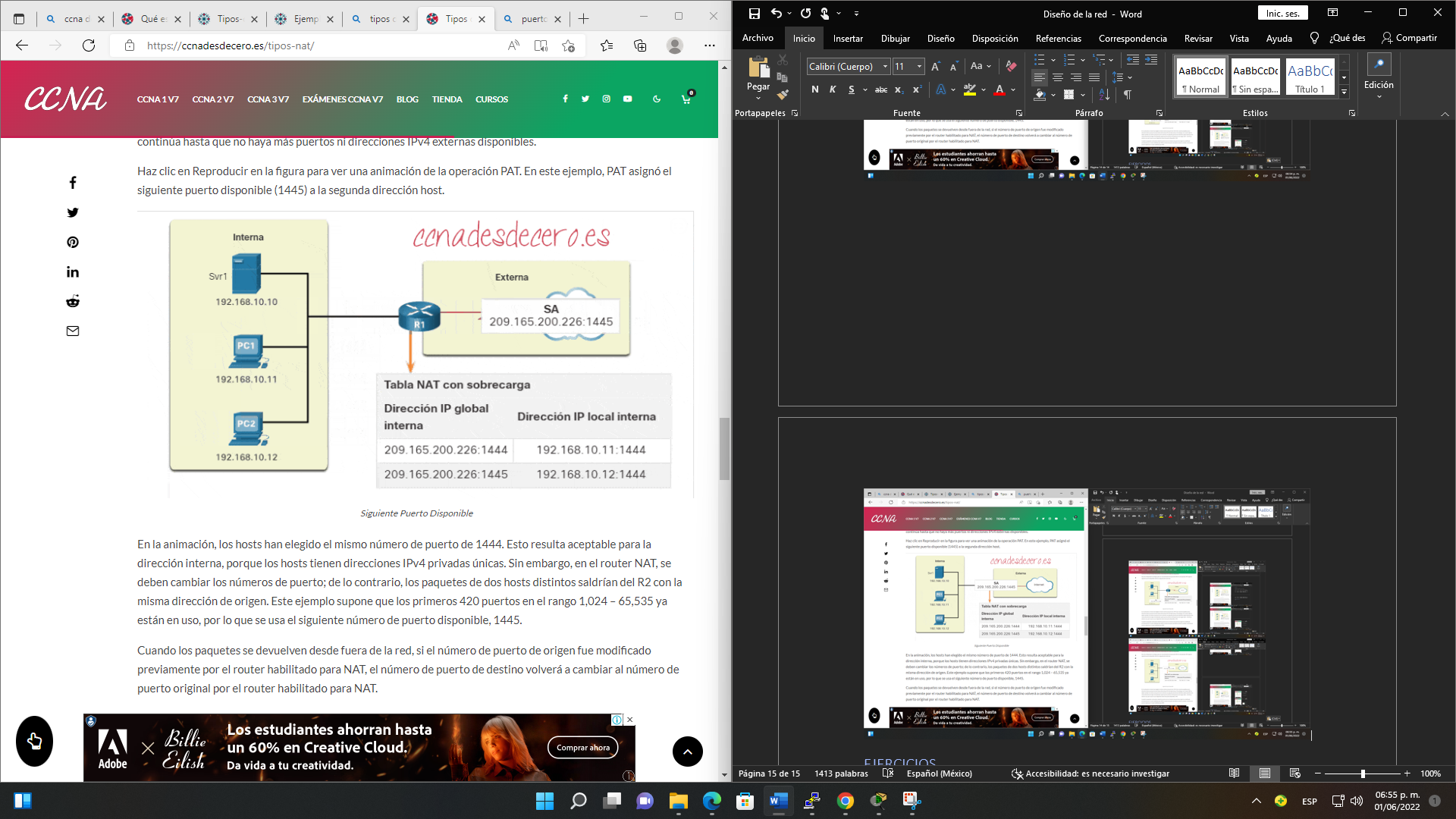












los hosts han elegido el mismo número de puerto de 1444. Esto resulta aceptable para la dirección interna, porque los hosts tienen direcciones IPv4 privadas únicas. Sin embargo, en el router NAT, se deben cambiar los números de puerto; de lo contrario, los paquetes de dos hosts distintos saldrían del R2 con la misma dirección de origen. Este ejemplo supone que los primeros 420 puertos en el rango 1,024 – 65,535 ya están en uso, por lo que se usa el siguiente número de puerto disponible, 1445.

Cuando los paquetes se devuelven desde fuera de la red, si el número de puerto de origen fue modificado previamente por el router habilitado para NAT, el número de puerto de destino volverá a cambiar al número de puerto original por el router habilitado para NAT.

<https://ccnadesdecero.es/tipos-nat/>

## Teléfono VoIP

Un teléfono VoIP, conocido como SIP phone o Softphone, es un teléfono basado en hardware o software diseñado para usar la tecnología de voz sobre IP (VoIP) para enviar y recibir llamadas telefónicas a través de una red IP. El teléfono convierte el audio de la telefonía analógica en un formato digital para transmitirse a través de Internet y convierte las señales entrantes del teléfono digital de Internet en audio telefónico estándar.

Un Teléfono Voip tiene unas determinadas funciones que lo caracterizan como son:

* Identificador de llamadas,
* Trasferencia de llamadas,
* Llamada en espera,
* Acceso a la agenda y la configuración de múltiples cuentas.

## ¿Qué es una base de datos?

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos.

<https://www.oracle.com/mx/database/what-is-database/>

## ¿Qué es un MariaDB?

MariaDB es un sistema de gestión de bases de datos que está muy relacionado con MySQL, El sistema de gestión de bases de datos MariaDB incorpora las distintas funciones características de MySQL añadiendo algunas mejoras, como la posibilidad de ejecutar consultas complejas y almacenarlas directamente en caché, la nueva gestión de conexiones a BD, la posibilidad de acceder a cluster de datos (interesante para el trabajo en la nube) o el soportar la utilización de jerarquías de graphs y estructuras más complejas.

En cuanto a seguridad y rendimiento, MariaDB incorpora mejoras, estando siempre en constante evolución gracias a la aportación de una gran comunidad que se encuentra tras de ella.

<https://www.hostingplus.mx/blog/que-es-mariadb-y-cuales-son-sus-caracteristicas/>

## ¿Qué es DBeaver?

DBeaver es una herramienta gratuita y multiplataforma para trabajar con las bases de datos. Soporta todas las bases de datos que puedas imaginar como MySQL, PostgreSQL, Oracle, MS SQL Server, MS Access, SQLite y muchas otras menos populares. Su potencia y cantidad de características la hacen ideal para desarrolladores, pero también para los administradores de bases de datos más exigentes. DBeaver tiene una serie de funcionalidades bastante completas. Algunas de las siguientes serían las que podemos obtener típicamente en los sistemas de administración de bases de datos.

* Conexión con bases de datos locales y bases de datos remotas a través de túneles SSH.
* Acceso a los datos con posibilidad de filtrados y ordenación.
* Modificación de los datos.
* Editor de SQL.
* Importación y exportación de las bases de datos.

## Descripción del driver ODBC.

En base de datos tenemos muchas opciones como lo son “MySQL”, “MariaDB”, “PostgreSQL”, “Microsoft SQL Server” entre otros. Cada base de datos tiene su propio archivo de configuración, lo cual no es muy conveniente, si el cliente quisiera migrar de una base de datos se tendrían que reconfigurar gran parte del Asterisk para hacer las conexiones, lo cual no es muy práctico, para ello existe un conector llamado “ODBC”, en donde Asterisk solamente se conecta al “OBDC” y este se encarga de hacer las conexiones a las diversas bases de datos.

Para comenzar es necesario una base de datos y una vez hecho eso hay que instalar el “ODBC” y una vez instalado necesitamos instalar el driver de la conexión de ODBC hacia la base de datos.

Luego una vez que el driver de la conexión de ODBC hacia la base de datos necesitamos configurar dos archivos propios, ambos se encuentran en los sigs. Directorios:

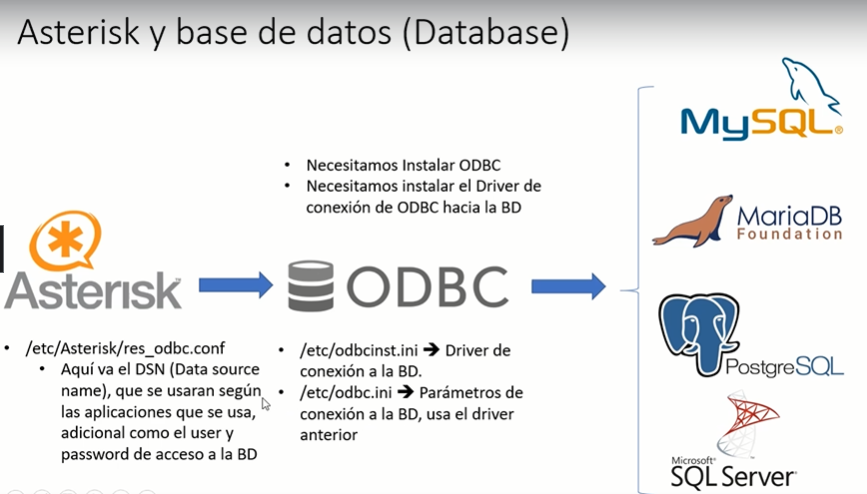
/etc/odbcinst.ini (Driver de conexión a la base de datos).

/etc/odbc.ini (Parámetros de conexión a la base de datos, usa el driver anterior).

Por último, el único archivo que necesitamos para la conexión de Asterisk “ODBC” hacia la base de datos sería el sig.:

/etc/asterisk/res\_odbc.conf

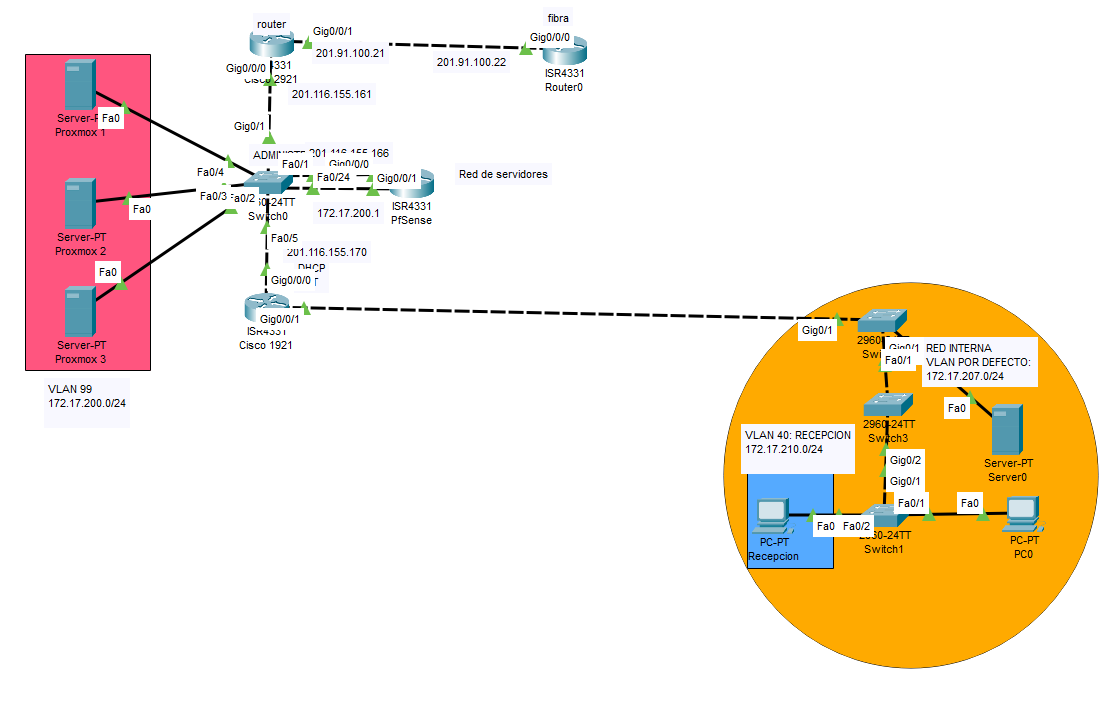
Que será donde se configura la conexión de Asterisk indicándole que va a usar el “ODBC”. Donde se coloca el DSN (Data source name) que se usaran según las aplicaciones que se usa, adicional como el “user” y el “password”.



# DESARROLLO DE LA ESTADIA:

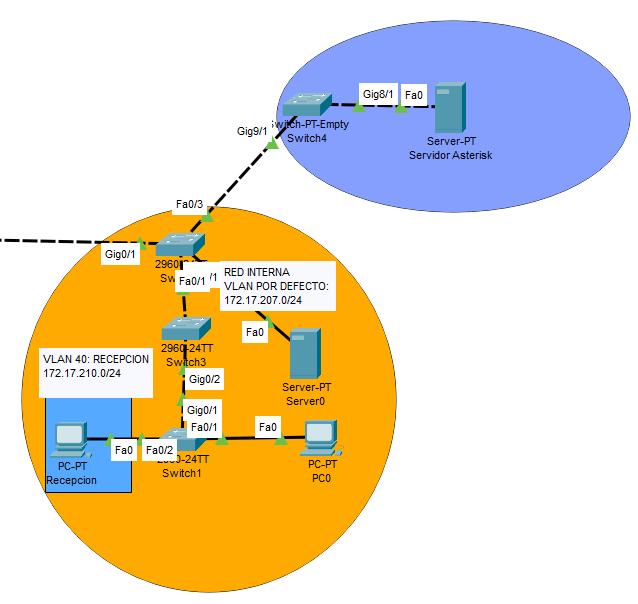
## Instalación del servidor

## Diseño de red.



En la imagen que pueden observar se trata de la topología de red de la empresa DiagnoCons, la cual está distribuida en dos áreas, una parte es la red interna y la otra es el área de servidores. En su red usan lo que es NAT con sobrecarga o PAT, donde básicamente tiene una IP publica y dentro de ella tiene su propia red privada, es decir, para que la red privada salga al exterior usa el proceso de PAT, donde la IP privada con su puerto, llegan a lo que se le llama router frontera el cual transforma la IP privada a una IP publica usando puertos a la vez y así tener una mejor conexión y seguridad. Nuestro objetivo es integrar nuestro servidor Asterisk a la red para lo cual antes de llevarlo en marcha realizaremos pruebas en Cisco Packet Tracer para que su integración a la red sea todo un éxito.

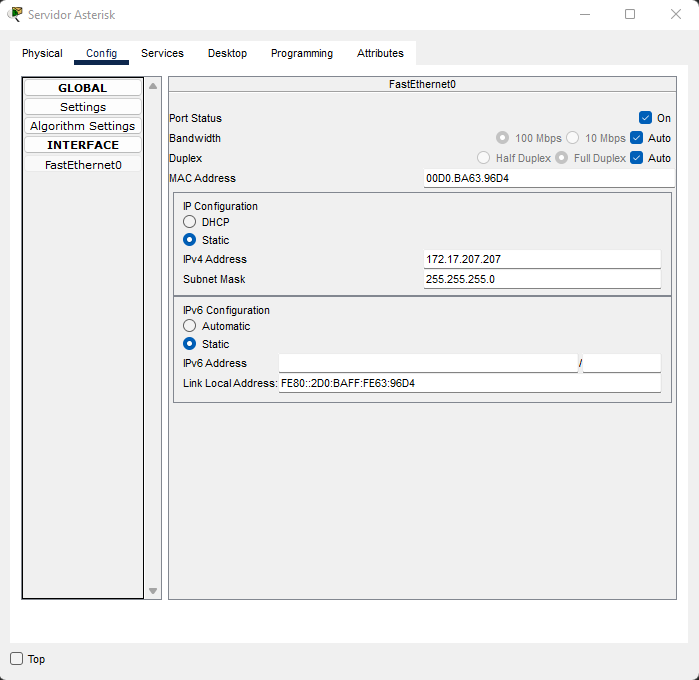
Lo primero que haremos será agregar un switch y un servidor.



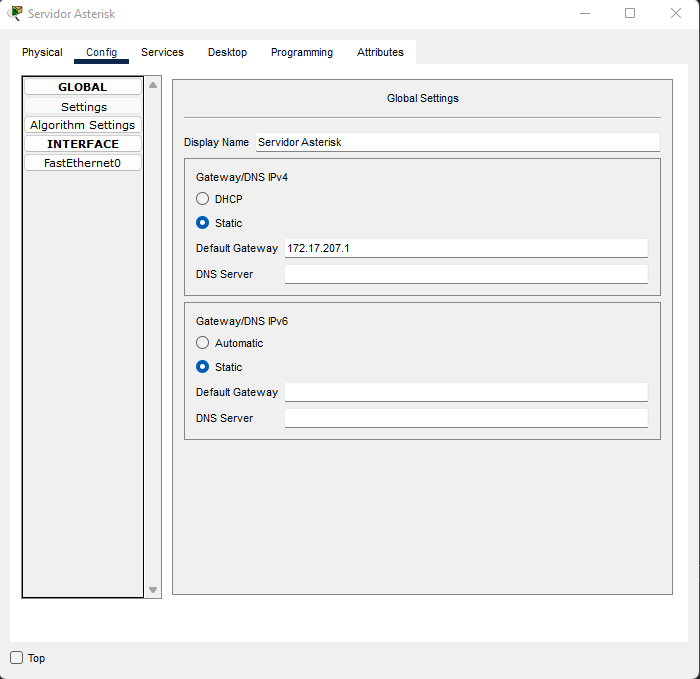
Como se puede observar el servidor mediante un puerto Fastethernet 0 (Fa0) se conecta a un switch en el puerto GigabitEthernet 8/1 (8/1) el cual a su vez se conecta a otro switch que se conecta al router que se comunica con el área de servidores principales.

Después entraremos a la configuración y le asignamos su IP y su Gateway.

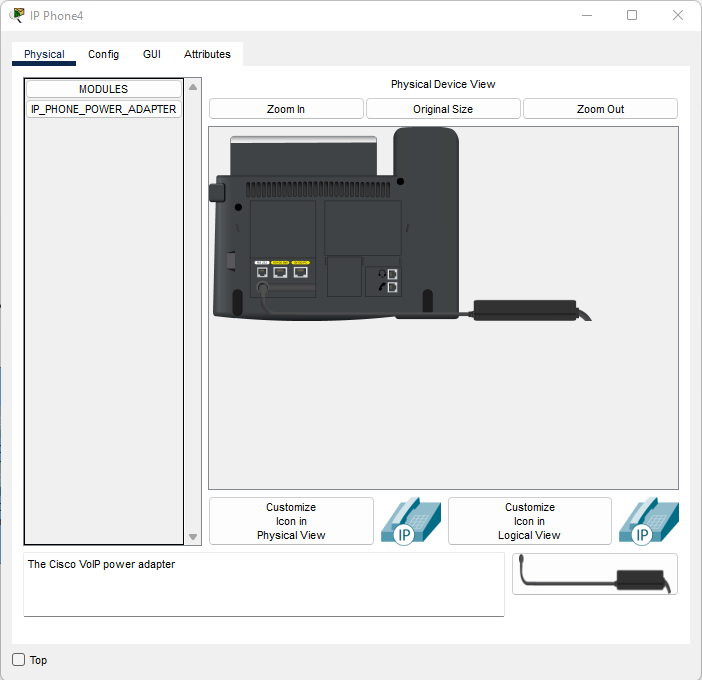
* IP: 172.17.207.207
* Mascara de red: 255.255.255.0



* Gateway: 172.17.207.1



Después solo sería cuestión de integrar los teléfonos VoIP y mediante el servidor Asterisk configurarlos para su funcionamiento (esto último es aparte ya que mediante Cisco Packet Tracer no se puede hacer, en este diseño de red solo se mostraría un ejemplo de cómo quedaría su estructura) y solamente los activamos.





## Configuración del servidor

### Base de datos

#### Instalación de ODBC Conector para MariaDB.

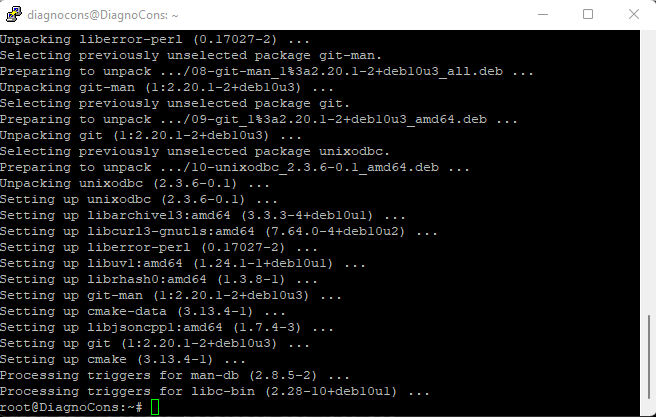
Lo primero que tenemos que hacer es instalar el ODBC y luego instalar el conector. El driver cambia de nombre de nombre dependiendo la distribución y dependiendo de la base de datos que conecta, en este caso que estamos usando Debian y MariaDB se llamara “libmaodbc.so”.

#### Instalación de ODBC y paquetes previos.

Lo primero será descargar el primero paquete con este comando:

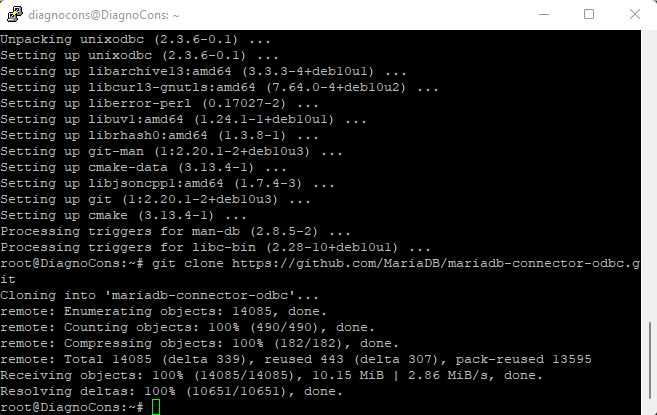
apt-get install unixodbc git cmake

* El git lo usaremos para descargar.
* EL cmake para instalar.



Después nos dirigimos al directorio “/usr/src/” después desargamos el paquete de git con el sig. comando:

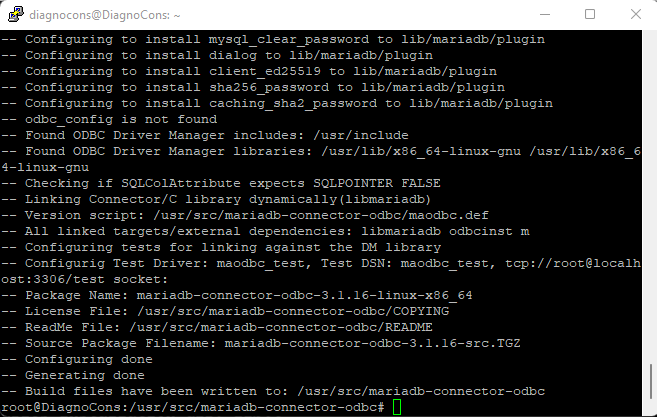
git clone https://github.com/MariaDB/mariadb-connector-odbc.git



Listo una vez descargado el git, entramos a la carpeta “mariadb-connector-odbc” y hacemos la compilación para lo cual insertaremos los sigs. comandos:

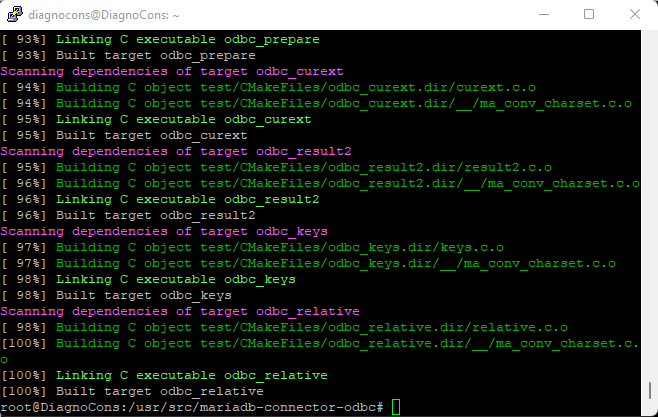
cd mariadb-connector-odbc

cmake -DCMAKE\_BUILD\_TYPE=RelWithDebInfo -DCONC\_WITH\_UNIT\_TESTS=Off -DCONC\_WITH\_MSI=OFF -DCMAKE\_INSTALL\_PREFIX=/usr/local .



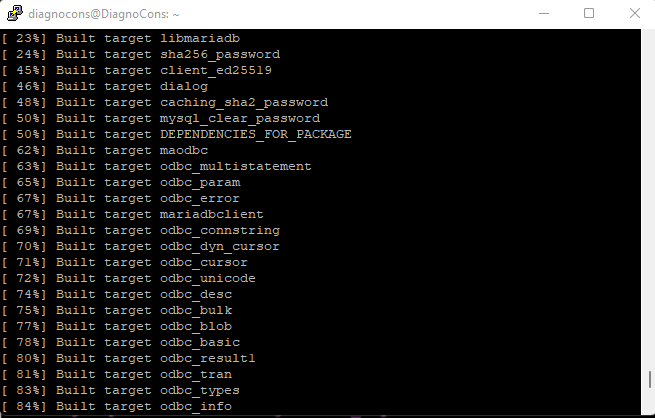
Luego insertamos el sig. comando:

cmake --build . --config RelWithDebInfo

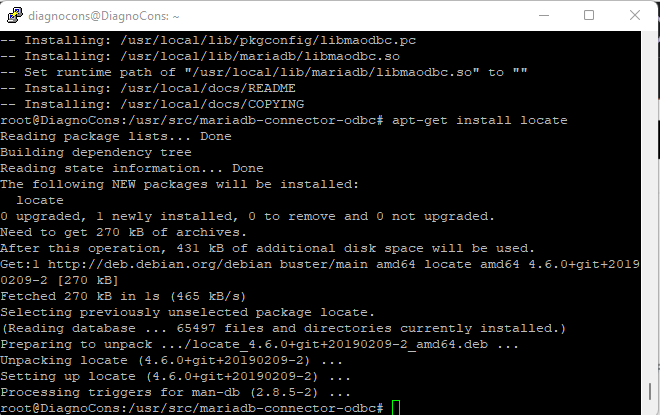


Por último, insertamos el sig. comando:

make install



Linux posee una herramienta llamada “locate” que nos sirve para buscar archivos, esta la usaremos para ubicar el archivo “libmaodbc.so”, para instalarla usamos el sig. comando:

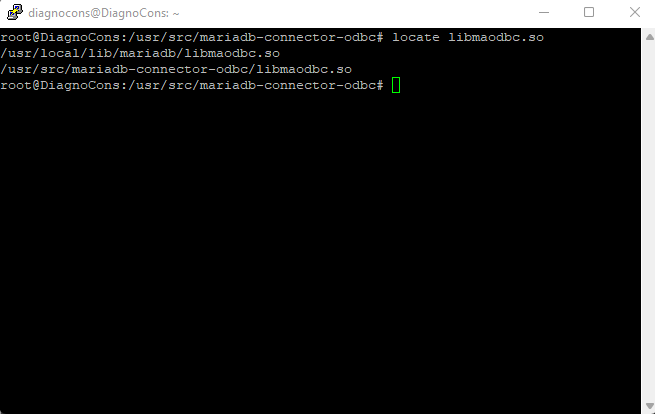


Una vez instalado actualizamos su base de datos para ello usaremos el sig. comando:

updatedb

Luego para buscar el archivo usamos el sig. comando

locate libmaodbc.so



Aparecen dos opciones, pero la que nos interesa es la primera que es donde esta ubicado, el otro es de donde se ha descargado.

/usr/local/lib/mariadb/libmaodbc.so

#### Configuración de odbcinst.ini, odbc.ini y res\_odbc.conf.

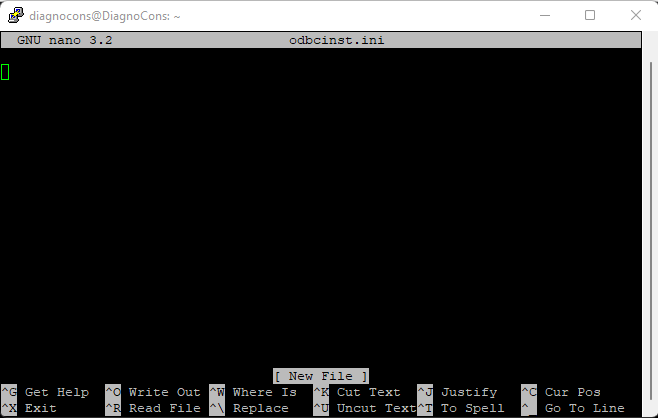
Recordemos:

* odbcinst.ini: driver de la conexión a la BD.
* odbc.ini: parámetros de conexión a al BD y usa el anterior.

##### odbcinst.ini

Lo primero será ubicarnos en el directorio “/etc/”, ahora el archivo “odbcinst.ini” no existe como tal, por lo cual tendremos que crearlo, para ello usaremos el sig. comando:

nano odbcinst.ini

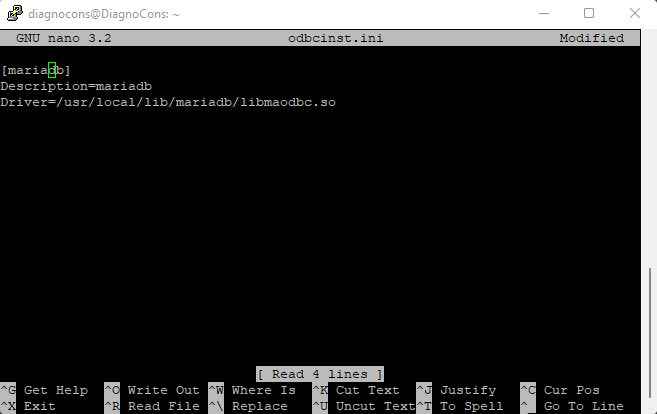


Lo primero será crear el identificador de nuestro conector, para lo cual colocaremos lo sig.:

[mariadb] (ID con el que lo vamos a identificar)

Description=mariadb (es opcional la descripción)

Driver=/usr/local/lib/mariadb/libmaodbc.so (Aquí colocamos donde se encuentra ubicado el driver)



##### odbc.ini

Después crearemos el “odbc.ini” donde vendrían los parámetros de conexión, para crearlo usaremos el sig. comando:

nano odbc.ini



Después le agregaremos lo sig.:

[mariadbcliente] (Su ID)

Description=mariadbcliente

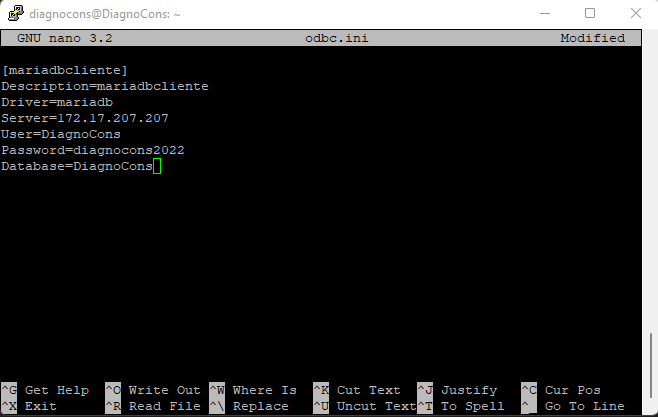
Driver=mariadb (Se coloca el ID que le colocamos al ID de “odbcinst.ini”)

Server=172.17.207.207 (Se coloca la IP del servidor)

User=DiagnoCons (El usuario que creamos para la base de datos)

Password=diagnocons2022 (La contraseña para la base de datos)

Database=demo (Nombre de la base de datos que creamos en dbeaver)

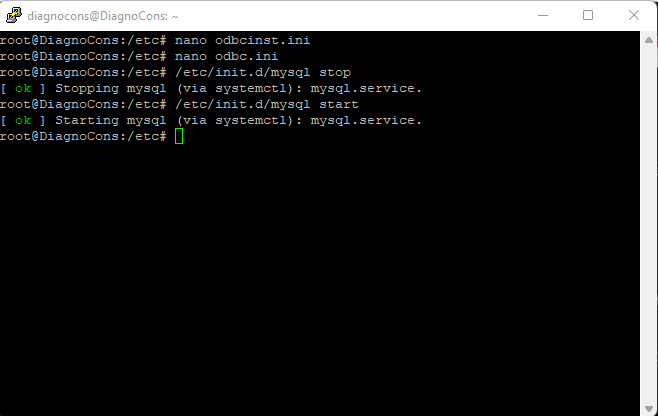


Una vez hecho todo esto salvamos.

Después reiniciamos la base de datos, para ello usamos los sigs. comandos:

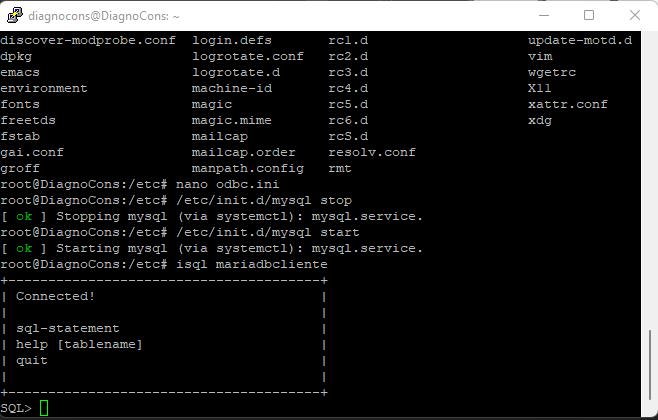
/etc/init.d/mysql stop

/etc/init.d/mysql start



Para saber si hay conexión usamos el sig. comando:

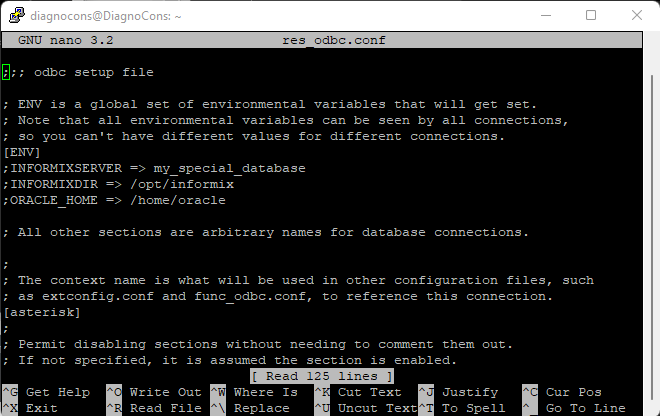
isql mariadbcliente



Nota: si esta correcta la información mostrara este cuadro.

Para que haya conexión del Asterisk hacia la base de datos hay que configurar el archivo “res\_odbc.conf” pero siempre manteniendo el orden, para ingresar a el nos ubicaremos el el directorio “/etc/asterisk” y después usaremos el sig. comando:

nano res\_odbc.conf



Como siempre lo moveremos a la carpeta bak usando el sig. comando:

mv res\_odbc.conf res\_odbc.conf.bak

Después volvemos a entrar a “res\_odbc.conf” y le vamos a crear el DSN.

[asterisk] (Le creamos su ID)

enable=yes

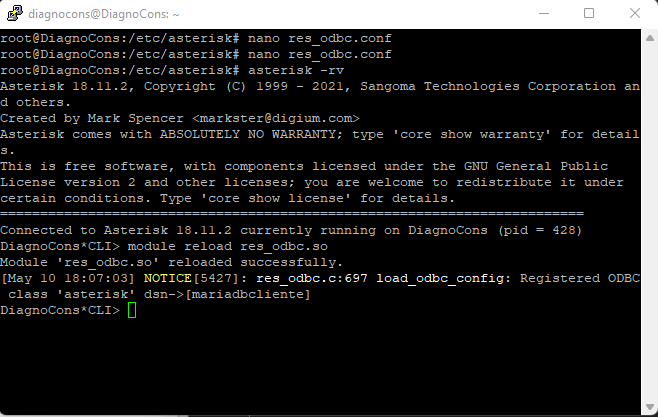
dsn=mariadbcliente (ID del “odbc.ini”)

pre-connect=yes (Para que se establezca la conexión)



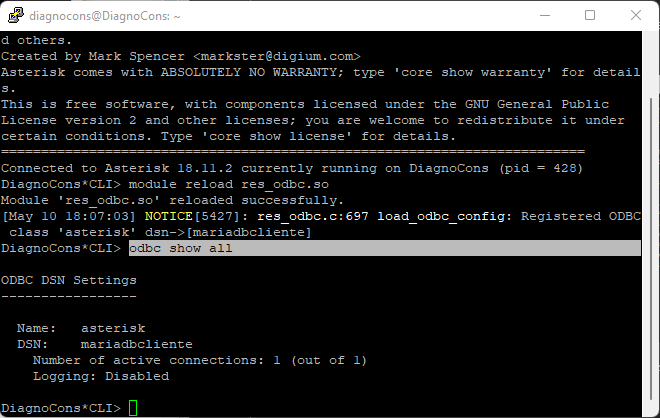
Después entramos en la consola Asterisk y recargamos el modulo con el sig. comando:

module reload res\_odbc.so



Para saber si tiene conexión la consola usamos el sig. comando:

odbc show all



Nota: nos mostrara un mensaje de que la conexión está hecha.

Con esto el Asterisk está listo para conectar con la base de datos.

#### Descripción de func\_odbc.conf para ejecutar Querys en el Dialplan.

Ahora veremos cómo le hacemos interactuar al dialplan con la base de datos. Lo que haremos para interactuar seria leer valores, insertar valores o actualizar valores en l a base de datos. Recordemos el archivo del dialplan seria “extensions.conf”. Para interactuar el dial plan con la base de datos usaremos el archivo “func\_odbc.conf” ubicado en el directorio “/etc/asterisk/”.

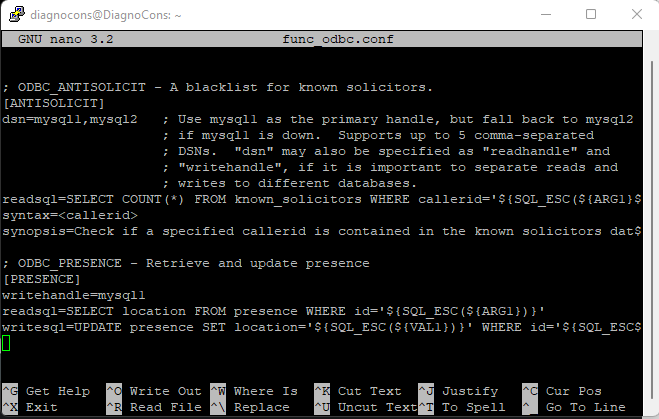
Para verlo usaremos el sig. comando:

nano func\_odbc.conf



En la parte final vemos que van los querys, los cuales son 3 principalmente:

* INSER: cuando queremos insertar registros en la base de datos.
* SELECT: cuando queremos obtener datos de la base de datos.
* UPDATE: cuando queremos actualizar los datos de la base de datos.



Primero vamos a conocer las sintaxis del archivo:

nombreFuncion

prefix=PRE (El prefijo es opcional, por defecto el prefijo es ODBC)

dsn=asterisk (Conexión de Asterisk al ODBC)

readsql= SELECT

writesql= INSERT,UPDATE

Cuando hagamos los querys tenemos dos tipos de datos (que en realidad son los mismo) que son:

* Argumentos (Se usan para cosas que no son valores como las condiciones)
* Valores (Se usa para cantidades que se van a actualizar)

En el dialplan se colocaría en “extensions.conf” lo sig.:

prefix\_nombreFuncion, OBDC\_nombreFuncion, PRE\_nombreFuncion

Después como siempre lo moveremos al bak con el sig. comando:

mv func\_odbc.conf func\_odbc.conf.bak

Volvemos a insertar el comando:   
 nano func\_odbc.conf

## Plan de marcado

### Planeación de plan de marcado

### Configuración del plan de marcado

## Integración a base de datos RIS

## Pruebas.

# ANALISIS DE LOS RESULTADOS:

# REFERENCIA BIBLIOGRAFICA:

# APÉNDICE: