

GRADO EN INGENIERÍA EN TECNOLOGÍA DE  
TELECOMUNICACIÓN

**TRABAJO FIN DE GRADO**

***SISTEMA DE MONITORIZACIÓN Y CONTROL DE  
CONEXIONES ENTRE SISTEMAS AUTÓNOMOS***

**Alumno:** Gangoiti, Bilbao, Danel

**Directora:** Higuero, Aperribay, María Victoria

**Curso:** 2018-2019

**Fecha:** Bilbao, 18 de junio de 2019

## RESUMEN

En las últimas décadas ha aumentado el uso de Internet drásticamente y en paralelo también las relaciones entre Sistemas Autónomos, que son las entidades que constituyen Internet. Sin embargo, a día de hoy los Sistemas Autónomos tienen tantos enlaces, que el tráfico que se envía por ellos es casi imposible de controlar. En este proyecto se estudia la situación actual del entorno tecnológico y se propone un diseño, implementación y despliegue software como solución. En este Trabajo Fin de Grado se desarrolla un sistema que monitoriza y supervisa el tráfico que se intercambia entre Sistemas Autónomos que tienen una relación de tipo peering. Para ello, el sistema deberá controlar dicho tráfico en tiempo real y de forma automática y continua, sin la intervención de ningún ser humano. Además, también será capaz de detectar errores de configuración tanto en el sistema interno de la organización como en organizaciones externas, proponiendo una respuesta rápida como corrección.

## LABURPENA

Azken hamarkadetan, Interneten erabilera nabarmen handitu da, eta era berean, Sistema Autonomoen, Internet osatzen duten entitateen, arteko harremanak. Hori dela eta, gaur egungo enpresek elkar trukaturako trafikoa kontrolatzea ia ezinezkoa da, lotura kantitate handiaren ondorioz. Proiektu honetan gaur egungo teknologiaren egoera aztertzen da eta irtenbide gisa software baten diseinua, inplementazioa eta hedapena proposatzen dira. Gradu Amaierako Lan honetan peering motatako harremana duten Sistema Autonomoek trukatzeko duten trafikoa monitorizatu eta gainbegiratzeko duen sistema garatzen da. Horretarako sistemak trafiko hau denbora errealean eta modu automatiko eta etengabe kontrolatu beharko du, gizakiaren parte hartzerik gabe. Gainera, konfigurazio akatsak antzemateko gai izango da, bai erakundearen barne-sisteman, zein kanpo erakundeetan, zuzenketa moduan erantzun azkar bat proposatuz.

## ABSTRACT

In recent decades the use of the Internet has increased significantly and in parallel also the relationships between Autonomous Systems, which are the entities that constitute Internet. However, nowadays companies have so many links that the traffic sent by them is almost impossible to control. In this project the current situation of the technological world is studied and a design, implementation and software deployment is proposed as a solution. In this End of Degree Project, a system that monitors and supervises the traffic that is exchanged between Autonomous Systems that have peering type relationships is developed. For that, the system must control such traffic in real time, automatically and continuously, without the intervention of any human being. In addition, it will also be able to detect configuration errors, both in the internal system of the organization and in external organizations, proposing a quick response as a correction.

# ÍNDICE

1	INTRODUCCIÓN.....	9
2	CONTEXTO.....	10
3	OBJETIVOS Y ALCANCE .....	12
4	BENEFICIOS.....	13
4.1	Técnicos.....	13
4.2	Económicos.....	13
4.3	Sociales.....	13
5	ANÁLISIS DE ALTERNATIVAS.....	14
5.1	Distribución de la inteligencia .....	14
5.1.1	Inteligencia centralizada .....	14
5.1.2	Inteligencia distribuida .....	14
5.1.3	Selección de alternativas.....	14
5.2	Comunicación con base de datos RIPE.....	15
5.2.1	Interfaz web.....	15
5.2.2	Herramienta whois.....	16
5.2.3	RESTful API .....	16
5.2.4	Selección de alternativas.....	17
5.3	Desarrollo del sistema completo.....	17
5.3.1	Herramienta bgpq3.....	17
5.3.2	Solución propia.....	18
5.3.3	Selección de alternativas.....	18
5.4	Desarrollo del módulo Optimizador de prefijos.....	18
5.4.1	Módulo pyipcalc.....	18
5.4.2	Desarrollo propio .....	18
5.4.3	Selección de alternativas.....	19
6	DESCRIPCIÓN DE LA SOLUCIÓN .....	20
6.1	Visión general de la arquitectura.....	20
6.2	Diseño de la solución.....	20
6.2.1	Base de datos.....	21
6.2.2	Transmisor de alarmas .....	25
6.2.3	CORE .....	25
6.2.4	Buscador .....	26
6.2.5	Optimizador de prefijos.....	31
6.2.6	Comparador .....	31
6.2.7	Comunicación.....	32

6.2.8	Optimizador router .....	35
6.2.9	Corrector de errores .....	38
6.2.10	Temporización .....	40
6.3	Funcionamiento del sistema .....	41
7	METODOLOGÍA.....	42
7.1	Descripción de tareas .....	42
7.2	Planificación .....	42
7.3	Diagrama de Gantt .....	45
8	RESUMEN DE COSTES .....	46
8.1	Horas internas .....	46
8.2	Amortizaciones .....	46
8.3	Gastos .....	47
8.4	Resumen de costes .....	47
9	ANÁLISIS DE RIESGOS .....	49
9.1	Identificación de riesgos .....	49
9.1.1	Incompatibilidad parcial o completa (R1) .....	49
9.1.2	Desviaciones en la planificación (R2) .....	49
9.1.3	Desvíos en el presupuesto (R3).....	49
9.1.4	Fallos en el equipamiento (R4) .....	49
9.2	Análisis de riesgos.....	50
9.3	Planificación de la respuesta .....	51
10	CONCLUSIONES .....	52
11	BIBLIOGRAFÍA .....	53
ANEXO I: FUNDAMENTOS TEÓRICOS .....		55
1	Estructura de Internet y Sistemas Autónomos.....	55
1.1	Tipos de Sistemas Autónomos.....	56
2	Protocolo BGP.....	56
2.1	Tipos de mensajes .....	57
2.2	Estados.....	57
2.3	Sesiones BGP .....	58
2.4	Atributos .....	58
2.5	Toma de decisiones.....	59
3	Peering.....	59
3.1	Tipos de peering .....	61
4	Internet Assigned Numbers Authority .....	61
4.1	RIPE NCC.....	62
4.2	Objeto route .....	63

ANEXO II: HERRAMIENTAS .....	65
1 Herramienta bgpq3.....	65
2 Herramienta Ansible .....	66
2.2 Requerimientos del sistema .....	66
2.2 Características .....	66
2.3 Conceptos.....	67
3 NETCONF .....	68
3.1 Características .....	68
3.2 Sesión.....	69
3.3 Comandos .....	69

# LISTA DE FIGURAS

Figura 1: ASs, IBGP y EBGP .....	9
Figura 2: Comunicación entre Sistemas Autónomos.....	10
Figura 3: Búsqueda interfaz web .....	15
Figura 4: Resultados interfaz web.....	15
Figura 5: Herramienta whois .....	16
Figura 6: Petición RESTful API .....	16
Figura 7: Arquitectura general.....	20
Figura 8: Módulos del sistema .....	21
Figura 9: Bases de datos .....	22
Figura 10: Consulta a múltiples bases de datos.....	22
Figura 11: Consulta a objeto local ISP .....	23
Figura 12: Consulta a objeto remoto ISP .....	23
Figura 13: Fragmento de lista de vecinos obtenida mediante el router .....	24
Figura 14: Notificación .....	25
Figura 15: Consulta AS-set.....	26
Figura 16: Consulta members de un AS-set .....	26
Figura 17: Members de as-set .....	27
Figura 18: Consulta de objeto route con búsqueda inversa de atributo origen .....	28
Figura 19: Prefijos de un AS .....	29
Figura 20: Consulta de objeto route6 con búsqueda inversa de atributo origen .....	30
Figura 21: Estructura de archivo YAML .....	32
Figura 22: Plantilla Jinja2.....	33
Figura 23: Lista de prefijos IPv4 e IPv6.....	34
Figura 24: Política de filtrado IPv4 .....	34
Figura 25: Política de filtrado IPv6 .....	34
Figura 26: Filtro aplicado en vecino IPv4 .....	35
Figura 27: Filtro aplicado en vecino IPv6 .....	35
Figura 28: Bogon ASN filter .....	36
Figura 29: Bogon Prefix filter.....	37
Figura 30: Long AS Paths filter .....	38
Figura 31: NuevoAS.py.....	39
Figura 32: NuevoAS.py -s.....	40
Figura 33: Funcionamiento del sistema .....	41
Figura 34: Diagrama de Gantt.....	45
Figura 35: Comparativa de partidas del coste total del proyecto .....	48
Figura 36: Matriz probabilidad-impacto .....	50
Figura 37: Registros Regionales de Internet .....	56
Figura 38: Estados de BGP .....	57
Figura 39: Peering vs tránsito .....	60
Figura 40: NETCONF .....	68

## LISTA DE TABLAS

Tabla 1: Selección de alternativas "Distribución de la inteligencia" .....	14
Tabla 2: Selección de alternativas "Comunicación con base de datos RIPE" .....	17
Tabla 3: Selección de alternativas "Desarrollo del sistema completo" .....	18
Tabla 4: Selección de alternativas "Desarrollo del módulo Optimizador de prefijos" ...	19
Tabla 5: Planificación.....	44
Tabla 6: Partida de horas internas .....	46
Tabla 7: Partida de amortizaciones.....	46
Tabla 8: Partida de gastos .....	47
Tabla 9: Resumen de costes .....	47
Tabla 10: Análisis de riesgos .....	50
Tabla 11: Lista de ASNs .....	55
Tabla 12: Atributos BGP .....	58
Tabla 13: Atributos objeto route .....	64
Tabla 14: Herramienta bgpq3 .....	66
Tabla 15: Comandos NETCONF .....	69

# LISTA DE ACRÓNIMOS

BGP	<i>Border Gateway Protocol</i>
EBGP	<i>Exterior Border Gateway Protocol</i>
IBGP	<i>Interior Border Gateway Protocol</i>
EIGRP	<i>Enhanced Interior Gateway Routing Protocol</i>
AS	<i>Autonomous System</i>
ASN	<i>Autonomous System Number</i>
RIR	<i>Regional Internet Registry</i>
IANA	<i>Internet Assigned Numbers Authority</i>
RADb	<i>Routing Assets Database</i>
ISP	<i>Internet Service Provider</i>
IP	<i>Internet Protocol</i>
RPSL	<i>Routing Policy Specification Language</i>
API	<i>Application Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
HTML	<i>HyperText Markup Language</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
HTTP	<i>Hypertext Transfer Protocol</i>
WWW	<i>World Wide Web</i>
URL	<i>Uniform Resource Localizator</i>
YAML	<i>Aint't Markup Language</i>
XML	<i>Extensible Markup Language</i>
DNS	<i>Domain Name System</i>
IXP	<i>Internet Exchange Point</i>
SSH	<i>Secure Shell</i>
SGML	<i>Standard Generalized Markup Language</i>
RPSS	<i>Routing Policy System Security</i>
DFZ	<i>Default Free Zone</i>
ASCII	<i>American Standard Code for Information Interchange</i>
NETCONF	<i>Network Configuration Protocol</i>
RPC	<i>Remote Procedure Call</i>



# 1 INTRODUCCIÓN

Internet no es únicamente una red de ordenadores, sino una red de redes [1]. Tiene como origen la red ARPANET que fue creada en el año 1969 por encargo del Departamento de Defensa de los Estados Unidos, con el objetivo de interconectar redes universitarias. Desde entonces ha ido evolucionando y creciendo exponencialmente hasta la actualidad.

A día de hoy, Internet conecta a 4300 millones de usuarios en todo el mundo [2]. Está claro que esta enorme infraestructura debe ser administrada y gestionada. Internet está formada por muchas redes pequeñas, llamadas Sistemas Autónomos, que son los que estructuran Internet en dominios de rutado individuales y más pequeños. Este sistema de organización simplifica la administración y gestión y facilita la configuración de políticas. Cada dominio de rutado es un único dominio administrativo independiente y trabaja con políticas diferentes. Se puede decir que el dominio es autónomo y es por eso por lo que se denomina Sistema Autónomo (AS). Un Sistema Autónomo se define como un grupo de redes IP que es gestionado por una entidad administrativa que posee una clara y propia política de rutado [3].

En este escenario, el rutado de internet está estructurado en dos niveles jerárquicos. Por un lado, dentro de los dominios el rutado se establece mediante protocolos de rutado interior (IGP), los cuales tienen conocimiento del dominio. Ejemplos de este tipo de protocolos son Open Shortest Path First (OSPF), Routing Information Protocol (RIP) e Enhanced Interior Gateway Routing Protocol (EIGRP). Por otro lado, existen los protocolos interdominio o de rutado exterior (EGP), los cuales proporcionan la comunicación entre diferentes Sistemas Autónomos. Hoy en día prácticamente solo se utiliza BGP (Border Gateway Protocol) en este tipo de conectividades.

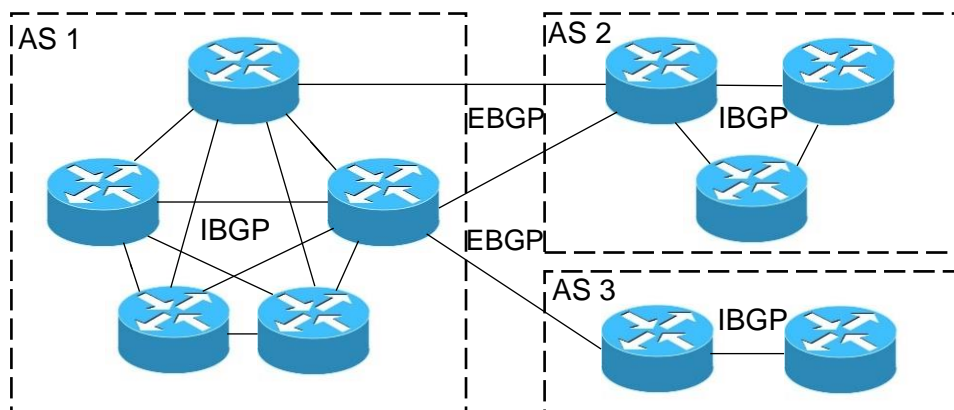


Figura 1: ASs, IBGP y EBGP

BGP es un protocolo simple, de tipo “path vector” y se centra en optimizar la escalabilidad de la red en lugar de otro tipo de métricas, como puede ser el tiempo de convergencia que es un parámetro muy utilizado en otros escenarios y protocolos. Debido al crecimiento de Internet, BGP debe ser capaz de manejar de manera eficaz y eficiente inmensas cantidades de información (direcciones, tablas de rutado, cambios en la red...). Desde 2006 se utiliza la versión 4 de BGP que se encuentra recogida en el RFC 4271 [4].

## 2 CONTEXTO

Hoy en día cualquier persona se conecta a Internet desde casi cualquier dispositivo. Cuando un usuario se conecta a un ordenador o servidor de otra red puede pasar alguna de las siguientes acciones:

- La red destino vende el servicio de acceso directo al proveedor de Internet del usuario (tránsito directo). Es decir, la red del usuario toma el rol de cliente de la red a la que se quiere conectar al usuario.
- La red del usuario compra el servicio de tránsito a una red, donde esta otra red a su vez también ha comprado el acceso directo a la red destino.

Es muy común que existan situaciones en las que dos redes se están comunicando continuamente y están pagando a una tercera red por dicho servicio de tránsito. Además de este coste también se tienen que hacer cargo de la infraestructura física que necesitan para conectarse a la red de tránsito. Por otro lado, existen situaciones donde las dos redes que se quieren comunicar están físicamente en un lugar geográfico cercano como puede ser una zona empresarial, y necesitan intercambiar mucho tráfico de sus usuarios que viajaría a través de los proveedores de tránsito de cada entidad.

Debido a esta situación, se creó un tercer tipo de relación entre ASs llamada peering. El peering es un acuerdo mutuo entre redes de Internet administrativamente independientes con el fin de intercambiar tráfico entre los usuarios de cada red y solo entre ellos, sin proporcionar tránsito a otros destinos. El coste que conlleva esta técnica es mucho menor comparándola con las mencionadas previamente.

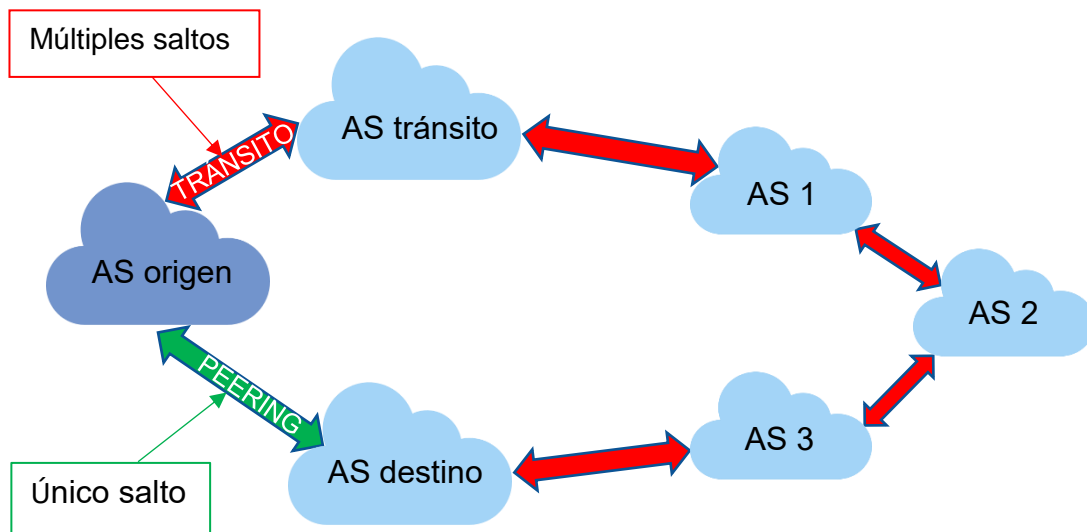


Figura 2: Comunicación entre Sistemas Autónomos

Sin embargo, el peering tiene sus riesgos. Si una empresa tiene un acuerdo de peering con otra, esta podría inyectar a la otra empresa tráfico que no está registrado en el acuerdo. De esta manera, podría utilizar los recursos de la otra parte para evitar consumir capacidad de su enlace a Internet, ya que es muy caro.

Controlar estas situaciones en un principio fue sencillo ya que las empresas solo disponían de unos pocos peerings. Bastaba con tener a un operario controlando el tráfico del enlace peering verificando que no sucediera ninguna irregularidad. Sin

embargo, a medida que ha pasado el tiempo, el número de enlaces peering ha ido aumentando significativamente.

En este contexto, se desarrolla este Trabajo Fin de Grado en colaboración con la empresa Sarnet, con el objetivo de elaborar, diseñar y desarrollar una propuesta a la problemática descrita.

Sarnet nace en 1995 como proveedor de Internet especializado en soluciones corporativas. Desde entonces ha recorrido un largo camino incorporando muchas innovaciones, al igual que el sector de Internet. Hoy es uno de los principales operadores de voz y datos especializado en empresas de España.

Actualmente en Sarnet están operativas cerca de 300 sesiones de peering y consideran importante controlar el tráfico en cada uno de los enlaces. Prácticamente todos los días hay cambios en las redes de los vecinos peering (nuevos rangos, exclusiones de direcciones, etc.). Por otro lado, existen técnicas maliciosas que algunos clientes utilizan para ganar preferencia en la red y así obtener más recursos que los que están pagando al proveedor, en este caso a Sarnet.

Para ello, surge la necesidad de crear una herramienta que monitorice y supervise el adecuado funcionamiento de las relaciones de peering, conforme a lo negociado y establecido, de forma que se esté seguro de que el tráfico que viaja por los enlaces de Sarnet es el que tiene que ser y de esta situación ha surgido la necesidad de este Trabajo Fin de Grado.

### 3 OBJETIVOS Y ALCANCE

El principal objetivo de este proyecto es conseguir un sistema que permita monitorizar y supervisar las conexiones de un Sistema Autónomo con sus Sistemas Autónomos vecinos con los que tiene una relación de peering, de forma que se asegure su adecuado funcionamiento. Además, esta herramienta tiene que funcionar de forma automatizada para que no sea necesaria la intervención de un operario cada vez que se quiera hacer funcionar, sino que teniendo en cuenta que Internet está funcionando 24 horas 365 días al año, estén permanentemente monitorizadas esas conexiones. Si el sistema descubre algún problema tendrá que actuar de la forma que se defina como más conveniente. Este sistema se utilizará como control del tráfico de dichos enlaces y como barrera de seguridad. Todo ello se diseñará de la manera más óptima posible para obtener los mayores beneficios y al menor coste. Para alcanzar este propósito se han definido algunos objetivos parciales:

- **Monitorizar en tiempo real y de forma continua** el tráfico que se envía por estos enlaces. De esta forma se podrán detectar posibles anomalías y proceder a solucionarlas lo antes posible.
- **Enviar notificaciones** de la información obtenida y en caso de encontrar algún error, que las entidades encargadas de los sistemas donde se han encontrado los errores sean informadas de ellos.
- En función del tipo de error encontrado, se modificará la **configuración del sistema** para que funcione adecuadamente.

Además, en este proyecto también se han tenido en cuenta los siguientes objetivos secundarios:

- Como las relaciones de peering están en continuo crecimiento, el sistema se va a desarrollar de manera que sea **escalable**. Se tendrá en cuenta el posible crecimiento de este tipo de relaciones. Si el número de conectividades de este Sistema Autónomo es muy superior, el sistema seguirá siendo válido.
- También se diseñará el sistema de forma **modular** para que si en un futuro próximo se decide recoger información correspondiente a rutas o a direcciones IP de vecinos que tengan otra fuente de información sea sencillo de añadirlo y por otro lado, si se decide tener otro tipo de acciones correctivas también se pueda hacer de forma que el resto del sistema no sufra apenas ningún cambio.
- Además, se buscará que el sistema **no implique una disminución del rendimiento de los routers** ni de la calidad de servicio que ofrecen. Para ello se utilizarán los mínimos recursos posibles cada vez que haya que modificar una configuración.

## **4 BENEFICIOS**

En esta sección se describen algunos de los beneficios que aporta este proyecto desde el punto de vista técnico, económico y social.

### **4.1 Técnicos**

El objetivo de este trabajo es monitorizar y supervisar las conexiones de un Sistema Autónomo con sus Sistemas Autónomos vecinos con los que tiene una relación de peering. Alcanzar este propósito tendría una serie de beneficios en el ámbito técnico y de investigación.

Teniendo en cuenta que las relaciones de tipo peering entre Sistemas Autónomos han crecido de forma exponencial tanto a niveles nacionales como internacionales, es muy importante controlar que funcionan adecuadamente, especialmente para los Sistemas Autónomos que tengan un número elevado de conexiones. Este proyecto contribuirá a asegurar este funcionamiento.

A medida que se desarrollen sistemas como el que se propone en este Trabajo Fin de Grado y que cada parte de Internet funcione adecuadamente, cada vez más ISPs se prestarán a conectarse mediante peering con otros Sistemas Autónomos. Con lo cual Internet seguirá creciendo de una forma adecuada y los proveedores verán beneficios derivados de la mejora de estas conectividades. Es decir, el sistema desarrollado contribuye al mejor funcionamiento de Internet y por tanto a su desarrollo.

### **4.2 Económicos**

El objetivo de cualquier empresa es obtener beneficios económicos. Es por ello fundamental que cada recurso de la organización se esté utilizando exclusivamente para lo que está definido. Mediante este proyecto se va a asegurar que las conectividades de internet se estén utilizando para lo que se han pagado y no para otros fines. Esto conlleva gran parte del coste económico ya que estas conectividades son las más caras. De esta forma, los clientes estarán más contentos por la mejora de las conectividades y esto conlleva más fidelidad por parte de los clientes y así el negocio mejorará.

Por otro lado, es común que en los Sistemas Autónomos no exista ningún control del tráfico de los enlaces peering o sea controlado por uno o más operarios. Mediante esta herramienta se puede automatizar el trabajo de dichos empleados y así, la organización podrá ahorrar los costes de estos.

### **4.3 Sociales**

Uno de los beneficios sociales que puede generar el desarrollo de este proyecto es la mejora de la calidad de servicio en los usuarios finales. Con la herramienta diseñada, los proveedores de Internet sabrán que sus conectividades a Internet se utilizan de forma adecuada y no están siendo utilizadas por otros. Cuanto mejor controladas estén estas conectividades, más peering habrá y las calidades de servicio ofrecidas a los usuarios finales se verán mejoradas por los beneficios que conlleva el peering (véase ANEXO I, apartado 3). Del mismo modo, la posibilidad de ofrecer nuevos servicios aumentará, mejorando así el bienestar y el ocio de los usuarios.

## 5 ANÁLISIS DE ALTERNATIVAS

Durante el desarrollo de cualquier proyecto se deben tomar decisiones que condicionan la solución del mismo. En esta sección se analizan las diferentes alternativas que se han valorado en su diseño y se explica el motivo de su selección.

### 5.1 Distribución de la inteligencia

A la hora de diseñar la arquitectura de red es primordial decidir dónde irá situada la inteligencia del sistema. Esta hace referencia a la parte troncal de la herramienta donde por ejemplo se suele situar el CORE o el núcleo del sistema. Dependiendo de dónde esté situada, el diseño del sistema completo cambiaría por completo. Las alternativas que se han evaluado son las siguientes:

#### 5.1.1 Inteligencia centralizada

Esta alternativa consiste en un diseño centralizado donde la inteligencia de todo el sistema reside en un único punto. Este sería un terminal situado en la propia organización. Para dotar al sistema de inteligencia bastaría con diseñar un módulo CORE que actuara como motor y que se encargara de llamar y activar o desactivar otros módulos cuando sea necesario. Este CORE también estaría implantado en el terminal centralizado (véase apartado 6.1 Visión general de la arquitectura, Figura 7).

Esta alternativa permite centrarnos únicamente en un punto de la red y gestionar y controlar dicha red desde un lugar. De esta manera tendríamos un único terminal con los recursos necesarios y dedicado en exclusiva a realizar dicha función.

#### 5.1.2 Inteligencia distribuida

Otra posible opción consistiría en distribuir la inteligencia del sistema en varios puntos, como pueden ser el propio terminal y los routers. En este caso sería interesante que a los routers se les dotara de capacidad de actualización y al terminal de las demás funciones. De esta forma los recursos necesarios para la inteligencia del programa se situarían en varios puntos y así, se reduciría la carga que soportaría el terminal en comparación a un diseño centralizado. Sin embargo, la inteligencia implantada en los routers debería estar replicada en todos los routers del sistema y por lo tanto la suma total de recursos utilizados para hacer la misma función sería mayor. Por otro lado, se consumirían recursos de los routers que podrían estar siendo utilizados para el propio rutado y así, ofrecer un servicio más óptimo.

#### 5.1.3 Selección de alternativas

En base a los siguientes criterios se han valorado las diferentes opciones:

Criterio	Peso	Centralizado	Distribuido
Rendimiento	0,3	8	8
Recursos necesarios	0,3	8	5
Gestión	0,2	7	4
Coste	0,2	7	6
<b>RESULTADO</b>		<b>7,6</b>	<b>5,9</b>

Tabla 1: Selección de alternativas "Distribución de la inteligencia"

En base a esta valoración se ha optado por un sistema con inteligencia centralizada por los beneficios e inconvenientes descritos previamente.

## 5.2 Comunicación con base de datos RIPE

Una parte importante de este proyecto consiste en la obtención de la información de rutas para comprobar después que la información que se está transmitiendo es adecuada. Eso forma parte de la monitorización del sistema. En este contexto esta base de datos que se indica, RIPE, es un elemento fundamental porque es una entidad que dispone de una base de datos donde se incluyen las direcciones asignadas a los proveedores que es precisamente lo que se va a controlar en este proyecto y que se indicará en este documento más adelante (véase ANEXO I, apartado 4.1). RIPE permite acceder a su base de datos de tres formas diferentes:

### 5.2.1 Interfaz web

Este método proporciona una sencilla forma de realizar una consulta a la base de datos RIPE [5]. Sólo es necesario hacer un click. Es una interfaz web muy intuitiva y como se puede apreciar en las figuras siguientes, permite definir diferentes tipos de búsquedas como pueden ser las búsquedas por tipo de objetos o búsquedas inversas. También permite incluir otras bases de datos en las búsquedas para que en caso de que la información consultada no se encuentre en RIPE, esta derive la petición a otras bases de datos. Sin embargo, este método está orientado solamente para hacer consultas esporádicas y no para trabajar con ella debido a su escasa capacidad de programación.

You can search up to five terms at once in the search box above, separating them with a semi-colon.

Sources Types Hierarchy flags Inverse lookup

?  
 as-block  
 as-set  
 aut-num  
 domain  
 filter-set  
 inet6num  
 inetnum  
 inet-rtr  
 irt  
 key-cert  
 mntner  
 organisation  
 peering-set  
 person  
 poem  
 poetic-form  
 role  
 route  
 route6  
 route-set  
 rtr-set

The equivalent Whois query flags are shown below.

`-r --resource as3262`

By submitting this form you explicitly express your agreement with the [RIPE Database Terms and Conditions](#)

Search

Figura 3: Búsqueda interfaz web

Search results PERMA XML JSON

This is the RIPE Database search service. The objects are in RPSL format. The RIPE Database is subject to [Terms and Conditions](#).

Responsible organisation: [SARENET, S.A.](#)  
Abuse contact info: [abuse@sarenet.es](mailto:abuse@sarenet.es)

aut-num: AS3262  
as-name: SARENET  
org: ORG-SS1-RIPE  
import: from AS42 action pref=100; accept AS-PCH  
export: to AS42 announce AS-SARENET  
import: from AS174 action pref=100; accept ANY  
export: to AS174 announce AS-SARENET  
import: from AS260 accept AS-XCONNECT24  
export: to AS260 announce AS-SARENET  
import: from AS714 action pref=100; accept AS714  
export: to AS714 announce AS-SARENET  
import: from AS766 action pref=100; accept AS-REDIRIS  
export: to AS766 announce AS-SARENET  
import: from AS1126 action pref=100; accept RS-VANCIS-ROUTESET  
export: to AS1126 announce AS-SARENET  
import: from AS1200 action pref=100; accept AS1200  
export: to AS1200 announce AS-SARENET

Login to update Ripestat

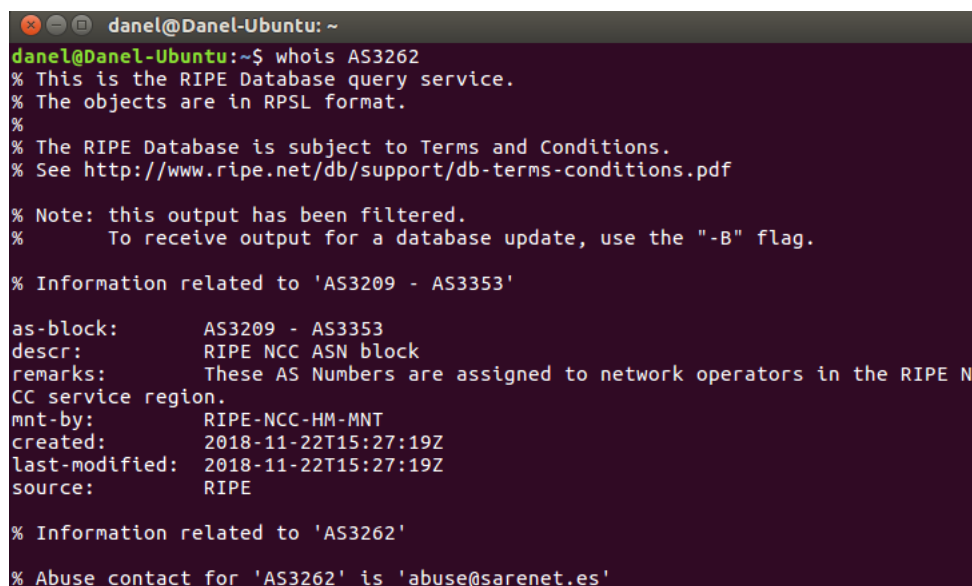
Figura 4: Resultados interfaz web

## 5.2.2 Herramienta whois

Otra alternativa es el cliente whois. Whois es una herramienta proporcionada por RIPE que está a disposición libre [6]. Esta alternativa tiene la diferencia de que hay que interactuar en forma de comandos.

Whois es un pequeño programa que se conecta a la base de datos Whois de RIPE, pasa la solicitud de búsqueda a la base de datos e imprime la respuesta. La mayoría de los sistemas operativos incluyen un cliente whois básico por defecto. Sin embargo, estos clientes a menudo no reconocen todas las opciones de consulta agregadas que permite la base de datos RIPE.

Esta alternativa se caracteriza por ser más eficiente que la anterior porque en vez de estar basado en la transmisión de páginas HTML está basado en la utilización de comandos, lo que hace que los mensajes que se intercambian sean más ligeros. En la figura siguiente se muestra un ejemplo del uso de esta herramienta.



```
danel@Danel-Ubuntu: ~
danel@Danel-Ubuntu:~$ whois AS3262
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to 'AS3209 - AS3353'
as-block:      AS3209 - AS3353
descr:         RIPE NCC ASN block
remarks:       These AS Numbers are assigned to network operators in the RIPE N
CC service region.
mnt-by:        RIPE-NCC-HM-MNT
created:       2018-11-22T15:27:19Z
last-modified: 2018-11-22T15:27:19Z
source:        RIPE

% Information related to 'AS3262'
% Abuse contact for 'AS3262' is 'abuse@sarenet.es'
```

Figura 5: Herramienta whois

## 5.2.3 RESTful API

RIPE ofrece una interfaz estándar que se utiliza frecuentemente entre sistemas con el objetivo de proporcionar accesos a informaciones estructuradas de una determinada manera [7]. Hace uso de estándares como JSON o XML y que igualmente son muy utilizados y que por tanto es muy sencillo encontrar librerías que permitan trabajar con ellas. Con la figura siguiente se muestra una consulta a RIPE mediante la API RESTful.

```
https://rest.db.ripe.net/search.json?source=ripe&query-string=as3262
```

Figura 6: Petición RESTful API

Como se puede apreciar en la petición, este método hace uso del protocolo HTTPS, que es la versión mejorada y segura del protocolo HTTP. Es el protocolo de comunicación que permite las transferencias de información en la World Wide Web [8]. Es un protocolo



sin estado que se basa en el intercambio de dos tipos de mensajes: las peticiones y sus respectivas respuestas.

Para realizar una petición se debe crear una URL como la que aparece en la Figura 6. Esta se divide en cuatro partes:

- **https**: es el protocolo de comunicación.
- **rest.db.ripe.net**: es el anfitrión de los recursos.
- **/search.json**: es la ruta de acceso.
- **?source=ripe&query-string=as3262**: es la cadena de consulta. Esta se utiliza para interactuar con una base de datos. Es la parte de la URL que se debe pasar a aplicaciones web para que estas los procesen de forma adecuada.

Este es el método que más opciones y alternativas ofrecerá cuando se profundice más en el desarrollo del proyecto.

### 5.2.4 Selección de alternativas

Tras analizar las alternativas propuestas se procede a elegir la más adecuada:

Criterio	Peso	Interfaz web	Herramienta whois	RESTful API
Rendimiento	0,3	5	7	8
Compatibilidad	0,2	9	6	9
Nivel de programación	0,4	4	6	8
Facilidad de uso	0,1	9	7	5
	<b>RESULTADO</b>	5,8	6.4	7,9

Tabla 2: Selección de alternativas "Comunicación con base de datos RIPE"

Como se puede observar, la alternativa que mejor se adapta a lo que este proyecto busca es acceder a la base de datos de RIPE mediante RESTful API.

## 5.3 Desarrollo del sistema completo

Tanto cuando se definió el proyecto como en sus etapas posteriores, se han encontrado varias herramientas creadas por otras entidades que solucionan algunos de los problemas que se plantean al comienzo de este documento.

### 5.3.1 Herramienta bgpq3

Es una herramienta de software libre que se encuentra disponible en github. La herramienta bgpq3 se utiliza para crear filtros automatizados en routers Cisco y Juniper. Además, permite generar listas de prefijos entre otro tipo de listas (véase ANEXO II, apartado 1).

A primera vista, la función que realiza esta herramienta se asemeja mucho a lo que este proyecto tiene como objetivo lograr. Nada más lejos de la realidad, ya que la información que se obtiene con esta herramienta no es la que se busca. Además, se crearon complicaciones al adaptarlo a la arquitectura que se necesita. Al comenzar el proyecto se valoró la utilización de esta herramienta, pero como se puede ver, no se llevó a cabo.

### 5.3.2 Solución propia

Esta alternativa se basa en que nosotros mismos creáramos un software que solucionara la problemática que se plantea. Así, tendríamos más libertad para crear una herramienta que se ajuste perfectamente a las especificaciones y situación del proyecto. También permitiría limitar las funcionalidades hasta donde lo viéramos oportuno y dejar puertas abiertas para su mejora posterior.

### 5.3.3 Selección de alternativas

La selección se ha hecho en función de los siguientes criterios:

Criterio	Peso	bgpq3	Solución propia
Rendimiento	0,5	5	10
Adaptabilidad	0,3	4	9
Progresión	0,2	5	10
RESULTADO		4,7	9,7

Tabla 3: Selección de alternativas "Desarrollo del sistema completo"

Se ha decidido empezar el desarrollo sin seguir herramientas ya creadas que dirijan el camino del proyecto. Implantar una solución propia brinda mayor libertad y mayor capacidad para alcanzar mejor los objetivos.

## 5.4 Desarrollo del módulo Optimizador de prefijos

Como se explica con más detalle en la sección del diseño de la solución, tras obtener la información necesaria de la base de datos RIPE, se crea una lista de prefijos de red de los Sistemas Autónomos vecinos. Para mejorar el rendimiento y disminuir la carga de información de los routers se ha diseñado el módulo Optimizador de prefijos que optimiza dicha lista. Para ello, una de las técnicas que aplica es la técnica de supernetting o agregación de rutas y es aquí donde surgen dos alternativas de diseño:

### 5.4.1 Módulo pyipcalc

A simple vista parece una tarea sencilla diseñar el código que simplifique una lista de prefijos. Sin embargo, es una tarea compleja ya que la casuística es muy grande (una red puede contener a otra, pueden ser contiguas, pueden tener rangos totalmente diferentes...). El módulo pyipcalc no soluciona todo el problema, pero sí lo disminuye. Consiste en una librería para Python [9] que se basa en hacer cálculos sencillos de direcciones IP [10]. Contiene una función que insertándole como argumentos dos redes y un límite de supernetting, devuelve una red simplificada.

Este módulo no aporta la solución completa que se necesita, pero sí simplifica mucho la programación del módulo Optimizador de prefijos.

### 5.4.2 Desarrollo propio

Otra alternativa sería que fuéramos nosotros mismos los que implementáramos dicha funcionalidad al completo. Sería una tarea que conllevaría demasiado tiempo puesto que se debería tener en cuenta toda la casuística de la agregación de rutas de dos redes y habría que escribir muchas líneas de código.

### 5.4.3 Selección de alternativas

Una vez descritas las dos alternativas se procede a su selección en base a los siguientes criterios:

Criterio	Peso	pyipcalc	Desarrollo propio
Rendimiento	0,4	9	9
Tiempo invertido	0,3	8	3
Coste	0,3	7	5
	<b>RESULTADO</b>	<b>8,1</b>	<b>6</b>

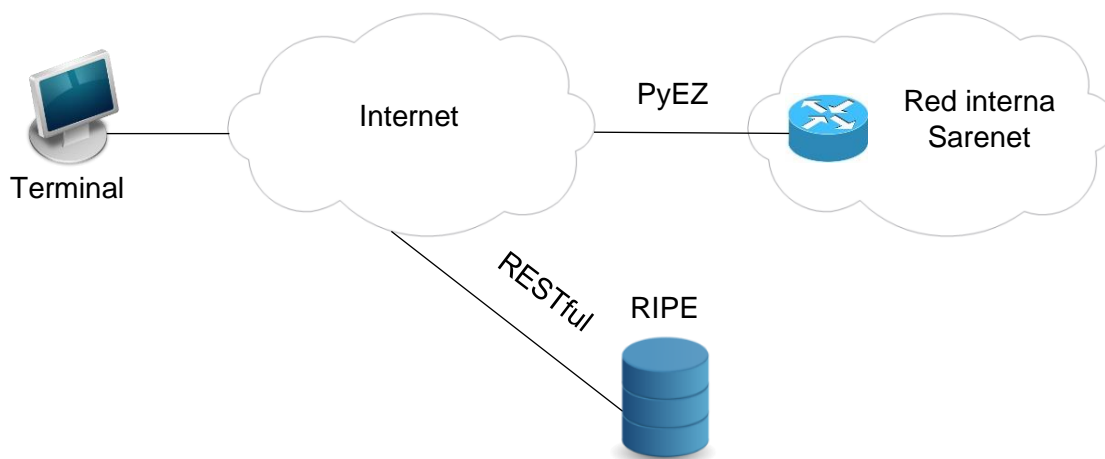
Tabla 4: Selección de alternativas "Desarrollo del módulo Optimizador de prefijos"

Para el diseño del módulo optimizador de prefijos se ha decidido utilizar el módulo pyipcalc, principalmente por el tiempo que se ahorra y a su vez, por la disminución del coste que conlleva.

## 6 DESCRIPCIÓN DE LA SOLUCIÓN

### 6.1 Visión general de la arquitectura

Antes de comenzar el diseño, se muestra un diagrama que permite observar el escenario de funcionamiento del sistema. La siguiente figura representa una idea genérica de los elementos que forman parte y cómo es la comunicación entre ellos. Por un lado, tenemos el núcleo del sistema situado en un ordenador, el cual se conecta a la base de datos RIPE a través de Internet para realizar consultas con el objetivo de obtener información. Por otro lado, tenemos otra comunicación entre el ordenador y la red interna del proveedor, en este caso Sarenet, para realizar las modificaciones oportunas que resuelvan los errores de configuración del router. En la figura se muestran además algunos de los componentes indicados en los apartados de análisis de alternativas y de diseño de la solución como son RESTful y PyEZ.



*Figura 7: Arquitectura general*

### 6.2 Diseño de la solución

En este apartado se describen detalladamente los módulos que forman parte del sistema completo. Se analizarán cada uno de ellos por separado destacando las funciones más importantes. Enlazando con la Figura 7 que se acaba de mostrar, en la Figura 8 se pueden observar los diferentes módulos que componen el sistema junto con sus respectivas localizaciones.

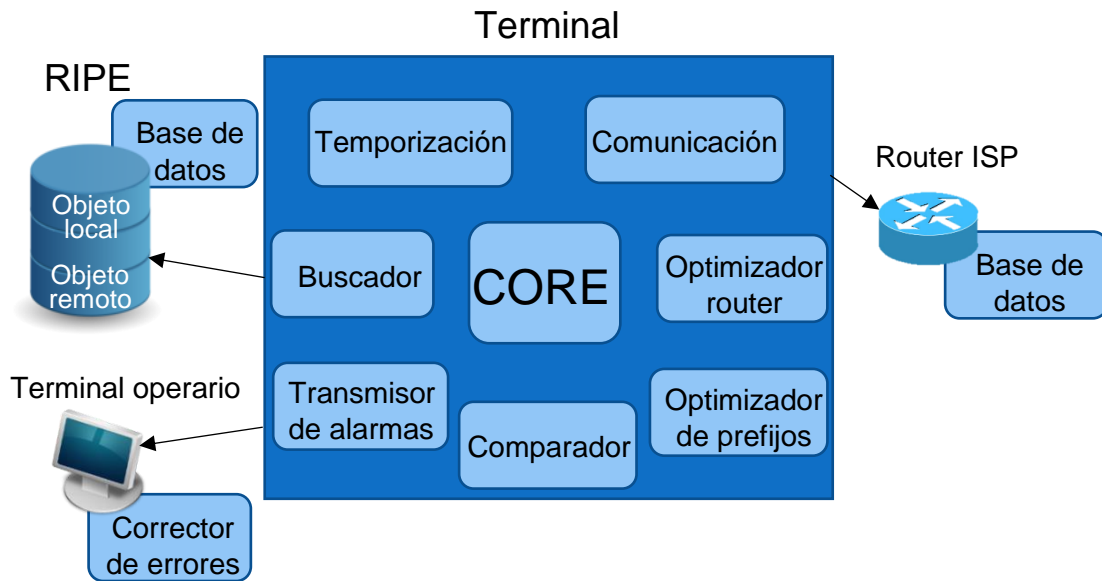


Figura 8: Módulos del sistema

A continuación, se describen en detalle los módulos que constituyen el sistema y que son mencionados en la Figura 8.

### 6.2.1 Base de datos

En este apartado se analizan las diferentes fuentes de información utilizadas para obtener la información necesaria para lograr los objetivos propuestos.

La principal base de datos que se utiliza es la de RIPE NCC. Es una base de datos pública que proporciona información detallada sobre los registros de las direcciones IP y de los números AS. Muestra las organizaciones que poseen los recursos, dónde se realizaron las asignaciones y los detalles de contacto de las redes. También contiene información sobre las políticas de rutado (en el RIR) de los operadores de red. No existe una entidad que controle la actualización y la validez de esta base de datos, sino que son las propias organizaciones poseedoras de esos recursos los responsables.

Como se ha analizado en el apartado de análisis de alternativas, existen diferentes opciones para consultar la información, que varían desde una interfaz web, pasando por una herramienta para la terminal de comandos y llegando a una API RESTful. Siendo esta última la que se ha seleccionado.

En Sarenet hay más de 300 sesiones de Peering y muchas de ellas anuncian ASs que no están registrados en RIPE. Es por ello la necesidad de consultar múltiples bases de datos. RIPE NCC opera con mirrors de las otras bases de datos de los RIR, así como con los principales registros de rutado, conocidos como Global Resource Service (GRS). Aquí se incluyen las siguientes bases de datos:

- **AFRINIC:** African Network Information Centre es el Registro Regional de Internet para África.
- **APNIC:** Asia-Pacific Network Information es el RIR para la región de Asia y el Pacífico.
- **ARIN:** American Registry for Internet Numbers es la organización encargada en América Anglosajona y varias islas de los océanos Pacífico y Atlántico.

- **LACNIC:** Latin America and Caribbean Network Information Centre es el correspondiente a América Latina y el Caribe.
- **JPIRR:** Japan Network Information Center es el RIR para Japón.
- **RADB:** Routing Assets Database dirigida por Merit Networks.

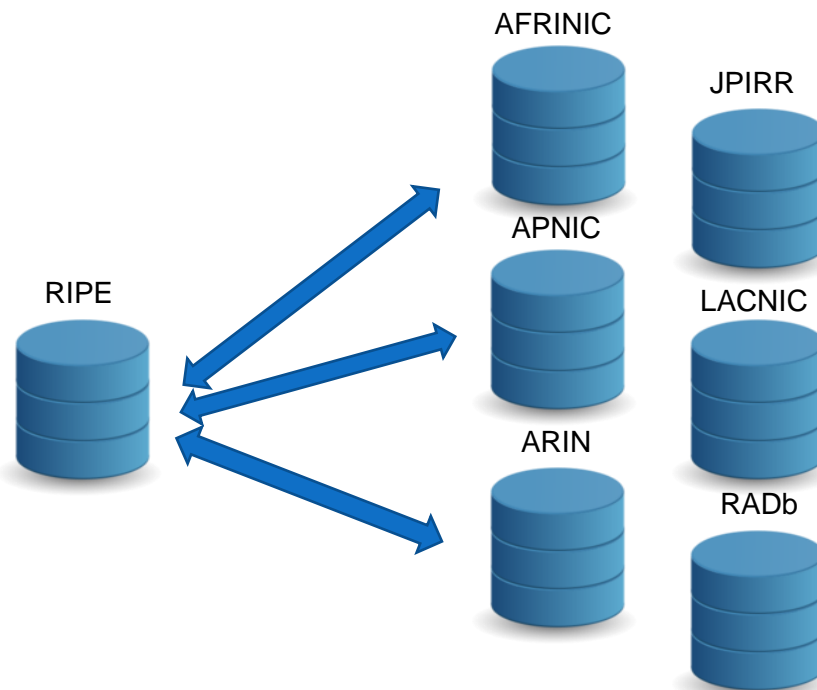


Figura 9: Bases de datos

Un ejemplo para una consulta en múltiples bases de datos sería:

```
https://rest.db.ripe.net/search.json?source=ripe&source=AFRINIC-GRS&source=ARIN-GRS&source=RADB-GRS&source=APNIC-GRS&source=JPIRR-GRS&source=LACNIC-GRS&query-string=as20815
```

Figura 10: Consulta a múltiples bases de datos

Como se puede observar, para realizar la consulta se utiliza el protocolo HTTPS explicado previamente en el apartado de análisis de alternativas. En este caso, en la cadena de consultas se indicarán las bases de datos que se necesitan utilizar, que son AFRINIC-GRS, ARIN-GRS, RADB-GRS, APNIC-GRS, JPIRR-GRS y LACNIC-GRS.

En este proyecto el primer paso es obtener la lista de todos los ASs con las que el ISP tiene relación. Para ello, tenemos tres fuentes de información:

- **Objeto local ISP:** buscamos en la base de datos RIPE el objeto correspondiente al ISP, en este caso Sarnet, es decir el AS3262, mediante la siguiente consulta:

```
https://rest.db.ripe.net/search.json?source=ripe&source=AFRINIC-GRS&source=ARIN-GRS&source=RADB-GRS&source=APNIC-GRS&source=JPIRR-GRS&source=LACNIC-GRS&query-string=AS3262
```

*Figura 11: Consulta a objeto local ISP*

Para un AS se almacenan varios tipos de objetos, como el “aut-num”, as-set” o “person” entre muchos otros. En este módulo sólo es necesaria la información que hay en los objetos tipo “aut-num”.

El resultado que se obtiene, como se verá más adelante, es muy similar al obtenido en la Figura 4. Como se ha visto, obtenemos información sobre los “import” y “export”. Llegados a este punto, interesa conocer los campos “import” y “export”. El “import” indica que Sarnet recibe información del AS peer y el “export” que Sarnet le envía. El valor de estos dos campos puede tener dos valores: “ANY” o el AS del vecino correspondiente como puede ser “AS20815”.

Si el AS es un proveedor de Sarnet, en el “import” habrá un “ANY” y si es un cliente, el “ANY” estará en el “export”. En cambio, en cuanto a las relaciones de peering, sólo aparecerá el valor del AS correspondiente, ya que una relación de este tipo es exclusiva entre dos entidades.

Por lo tanto, para obtener la lista que tenemos como objetivo, hay que obtener todos los valores de los “import” o “export”.

- **Objeto remoto ISP:** en este caso, consultamos en el objeto del AS remoto la información que existe sobre el ISP. Para ello, se procede a hacer la siguiente consulta que es muy similar al de la Figura 10 y 11:

```
https://rest.db.ripe.net/search.json?source=ripe&source=AFRINIC-GRS&source=ARIN-GRS&source=RADB-GRS&source=APNIC-GRS&source=JPIRR-GRS&source=LACNIC-GRS&query-string=AS20815
```

*Figura 12: Consulta a objeto remoto ISP*

Ahora sólo habría que buscar el valor AS3262 en los campos “import” y “export”. En caso de no encontrarlo significa que no existe ninguna relación con el ISP.

- **Router ISP:** En los routers del ISP deben estar configurados todos los peer. Se puede obtener esta lista accediendo al router a:

```

[edit protocols bgp group eBGP-Peering-Espanix1]
neighbor 193.149.1.7 {
    description GRN;
    import bgp-AS20815-in;
    peer-as 20815;
}
neighbor 193.149.1.8 {
    description Colt;
    peer-as 8220;
}
neighbor 193.149.1.12 {
    description "Orange (antes Uni2)";
    family inet {
        unicast {
            prefix-limit {
                maximum 7000;
                teardown;
            }
        }
    }
    peer-as 12479;
}
neighbor 193.149.1.18 {
    description BT;
    peer-as 8903;
}

```

*Figura 13: Fragmento de lista de vecinos obtenida mediante el router*

Como se puede apreciar en la figura, se está accediendo al grupo BGP llamado eBGP-Peering-Espanix1. Estos grupos están configurados en los routers por los propios proveedores y simplemente agrupan a ciertos vecinos. Mediante la palabra reservada `neighbor` se puede ver la información respectiva a cada vecino que forma el grupo y mediante el `peer-as` el ASN de dicho vecino.

Al hacer todas las consultas en una base de datos pública donde cada organización publica su información, hay consultas que no son válidas. Es muy típico encontrar ASs que su información no está actualizada o que no siga los patrones de nombre utilizados por otras organizaciones como por ejemplo, que todos los grupos de AS se identifican con un guión en medio ("AS-WDSET"). También se ha tenido que tener en cuenta que unos ASs almacenaban la información en atributos erróneos. Es por ello la necesidad del siguiente módulo.



## 6.2.2 Transmisor de alarmas

Esta implementación sirve para notificar inconsistencias de las bases de datos y de las fuentes de información previamente analizadas.

En primer lugar, se deben comparar las 3 fuentes de información. Las 3 listas obtenidas por ellos deben ser idénticas. De lo contrario, significará que alguna de las 3 fuentes de información está corrupta. En este caso, se le enviará un correo al operario encargado describiendo detalladamente el fallo encontrado: qué fuente de información hay que corregir y qué información tendrá que añadir o eliminar. En caso de que la fuente dañada sea el objeto remoto, el correo irá dirigido a la organización correspondiente.

En la mayoría de los casos, las fuentes erróneas suelen ser las remotas, ajenas al ISP. Muchos ASs no están en la base de datos, forman bucles y otros no siguen los patrones. En estas situaciones también enviaremos un correo a la entidad advirtiéndole de sus fallos. En caso de que no los actualicen, el ISP cortará la relación que tiene con ellos filtrando sus paquetes BGP. Un ejemplo de notificación sería el siguiente:

```
Peering configurado en el router mx104lab y
no documentado en el objeto AS3262 local
ni en el objeto AS12430 remoto:

Router: mx104
  Group: eBGP-Peering-Espanix1
  Description: VODAFONE
  Peer IP: 193.149.1.66
  Peer AS: 12430
  import: no tenemos comunicación con AS12430
Política documentada en objeto AS3262 local:
  import: [NO ENCONTRADO]
  export: [NO ENCONTRADO]
Política documentada en objeto AS12430 remoto:
  import: [NO ENCONTRADO]
  export: [NO ENCONTRADO]
```

Figura 14: Notificación

## 6.2.3 CORE

En este módulo es donde reside la inteligencia del sistema. Se sitúa en el terminal y su objetivo principal es obtener toda la información que interesa de las fuentes de información que disponemos, trabajar con ella y dependiendo de la situación, tomar una acción u otra. Para ello, está en constante comunicación con los demás módulos para cuando sea necesario activarlos para que realicen sus respectivas funciones. Es el motor del sistema.

## 6.2.4 Buscador

Este módulo tiene como objetivo obtener la lista de prefijos que anuncia cada AS al ISP. En este punto tenemos la lista completa de AS vecinos del ISP. Por lo tanto, ahora debemos descubrir qué ASs anuncia cada uno de los vecinos al ISP. Existen dos posibles anuncios: que sólo se anuncie a sí mismo o que anuncie un grupo de ASs. A dicho grupo se le denomina AS-set. Haciendo una consulta del objeto local del ISP en RIPE y buscando en los “import” se obtiene el AS-set, como está remarcado en negrita en la Figura 15. El significado de la información recibida es el siguiente: el Sistema Autónomo AS3262, es decir el objeto local del ISP, acepta el grupo AS-GRN recibida del Sistema Autónomo AS20815 y a su vez, el AS3262 le envía al AS20815 el grupo AS-SARENET.

```
import:          from AS20815 action pref=100; accept AS-GRN
export:         to AS20815 announce AS-SARENET
```

*Figura 15: Consulta AS-set*

Este AS-set estará formado por varios “members”. Estos pueden ser ASs o AS-sets que a su vez estarán formados por otros “members”. Para lograr la lista de ASs que forman un grupo, es necesario buscar el objeto AS-set en la base de datos y buscar los campos “members”. La consulta a realizar es muy similar a las descritas anteriormente. En este caso en vez de añadir un AS se añadirá el AS-set, como se puede ver en el siguiente ejemplo:

```
https://rest.db.ripe.net/search.json?source=ripe&source=AFRINI  
C-GRS&source=ARIN-GRS&source=RADB-GRS&source=APNIC-  
GRS&source=JPIRR-GRS&source=LACNIC-GRS&query-string=AS-GRN
```

*Figura 16: Consulta members de un AS-set*

JSON	Datos sin procesar	Cabeceras
Guardar	Copiar	Contraer todo
Expandir todo		

```

▼ service:
  name: "search"
  parameters: {...}
  objects:
    ▼ object:
      ▼ 0:
        type: "as-set"
        link: {...}
        source: {...}
        primary-key: {...}
        attributes:
          ▼ attribute:
            ▼ 0:
              name: "as-set"
              value: "AS-GRN"
            ▼ 1:
              name: "descr"
              value: "GRN Serveis Telematics and BGP customers"
            ▼ 2:
              link:
                type: "locator"
                href: "http://rest.db.ripe.net/ripe/aut-num/AS20815"
                name: "members"
                value: "AS20815"
                referenced-type: "aut-num"
            ▼ 3:
              link:
                type: "locator"
                href: "http://rest.db.ripe.net/ripe/aut-num/AS44280"
                name: "members"
                value: "AS44280"
                referenced-type: "aut-num"
            ▼ 4:
              link:
                type: "locator"
                href: "http://rest.db.ripe.net/ripe/aut-num/AS60551"
                name: "members"
                value: "AS60551"
                referenced-type: "aut-num"
            ▼ 5:
              link:
                type: "locator"
                href: "http://rest.db.ripe.net/ripe/role/GN613-RIPE"
                name: "tech-c"
                value: "GN613-RIPE"
                referenced-type: "role"
            ▼ 6:
              link:
                type: "locator"

```

Figura 17: Members de as-set

Esta figura corresponde a un fragmento del objeto JSON recibido como respuesta a la consulta realizada en la figura anterior. Los campos name y value quedan remarcados que son los que se necesitan para saber cómo está formado el AS-set. Siguiendo con el ejemplo anterior, el AS-set AS-GRN está formado por:

```

members: AS20815
members: AS44280
members: AS60551

```

Todos los ASs tienen un objeto "route" donde se almacenan los prefijos de red. Para obtenerlos, es necesario buscar los objetos tipo "route" y hacer una búsqueda inversa con el atributo origin el AS vecino. Es decir, se pretende lograr los prefijos de red que envían cada uno de los "members", como por ejemplo el AS20815:

```
https://rest.db.ripe.net/search.json?source=ripe&source=RIPE-  
NONAUTH&source=AFRINIC-GRS&source=ARIN-GRS&source=RADB-  
GRS&source=APNIC-GRS&source=JPIRR-GRS&source=LACNIC-GRS&query-  
string=AS20815&inverse-attribute=origin&&type-  
filter=route&flags=no-referenced
```

*Figura 18: Consulta de objeto route con búsqueda inversa de atributo origin*

JSON	Datos sin procesar	Cabeceras
Guardar	Copiar	Contraer todo
		Expandir todo
▶ service:		{...}
▼ parameters:		
▶ inverse-lookup:		{...}
▼ type-filters:		
▼ type-filter:		
▼ 0:		
id:		"route"
▶ flags:		{...}
▶ query-strings:		{...}
▶ sources:		{...}
▼ objects:		
▼ object:		
▼ 0:		
type:		"route"
▶ link:		{...}
▶ source:		{...}
▼ primary-key:		
▼ attribute:		
▼ 0:		
name:		"route"
value:		"80.64.32.0/20"
▼ 1:		
name:		"origin"
value:		"AS20815"
▶ attributes:		{...}
▼ 1:		
type:		"route"
▶ link:		{...}
▶ source:		{...}
▼ primary-key:		
▼ attribute:		
▼ 0:		
name:		"route"
value:		"80.64.32.0/24"
▼ 1:		
name:		"origin"
value:		"AS20815"
▶ attributes:		{...}
▼ 2:		
type:		"route"
▶ link:		{...}
▶ source:		{...}
▼ primary-key:		
▼ attribute:		
▼ 0:		
name:		"route"
value:		"80.64.33.0/24"
▼ 1:		

Figura 19: Prefijos de un AS

La Figura 19 corresponde a un fragmento de la respuesta recibida a la consulta mencionada previamente y los campos name y value vuelven a estar destacados puesto que son los que se necesitan para conocer los prefijos que anuncia el AS20815.

Llegados a este punto del funcionamiento del programa y siguiendo con el ejemplo para el AS vecino AS20815, habríamos logrado esta lista:

route:	80.64.32.0/20	}	AS20815
route:	80.64.32.0/24		
route:	80.64.33.0/23		
route:	80.64.34.0/23		
route:	80.64.36.0/23		
route:	80.64.38.0/23		
route:	80.64.40.0/23		
route:	80.64.42.0/23		
route:	80.64.47.0/24	}	
route:	195.93.170.0/23	}	AS44280
route:	185.29.212.0/24	}	AS60551
route:	185.29.213.0/24		
route:	185.29.214.0/24		
route:	185.29.215.0/24		
route:	91.223.143.0/24		
route:	91.224.150.0/24		
route:	91.224.151.0/24		

Lo mismo hay que hacer para direcciones IPv6, pero el objeto en vez de ser "route" se denomina "route6". Lo único que cambia respecto a la consulta anterior, Figura 19, es lo que se puede apreciar en **negrita** en la siguiente figura:

```
https://rest.db.ripe.net/search.json?source=ripe&source=RIPE-
NONAUTH&source=AFRINIC-GRS&source=ARIN-GRS&source=RADB-
GRS&source=APNIC-GRS&source=JPIRR-GRS&source=LACNIC-GRS&query-
string=AS20815&inverse-attribute=origin&&type-
filter=route6&flags=no-referenced
```

Figura 20: Consulta de objeto route6 con búsqueda inversa de atributo origin

Añadiendo los nuevos resultados a la lista correspondiente de IPv6:

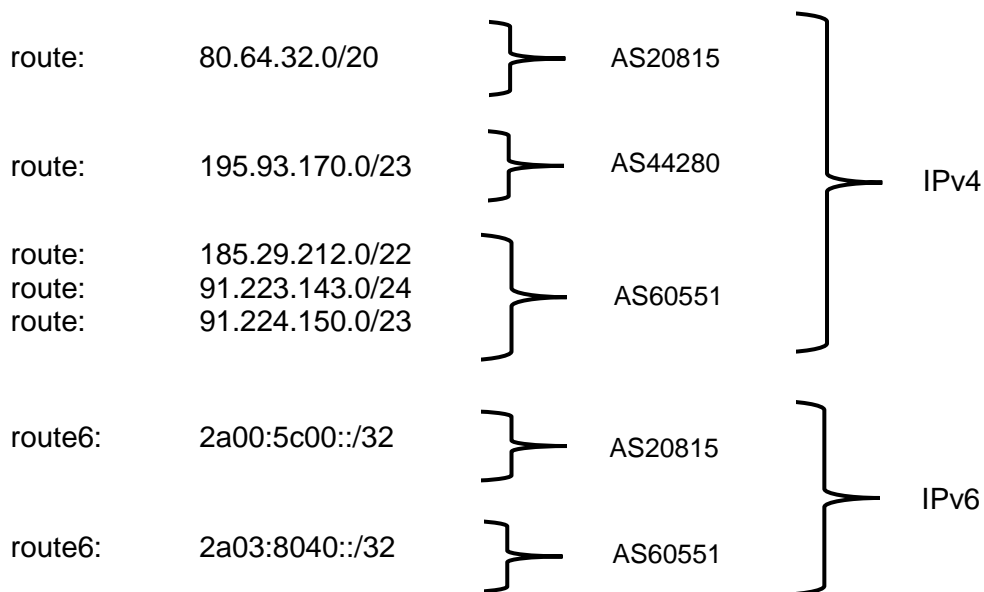
route6:	2a00:5c00::/32	}	AS20815
route6:	2a03:8040::/32	}	AS60551

Por lo tanto, se utilizarán dos listas de prefijos diferentes, una para prefijos IPv4 y otra para IPv6. Con estas listas de prefijos se crearán las políticas de filtrado y estas serán aplicadas a cada vecino correspondiente individualmente. Los vecinos pueden ser de dos tipos de familias: inet o inet6 (IPv4 e IPv6 respectivamente). Si se utilizara solamente una lista con todos los prefijos en ella, cuando el router verifique la configuración alertará de un error ya que a vecinos de la familia inet se le están aplicando filtros de direcciones IPv6 y viceversa.

## 6.2.5 Optimizador de prefijos

En este módulo se simplifica la lista de prefijos de cada AS. El objetivo de este módulo es disminuir los registros que hay en la configuración del router. Para ello se ha aplicado la técnica de “supernetting”. Esta técnica es el proceso de resumir un grupo de redes contiguas en una sola red más grande. Es la base para la mayoría de los protocolos de rutado que utilizan Internet actualmente. A “supernetting” también se le conoce como agregación de rutas [11]. En este proyecto hemos considerado que lo más óptimo es agregar rutas hasta crear redes de como máximo 65534 hosts, es decir, hasta un /16.

Como se ha visto en la sección de análisis de alternativas, para el desarrollo de este módulo se ha utilizado la librería “pyipcalc”. La lista de prefijos se ha simplificado hasta obtener lo siguiente:



## 6.2.6 Comparador

Una vez obtenida la lista total de AS vecinos junto a los prefijos de red que anuncian, los almacenaremos en un archivo con formato “.yaml”. YAML (“YAML Ain’t Markup Language”) es un lenguaje de serialización de datos legible por humanos [12]. Se usa comúnmente con archivos de configuración, pero podría usarse en muchas aplicaciones donde los datos se almacenan o se transmiten.

YAML se enfoca en muchas de las mismas aplicaciones que XML por ejemplo. Sin embargo, tiene una sintaxis mínima que intencionalmente rompe la compatibilidad con SGML. Utiliza la sangría al estilo Python para indicar el anidamiento y un formato más compacto para listas y para mapas que crean a su vez supersets de JSON. El archivo que utilizamos tiene la siguiente forma:

```
datavars_changes.yml
lista_ASN:
- eBGP-AS20815:
  - 185.29.212.0/22
  - 195.93.170.0/23
  - 80.64.32.0/20
  - 91.223.143.0/24
  - 91.224.150.0/23
- eBGP6-AS20815:
  - 2a03:8040::/32
  - 2a00:5c00::/32
```

Figura 21: Estructura de archivo YAML

El programa está pensado para ser ejecutado automáticamente cada un cierto periodo de tiempo, exactamente dos veces al día. Esta frecuencia de ejecución es suficiente para detectar los cambios en los vecinos y obtener una rápida respuesta, ya que dichos cambios no son tan frecuentes. De esta forma, lo más probable es que la lista obtenida de una ejecución a otra no varíe o apenas lo haga. Entonces, lo más eficiente sería tener en cuenta solamente los cambios que se hayan detectado de una ejecución a otra.

Para ello, se trabaja con tres archivos YAML que tienen la estructura de la Figura 21, llamados “datavars\_old.yaml”, “datavars\_new.yaml” y “ datavars\_changes.yaml”. Los nombres de estos archivos son irrelevantes, pero son descriptivos y facilitan la comprensión de este apartado del documento.

La primera vez que el programa se ejecute, almacenará la lista obtenida en un archivo llamado por ejemplo “datavars\_old.yaml”. En las siguientes ejecuciones, se almacenará la nueva lista obtenida en un archivo llamado “datavars\_new.yaml” y se comparará con el archivo anterior “datavars\_old.yaml”. Obteniendo así un archivo “datavars\_changes.yaml” que contiene únicamente la información que nos interesa actualizar en el router.

## 6.2.7 Comunicación

Este módulo es el encargado de establecer la conexión con el router del ISP. Los routers utilizados en este proyecto son los Juniper mx104. Se ha utilizado el software Junos PyEZ [13] para la comunicación entre el programa y los routers. Junos PyEZ es un microframework para Python que permite administrar y automatizar dispositivos que ejecutan el sistema operativo Junos OS. Este software está diseñado para proporcionar las capacidades que un usuario tendría en la interfaz de línea de comandos (CLI) del sistema operativo Junos en un entorno creado para tareas de automatización.

Junos PyEZ permite conectarse directamente a un dispositivo mediante una conexión de consola serie, telnet o una sesión de NETCONF a través de SSH. Además, también permite administrar, recuperar, comparar y cargar configuraciones en varios formatos estándar para datos de configuración que incluyen texto ASCII, elementos XML, notación JSON y plantillas Jinja2 [14].

Jinja2 es un motor de plantillas para Python que permite generar documentos a partir de plantillas predefinidas. Las plantillas, que son archivos de texto, proporcionan flexibilidad mediante el uso de expresiones y variables. En este proyecto hemos utilizado



este tipo de formato de configuración. Es muy recomendable el uso para aplicaciones con configuraciones similares. Por ejemplo, en lugar de agregar manualmente las mismas configuraciones para cada interfaz de dispositivo, es posible crear una plantilla Jinja2 que se repita para una lista de interfaces definida y cree las declaraciones de configuración necesarias. Como se puede apreciar, la plantilla Jinja2 que se ha utilizado tiene el formato de configuración del router:

```
policy-options {
  {%- for ANS in lista_ASN %}
  {%- for name, sublist in ASN.items() %}
    prefix-list {{name}} {
      {%- for prefix in sublist %}
        {{prefix}}
      {%- endfor %}
    }
    policy-statement bgp-{{ name }}-in{
      term1 {
        from {
          prefix-list-filter {{ name }} orlonger;
        }
        then accept;
      }
      term 2 {
        then reject;
      }
    }
  {%- endfor %}
  {%- endfor %}
}
```

Figura 22: Plantilla Jinja2

Siempre y cuando haya habido un cambio en nuestras fuentes de información y debamos actualizar los routers, el programa que reside en el terminal se conectará a los routers mediante una sesión de NETCONF a través de SSH y cargará el archivo de configuración. Este archivo será la combinación de la plantilla Jinja2 y los valores almacenados en el archivo YAML previamente creado (Figura 21).

Por lo tanto, en caso de haber detectado alguna modificación, se crearían las dos listas de prefijos en el router:

```
[edit policy-options]
prefix-list eBGP-AS20815 {
    80.64.32.0/20
    195.93.170.0/23
    185.29.212.0/22
    91.223.143.0/24
    91.224.150.0/23
}
```

```
[edit policy-options]
prefix-list eBGP6-AS20815 {
    2a03:8040::/32
    2a00:5c00::/32
}
```

Figura 23: Lista de prefijos IPv4 e IPv6

Las políticas de filtrado correspondientes a cada una de las dos listas serían:

```
[edit policy-options]
policy-statement bgp-eBGP-AS20815-in{
    term1 {
        from {
            prefix-list-filter eBGP-AS20815 orlonger;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
}
```

Figura 24: Política de filtrado IPv4

```
[edit policy-options]
policy-statement bgp-eBGP6-AS20815-in{
    term1 {
        from {
            prefix-list-filter eBGP6-AS20815 orlonger;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
}
```

Figura 25: Política de filtrado IPv6

Como se puede observar en las dos figuras, el nombre que está marcado en negrita al lado de la palabra reservada `prefix-list-filter` corresponde a los nombres de las dos listas de prefijos, Figura 23. El otro nombre que está destacado corresponde al nombre que se le denomina a la política que se acaba de crear. Por último, se le aplican las políticas de filtrado a los dos vecinos respectivamente mediante la palabra reservada `import` como se puede observar en las siguientes figuras. De esta forma, el vecino al que se le aplique la política solamente aceptará información recibida por los prefijos que forman la lista `eBGP-AS20815` por ejemplo y rechazará todos los demás.

```
[edit protocols bgp group eBGP-Peering-Espanix1]

  neighbor 193.149.1.7{
    description GRN;
    import bgp-eBGP-AS20815-in;
    peer-as 20815;
  }
```

*Figura 26: Filtro aplicado en vecino IPv4*

```
[edit protocols bgp group eBGP6-Peering-Espanix1]

  neighbor 193.149.1.7{
    description GRN;
    import bgp-eBGP6-AS20815-in;
    peer-as 20815;
  }
```

*Figura 27: Filtro aplicado en vecino IPv6*

## 6.2.8 Optimizador router

Este bloque ha sido desarrollado con el objetivo de optimizar el rendimiento de los routers. Para ello se han utilizado varios filtros estáticos de filtrado común para todos los vecinos.

- **Bogon ASN filtering:** los ASN privados o reservados no tienen lugar en el DFZ (Default Free Zone) público. El DFZ ayuda a amortiguar la exposición accidental de dispositivos de rutado interno. Es decir, un bogon puede ser cualquier recurso de Internet que de acuerdo con la autoridad de registro no debe aparecer en ninguna red.

```

policy-options {
  as-path-group bogon-asns {
    /* RFC7607 */
    as-path zero ".* 0 .*";
    /* RFC 4893 AS_TRANS */
    as-path as_trans ".* 23456 .*";
    /* RFC 5398 and documentation/example ASNs */
    as-path examples1 ".* [64496-64511] .*";
    as-path examples2 ".* [65536-65551] .*";
    /* RFC 6996 Private ASNs*/
    as-path reserved1 ".* [64512-65534] .*";
    as-path reserved2 ".* [4200000000-4294967294]
.*";
    /* RFC 6996 Last 16 and 32 bit ASNs */
    as-path last16 ".* 65535 .*";
    as-path last32 ".* 4294967295 .*";
    /* RFC IANA reserved ASNs*/
    as-path iana-reserved ".* [65552-131071] .*";
  }
  policy-statement import_from_ebgp {
    term bogon-asns {
      from as-path-group bogon-asns;
      then reject;
    }
  }
}

```

*Figura 28: Bogon ASN filter*

Una buena conducta sería rechazar todas las rutas EBGp que contengan un ASN Bogon en cualquier lugar del AS\_PATH.

- **Bogon Prefix filtering:** existen prefijos que deberían ser filtrados ya que la IETF no tiene la intención de que estos sean rutados a la red pública.

```

policy-options {
  prefix-list BOGONS_v4 {
    0.0.0.0/8;
    10.0.0.0/8;
    100.64.0.0/10;
    127.0.0.0/8;
    169.254.0.0/16;
    172.16.0.0/12;
    192.0.2.0/24;
    192.88.99.0/24;
    192.168.0.0/16;
    198.18.0.0/15;
    198.51.100.0/24;
    203.0.113.0/24;
    224.0.0.0/4;
    240.0.0.0/4;
  }
  policy-statement BGP_FILTER_IN {
    term IPv4 {
      from {
        prefix-list BOGONS_v4;
      }
      then reject;
    }
  }
}

```

*Figura 29: Bogon Prefix filter*

- **No small prefix filtering:** cualquier configuración de BGP debe incluir un filtro de pequeños prefijos. Esto evita “hijacks” dirigidos a las redes /32 o prefijos menores. La mayoría de los prefijos pequeños que se ven en una comunicación de Internet son fugas incorrectas debido a una mala configuración o a ingeniería de tráfico.

Mediante el uso de estos filtros, no se perderá información ya que, por lo general, se podrán ver los prefijos más grandes anunciados por el mismo IXP o fuente de tránsito.

Es cierto que existen redes IP pequeñas como un /29 o /28, pero son muy pocas. Es importante tener en cuenta que estas redes serían también filtradas. Sin embargo, la escasez de espacio de direcciones de IPv4 no es razón suficiente para no hacer uso de este tipo de filtros.

Por lo tanto, no deben esperarse que rutas menores que un /24 para IPv4 o un /48 en IPv6 tengan un enrutamiento global que funcione.

- **Filter Long AS Paths:** algunas redes sobrepasan exageradamente el número de ASs que se anteponen a ellos. Esto posibilita el tipo de ataque malicioso que ya fue descubierto hace muchos años.

Al momento de hacer este proyecto se conocen AS\_Path de como máximo 40 ASN. Por lo tanto, un número seguro en el filtro estaría en 100 AS en el AS\_Path.

```

policy-options {
  policy-statement bgp-import-policy {
    term no-transit-leaks {
      from as-path no-transit-import-in;
      then reject;
    }
  }
}
as-path no-transit-import-in ".*
(174|209|701|702|1239|1299|2914|3257|3320|3356|3549|3561
|4134|5511|6453|6461|6762|7018) .*";

```

*Figura 30: Long AS Paths filter*

- **Filter Known transit Networks in AS Paths:** a través de un IXP, las redes Tier 2 y Tier 3 no deben anunciar prefijos con una red de tránsito en el AS\_Path ya que probablemente no sea uno de sus clientes. También por la misma razón, no se debería aceptar a ninguno de sus clientes a través de uno de sus clientes. Resumiendo, filtra la lista de los ASs Tier 1.

### 6.2.9 Corrector de errores

Este módulo corresponde a un programa secundario independiente al que se ha estado describiendo hasta ahora. Este iría situado en el terminal de un operario. Tiene el objetivo de facilitarle el trabajo al empleado encargado de actualizar las fuentes de información.

El operario recibirá un correo cuando el programa principal detecte alguna anomalía. Entonces, este ejecutará el segundo programa con el AS a corregir como argumento y se le indicará cómo debería quedar configurado el router. Si, además, al comando de ejecución se le añade un "-s", el programa le indicará exactamente qué comandos debe introducir en la consola del router correspondiente para solucionar el fallo. Se pueden observar los resultados de las dos opciones en las siguientes imágenes:

```

policy-options {
  prefix-list eBGP-AS20815{
    185.29.212.0/22;
    195.93.170.0/23;
    80.64.32.0/20;
    91.223.143.0/24;
    91.224.150.0/23;
  }
  prefix-list eBGP6-AS20815{
    2a00:5c00::/32;
    2a03:8040::/32;
  }
  policy-statement bgp-eBGP-AS20815-in {
    term 1 {
      from {
        prefix-list-filter eBGP-AS20815 orlonger;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
  policy-statement bgp-eBGP6-AS20815-in {
    term 1 {
      from {
        prefix-list-filter eBGP6-AS20815
        orlonger;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
}

```

Figura 31: NuevoAS.py

```
set policy-options prefix-list eBGP-AS20815 185.29.212.0/22
set policy-options prefix-list eBGP-AS20815 195.93.170.0/23
set policy-options prefix-list eBGP-AS20815 80.64.32.0/20
set policy-options prefix-list eBGP-AS20815 91.223.143.0/24
set policy-options prefix-list eBGP-AS20815 91.224.150.0/23
set policy-options prefix-list eBGP6-AS20815 2a00:5c00::/32
set policy-options prefix-list eBGP6-AS20815 2a03:8040::/32
set policy-options policy-statement bgp-eBGP-AS20815-in term 1
from prefix-list-filter eBGP-AS20815 orlonger
set policy-options policy-statement bgp-eBGP-AS20815-in term 1
then accept
set policy-options policy-statement bgp-eBGP-AS20815-in term 2
then reject
set policy-options policy-statement bgp-eBGP6-AS20815-in term 1
from prefix-list-filter eBGP6-AS20815 orlonger
set policy-options policy-statement bgp-eBGP6-AS20815-in term 1
then accept
set policy-options policy-statement bgp-eBGP6-AS20815-in term 2
then reject
```

*Figura 32: NuevoAS.py -s*

## 6.2.10 Temporización

Los cambios de información de los vecinos BGP son frecuentes, pero no tanto como para diseñar un programa que esté comprobando dichos cambios constantemente. Es por eso que la aplicación debe estar programada para ejecutarse periódicamente. Con dos ejecuciones diarias sería más que suficiente, ya que la frecuencia con la que los vecinos modifican la información no es tan alta a día de hoy.

La función de este módulo es calcular el tiempo que se ha configurado entre ejecuciones (medio día en este caso) y enviarle una señal al CORE de la aplicación para que inicie la ejecución del programa. Para ello se utilizará una aplicación denominada cron. Cron es un administrador regular de procesos en segundo plano (demonio) que ejecuta procesos a predeterminados intervalos de tiempo (por ejemplo, cada día, semana o mes). Cron es un demonio, lo que significa que solo requiere ser iniciado una vez, generalmente con el mismo arranque del sistema. El servicio de cron se llama crond. Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero crontab [15].

Crontab simplemente es un archivo de texto que almacena una lista de comandos a ejecutar en un tiempo especificado por el usuario. Crontab verificará la fecha y hora en que se debe ejecutar el script o el comando, los permisos de ejecución y lo realizará en un segundo plano.



### 6.3 Funcionamiento del sistema

El sistema arrancará cuando haya pasado el tiempo predeterminado en cron. Este ejecutará el programa e iniciará el Core del sistema. Primero activará el módulo de la Base de datos para consultar las tres fuentes de información que se disponen. Por lo tanto, se obtendrán tres listas de vecinos BGP. Entonces, el CORE llamará al módulo Transmisor de alarmas para que verifique si existe alguna diferencia entre ellas. Si es así, se le enviará una alarma al operario indicándole la inconsistencia. Si no, el programa seguirá ejecutándose.

El siguiente paso es obtener la lista de prefijos que se reciben de cada vecino y para ello el Core arrancará el módulo Búsquedas y en función de la información obtenida variará las consultas. Una vez obtenida la lista, se optimizarán los prefijos de red gracias al módulo Optimizador de prefijos.

Puede que ningún vecino haya modificado nada en sus sistemas y la lista de prefijos que se haya obtenido haya sido la misma que la obtenida en la anterior ejecución del programa. Para verificarlo, el CORE iniciará el módulo Comparador. Si no ha habido ninguna modificación, el programa se detendrá aquí hasta la siguiente ejecución de cron. Si el módulo ha detectado algún cambio, le transmitirá dicha información al router del ISP. Esto se realizará mediante el módulo Comunicación.

El módulo Optimizador router también se sitúa en el terminal, pero a diferencia de los demás, solo se ejecuta una única vez, cuando se ejecute por primera vez el programa. Este módulo es el encargado de optimizar el router y para ello utiliza filtros estáticos. Es por eso que no necesita que se ejecute más de una vez en el mismo router.

Por último, el módulo Corrector de errores se encuentra en los terminales de los operarios encargados de arreglar las inconsistencias. Como previamente se ha descrito, el módulo Transmisor de alarmas comunicará los fallos a los operarios y estos con la ayuda de este módulo los solucionarán de una manera rápida y eficaz.

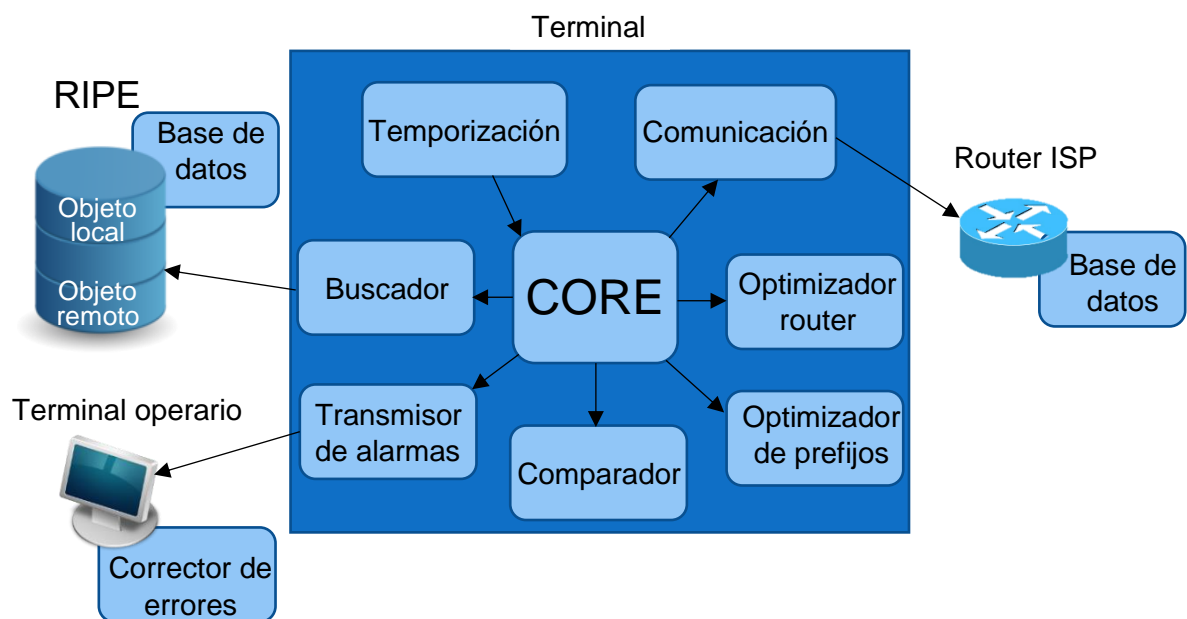


Figura 33: Funcionamiento del sistema

# 7 METODOLOGÍA

## 7.1 Descripción de tareas

En este apartado se describen las etapas en las que se ha dividido el proyecto:

- **Definición del proyecto:** en esta primera parte se definen los objetivos del proyecto, las especificaciones y la planificación que se necesita para estructurar de manera adecuada las tareas según el tiempo y recursos necesarios.
- **Diseño de la solución:** esta es la parte más importante del proyecto. Aquí se busca encontrar una solución a la problemática que se plantea y se diseñan los módulos y elementos que componen dicha solución.
- **Desarrollo de la solución:** en esta fase se desarrolla el proyecto. Aquí se implementa la arquitectura física y la herramienta que contiene la solución.
- **Diseño y realización de pruebas para validación de solución:** en esta parte se realizan diferentes pruebas para verificar el correcto funcionamiento del sistema. Además, es común encontrar nuevas vías que mejoran la solución o nuevos problemas que no se hayan previsto previamente.
- **Gestión y documentación:** esta fase dura todo el proyecto. Consiste en un seguimiento activo de la situación del proyecto e incluye todas las actividades relacionadas con la documentación, como recoger los fundamentos teóricos que ayudará el tenerlos documentados en fases posteriores o recoger los resultados finales del trabajo.

## 7.2 Planificación

En este apartado se describen las tareas del proyecto. Todo ello viene recogido en una tabla donde también se pueden observar la duración o la fecha de inicio de dichas tareas.

Código <sup>1</sup>	Nombre de tarea	Duración	Comienzo	Fin
<b>1</b>	<b>Definición del proyecto</b>	<b>7 días</b>	<b>vie 1/2/19</b>	<b>jue 7/2/19</b>
T.1.1	Definición de objetivos y alcance	2 días	vie 1/2/19	sáb 2/2/19
T.1.2	Definición de especificaciones	2 días	dom 3/2/19	lun 4/2/19
T.1.3	Definición de tareas	3 días	mar 5/2/19	jue 7/2/19
H.1.4	Definición del proyecto terminada	0 días	jue 8/2/19	jue 8/2/19
<b>2</b>	<b>Diseño de la solución</b>	<b>60 días</b>	<b>vie 8/2/19</b>	<b>mié 8/4/16</b>
T.2.1	Diseño de la arquitectura de red	20 días	vie 8/2/19	mié 27/2/19

<sup>1</sup> T = las tareas a realizar

H = los hitos que marcan el término de cada tarea

T.2.2	Diseño de los módulos	40 días	jue 28/2/19	lun 8/4/19
H.2.3	Diseño de la solución completa	0 días	lun 8/4/19	lun 8/4/19
<b>3</b>	<b>Desarrollo de la solución</b>	<b>30 días</b>	<b>mar 9/4/19</b>	<b>mié 8/5/19</b>
<b>T.3.1</b>	<b>Entorno de desarrollo</b>	<b>13 días</b>	<b>mar 9/4/19</b>	<b>dom 21/4/19</b>
T.3.1.1	Configuración del entorno	3 días	mar 9/4/19	jue 11/4/19
T.3.1.2	Despliegue de la maqueta de red	10 días	vie 12/4/19	dom 21/4/19
H.3.1.3	La maqueta de red funciona	0 días	dom 21/4/19	dom 21/4/19
<b>T.3.2</b>	<b>Módulos</b>	<b>12 días</b>	<b>lun 22/4/19</b>	<b>vie 23/5/19</b>
T.3.2.1	Base de datos	1 días	lun 22/4/19	lun 22/4/19
T.3.2.2	Transmisor de alarmas	1 días	mar 23/4/19	mar 23/4/19
T.3.2.3	CORE	4 días	mié 24/4/19	sáb 27/4/19
T.3.2.4	Buscador	1 días	dom 28/4/19	dom 28/4/19
T.3.2.5	Optimizador de prefijos	1 días	lun 29/4/19	lun 29/4/19
T.3.2.6	Comparador	1 días	mar 30/4/19	mar 30/4/19
T.3.2.7	Optimizador router	1 días	mié 1/5/19	mié 1/5/19
T.3.2.8	Corrector de errores	1 días	jue 2/5/19	jue 2/5/19
T.3.2.9	Temporización	1 días	vie 3/5/19	vie 3/5/19
H.3.3	Solución desarrollada	0 días	vie 3/5/19	vie 3/5/19
<b>T.4.4</b>	<b>Interconexión sistema completo</b>	<b>5 días</b>	<b>sáb 4/5/19</b>	<b>mié 8/5/19</b>
T.4.4.1	Interconexión	5 días	sáb 4/5/19	mié 8/5/19
H.4.5	Solución desarrollada	0 días	mié 8/5/19	mié 8/5/19
<b>4</b>	<b>Diseño y realización de pruebas para validación de solución</b>	<b>25 días</b>	<b>mié 8/5/19</b>	<b>sáb 1/6/19</b>
<b>T.4.1</b>	<b>Entorno de pruebas</b>	<b>10 días</b>	<b>mié 8/5/19</b>	<b>vie 17/5/19</b>

T.4.1.1	Configuración del entorno	5 días	mié 8/5/19	dom 12/5/19
T.4.1.2	Despliegue de la maqueta de red	5 días	lun 13/5/19	vie 17/5/19
H.4.1.3	La maqueta de red funciona	0 días	vie 17/5/19	vie 17/5/19
<b>T.4.4</b>	<b>Realización de pruebas</b>	<b>15 días</b>	<b>sáb 18/5/19</b>	<b>sáb 1/6/19</b>
T.4.4.1	Pruebas	15 días	sáb 18/5/19	sáb 1/6/19
H.4.5	Solución desarrollada	0 días	sáb 1/6/19	sáb 1/6/19
<b>5</b>	<b>Gestión y documentación</b>	<b>135 días</b>	<b>vie 1/2/19</b>	<b>sáb 15/6/16</b>
T.5.1	Memoria TFG	135 días	vie 1/2/19	sáb 15/6/19
H.5.2	Documentación completada	0 días	sáb 15/6/19	sáb 15/6/19

*Tabla 5: Planificación*

La figura que se incluye a continuación consiste en el diagrama de Gantt que es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto de las diferentes tareas o actividades a lo largo de un tiempo total determinado.

### 7.3 Diagrama de Gantt

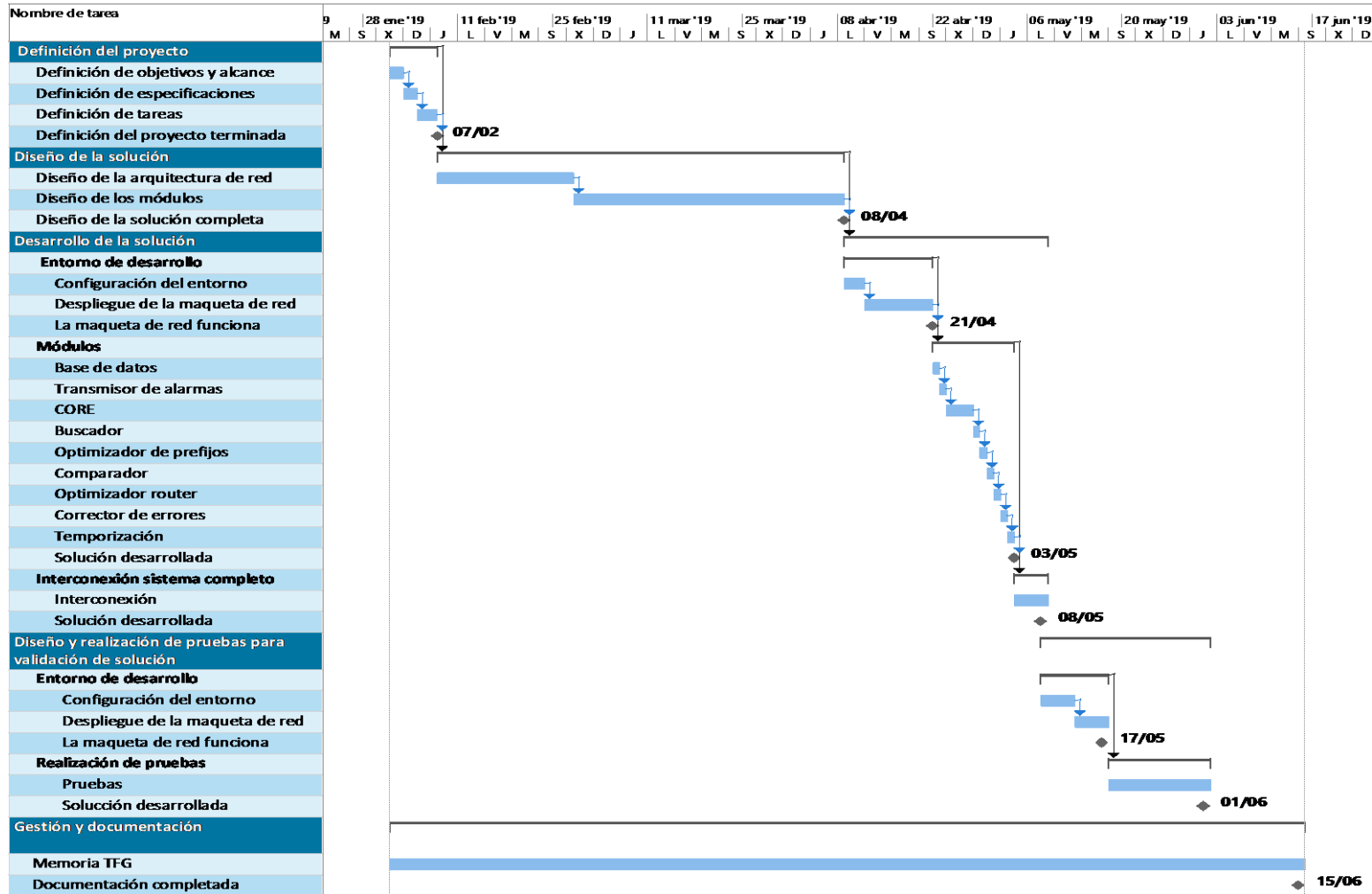


Figura 34: Diagrama de Gantt

## 8 RESUMEN DE COSTES

A continuación, se procede a incluir un análisis de los costes y del presupuesto del proyecto. Atendiendo a las características del mismo, se indica el análisis de costes de la fase realizada hasta ahora, por un ingeniero de telecomunicaciones que haya llevado a cabo la elaboración de un proyecto de ingeniería y su documentación.

Como se ha indicado, se adjunta una tabla con los gastos de la realización del proyecto, pero con los costes de mano de obra equivalentes a un proyecto en una organización. La duración del trabajo ha sido de 5 meses, comenzando en febrero y terminando a finales de junio. El tiempo en horas que se ha calculado ha sido una estimación equivalente al valor de créditos ECTS correspondientes al Trabajo Fin de Grado.

### 8.1 Horas internas

En este apartado se describen las horas de trabajo de cada miembro del grupo y sus correspondientes costes.

Concepto	Nº horas	Coste horario (€/h)	Coste total (€)
Ingeniero técnico	300	25	7 500
Ingeniero sénior	60	50	3 000
<b>SUBTOTAL HORAS INTERNAS</b>			10 500

Tabla 6: Partida de horas internas

### 8.2 Amortizaciones

En esta sección se detallan los recursos que han sido necesarios para llevar a cabo el proyecto.

Concepto	Coste adquisición (€)	Vida útil <sup>2</sup>	Uso en el proyecto (h)	Coste total (€)
PC 2018 (i7 12GB 4TB)	900	5 años	300	30,68
PC 2018 (i7 8GB 2TB)	700	5 años	60	4,77
Router Juniper mx104	32 000	5 años	100	363,63
<b>SUBTOTAL AMORTIZACIONES</b>				399,08

Tabla 7: Partida de amortizaciones

<sup>2</sup> A la hora de calcular el uso en el proyecto frente a la vida útil se ha considerado que un año tiene 1760 horas laborables

### 8.3 Gastos

En este apartado se clasifican los gastos básicos que cualquier oficina de trabajo puede tener.

Concepto	Coste (€)
Conexión a Internet	200
Material de oficina	30
<b>SUBTOTAL GASTOS</b>	<b>230</b>

Tabla 8: Partida de gastos

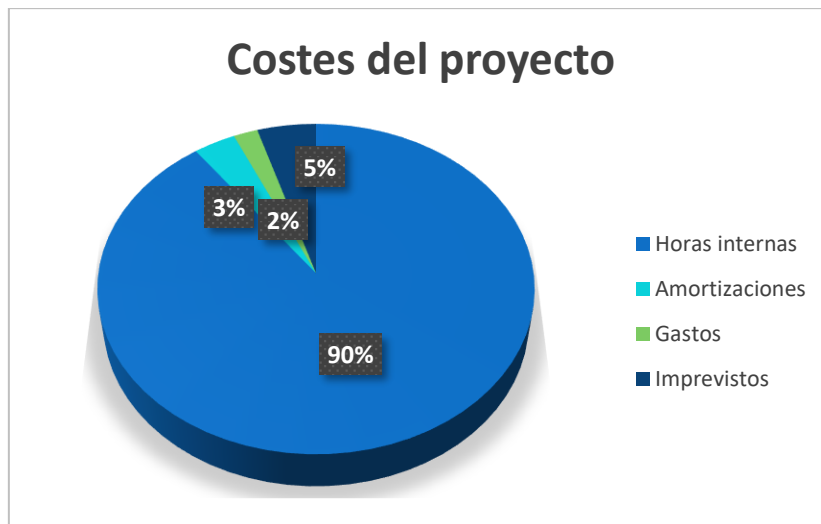
### 8.4 Resumen de costes

La mayoría de los recursos económicos utilizados para la realización del proyecto han sido para hacer frente a los gastos en recursos humanos. Es un proyecto que requiere de un número elevado de horas de trabajo que han sido invertidas sobre todo en la etapa del desarrollo de la solución. Por otra parte, se encuentran los costes de amortización que son significativamente pequeños ya que la utilización de software libre ha contribuido en la disminución de los costes de amortización. El elemento con mayor impacto en este apartado ha sido el router Juniper mx104. Por último, el resto de gastos presentados se corresponden a los gastos básicos que cualquier oficina de trabajo tiene.

Concepto	Coste (€)
<b>Horas internas</b>	<b>10 500</b>
<b>Amortizaciones</b>	<b>399,08</b>
<b>Gastos</b>	<b>230</b>
<b>SUBTOTAL</b>	<b>11 129,08</b>
<b>Costes indirectos e imprevistos (5 %)</b>	<b>556,45</b>
<b>TOTAL</b>	<b>11 685,53</b>

Tabla 9: Resumen de costes

Para observar mejor el impacto de cada coste en el cómputo total de gastos se incluye el siguiente gráfico:



*Figura 35: Comparativa de partidas del coste total del proyecto*

Por lo tanto, el coste total del proyecto (sin incluir el IVA) asciende a 11 685,53€.



## 9 ANÁLISIS DE RIESGOS

El propósito de este apartado es detectar y evaluar los posibles sucesos que podrían impactar de forma negativa en el desarrollo del proyecto. Esto es interesante porque permite analizar las posibles causas y las consecuencias que conllevan y poder reducir significativamente el valor de este proyecto.

### 9.1 Identificación de riesgos

En esta sección se recogen los riesgos identificados durante la fase de planificación del proyecto. Es importante clarificar que la probabilidad de que sucedan es muy baja ya que el escenario de desarrollo del proyecto no es muy grande, está bien definido y porque se trabaja con tecnologías y equipamiento afianzado en el mundo tecnológico. Los riesgos que se han identificado son los siguientes:

#### 9.1.1 Incompatibilidad parcial o completa (R1)

En el desarrollo de este proyecto se han utilizado varias tecnologías como el framework PyEZ o las bases de datos públicas de RIPE. En el momento que se diseñó este proyecto, estas tecnologías han sido la mejor vía para llegar hasta la solución más óptima. Sin embargo, no es posible saber con certeza si en un futuro seguirán estando disponibles o si sufrirán alguna modificación que pueda causar la inutilización de este proyecto. Las bases de datos pueden cambiar toda su organización o puede quedar inhabilitado el framework en los router Juniper.

Por otra parte, existe la posibilidad de que otras tecnologías superiores aparezcan y sustituyan a las actuales. Una nueva organización puede diseñar una base de datos que tenga más aceptación en el mercado que la que tiene RIPE y esta caer en una completa desactualización y en desuso. Lo mismo puede suceder para el caso de PyEZ. En este caso, el proyecto tendría que sufrir cambios considerables para que siga funcionando correctamente. Sin embargo, este riesgo está contenido como se verá después ya que se trabaja con tecnologías maduras y equipos con un gran nivel de calidad.

#### 9.1.2 Desviaciones en la planificación (R2)

Durante el proceso de desarrollo del proyecto es probable que se produzcan retrasos. Es normal que sucedan imprevistos y problemas no esperados en la planificación que conlleven al no cumplimiento de las fechas previstas en la planificación y a un desequilibrio en el reparto de tareas. Sin embargo, no es muy alto su impacto.

#### 9.1.3 Desvíos en el presupuesto (R3)

A medida que se desarrolla el proyecto, por diversos motivos, la organización decide tomar medidas que se superen el presupuesto inicial. La probabilidad de que esto ocurra es baja pero el impacto que tendría este riesgo es generalmente alto. La razón es que, sin dinero para un proyecto, difícilmente puede este evolucionar y seguir hacia adelante.

#### 9.1.4 Fallos en el equipamiento (R4)

Los fallos en el equipamiento suelen ser uno de los problemas más habituales. En cualquier momento los equipos pueden fallar o simplemente que se vaya la luz de las instalaciones, causando la pérdida total o parcial del proyecto. Aunque la probabilidad de que esto ocurra es baja ya que el escenario de este proyecto es muy concreto, el impacto sería muy alto.

## 9.2 Análisis de riesgos

En esta sección se realizará un análisis cualitativo-cuantitativo de los riesgos identificados y clasificarlos según su prioridad. Para ello se han valorado dos aspectos: la probabilidad de que ocurra y el impacto que supondría que ocurriese.

ID	Riesgo	Causa	Probabilidad	Impacto	Prioridad
R1	<b>Incompatibilidad parcial o completa</b>	Modificaciones en versiones futuras	Difícil (0,1)	Alto (0,9)	Baja (0,09)
R2	<b>Desviaciones en la planificación</b>	Superar / no alcanzar las fechas previstas	Media (0,5)	Bajo (0,2)	Baja (0,10)
R3	<b>Desvíos en el presupuesto</b>	Mala previsión de los costes	Difícil (0,2)	Alto (0,8)	Moderada (0,16)
R4	<b>Fallos en el equipamiento</b>	Problemas técnicos	Difícil (0,2)	Alto (0,9)	Moderada (0,18)

Tabla 10: Análisis de riesgos

Para representar estos valores de manera gráfica se ha utilizado una herramienta denominada matriz de probabilidad-impacto. Esta matriz contrasta la probabilidad y el impacto de cada riesgo para determinar su prioridad.

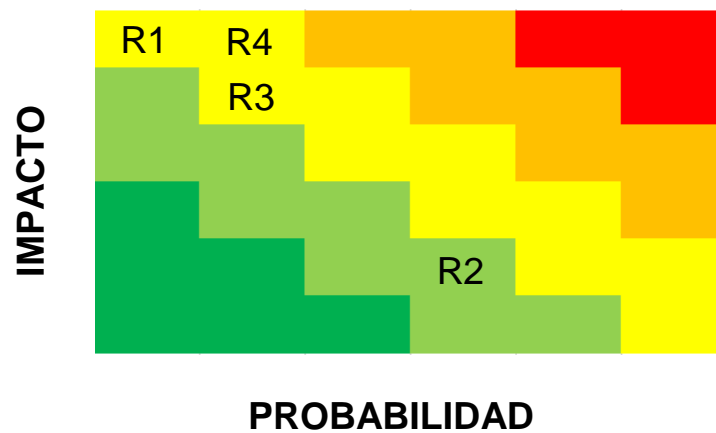


Figura 36: Matriz probabilidad-impacto

### 9.3 Planificación de la respuesta

Los planes de contingencia se estudian para cada uno de los riesgos identificados y tienen como objetivo prevenir el riesgo, reducir la probabilidad de que ocurra o simplemente buscar soluciones rápidas en caso de que ocurra.

El primer riesgo identificado (R1) es el más complicado de afrontar, ya que es imposible conocer el devenir de la tecnología ni el estado del mercado en un futuro. Es por ello que no hay más remedio que afrontar este riesgo y diseñar un plan de acción para cuando esto suceda. Para reducir el impacto, se ha optado por un diseño modular para que en caso de tener que modificar algo, la mayor parte de la solución sea reutilizable. Por otra parte, se llevará un seguimiento del mercado y de las tecnologías de cada momento para anticiparse y ganar un mayor tiempo de maniobra.

Para minimizar el riesgo de desviaciones en la planificación (R2) se puede utilizar un sistema de planes de control y entregas como, por ejemplo, estructurar el desarrollo total del proyecto en tareas menores con fechas exactas de entrega. Esto ayudaría también a detectar imprecisiones temporales lo antes posible.

En cuanto al tema relacionado con el presupuesto y la economía (R3), una de las soluciones más simples es incluir una partida de imprevistos en el presupuesto general con la suficiente holgura. Esto reduciría el riesgo de que se supere el presupuesto previsto.

Finalmente, el riesgo de tener fallos en el equipamiento (R4) siempre va a existir. Sin embargo, se puede reducir la probabilidad de que ocurra utilizando equipamiento de gran calidad y en caso de que ocurra se podría reducir su impacto guardando los cambios realizados en el proyecto frecuentemente y teniendo varias copias de seguridad almacenadas en diferentes lugares geográficos.

## 10 CONCLUSIONES

La conclusión más importante que se ha obtenido con este Trabajo Fin de Grado es que se han alcanzado los objetivos propuestos al principio de este proyecto. Se ha diseñado un software que soluciona la problemática actual de la empresa Sarenet pero perfectamente puede ser de gran utilidad para diferentes organizaciones. El resultado de este proyecto ha sido satisfactorio.

La interconexión entre Sistemas Autónomos es un área muy extensa y reconocida en el mundo tecnológico donde existe gran interés por parte de empresas y organizaciones del ámbito de las telecomunicaciones. Es por ello la necesidad y la gran relevancia que tienen proyectos como este.

Ha quedado claro que todavía queda mucho trabajo que hacer en este tema muy relacionado con el protocolo BGP. Este proyecto implementa las bases y es una ayuda para lograr la mejora entre las relaciones entre Sistemas Autónomos en Internet, pero es posible ir aún más allá y diseñar sistemas automatizados de verificación y de mejora de la seguridad de las empresas.

En resumen, este Trabajo Fin de Grado aborda uno de los problemas que tienen muchas empresas de telecomunicaciones que es la falta de monitorización y control en los enlaces de tipo peering. Mediante la solución propuesta, se mitigan todas las cuestiones y se desarrolla una herramienta que monitoriza de forma constante el tráfico de este tipo de enlaces y de manera automática los controla y actualiza según la información que recibe de otras empresas.

# 11 BIBLIOGRAFÍA

- [1] James F. Kurose y Keith W. Ross, << *Redes de computadoras: un enfoque descendente*>> 2010, 5ª edición, capítulo 1
- [2] <<*INTERNET USAGE STATISTICS*>> 2019. [En línea]. Disponible en: <https://www.internetworldstats.com/stats.htm> . [Último acceso: 5 junio 2019].
- [3] [En línea]. Disponible en: <https://www.ripe.net/publications/docs/ripe-525> . [Último acceso: 5 junio 2019].
- [4] [En línea]. Disponible en: <https://tools.ietf.org/html/rfc4271> . [Último acceso: 5 junio 2019].
- [5] <<*RIPE Database Query*>> [En línea]. Disponible en: <https://apps.db.ripe.net/db-web-ui/#/query>. [Último acceso: 5 junio 2019].
- [6] <<*Whois Client*>> 2016. [En línea]. Disponible en: <https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation/how-to-query-the-ripe-database/14-4-command-line-queries/14-4-3-whois-client>. [Último acceso: 5 junio 2019].
- [7] <<*RESTful API Queries*>> 2016. [En línea]. Disponible en: <https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation/how-to-query-the-ripe-database/14-3-restful-api-queries>. [Último acceso: 5 junio 2019].
- [8] Eric Rescorla, <<*HTTP Over TLS*>> [En línea]. Disponible en: <https://tools.ietf.org/html/rfc2818>. [Último acceso: 5 junio 2019].
- [9] <<*Python 3.7.3 documentation*>> 2019. [En línea]. Disponible en: <https://docs.python.org/3/>. [Último acceso: 5 junio 2019].
- [10] <<*pyipcalc 3.0.2*>> 2018. [En línea]. Disponible en: <https://pypi.org/project/pyipcalc/>. [Último acceso: 5 junio 2019].
- [11] Kannan Varadhan, <<*Supernetting: an Address Assignment and Aggregation Strategy*>> [En línea]. Disponible en: <https://tools.ietf.org/html/rfc1338>. [Último acceso: 5 junio 2019].
- [12] <<*YAML Ain't Markup Language*>> 2009, 3ª edición. [En línea]. Disponible en: <https://yaml.org/spec/1.2/spec.html>. [Último acceso: 5 junio 2019].
- [13] <<*Junos PyEZ Developer Guide*>> 2019. [En línea]. Disponible en: [https://www.juniper.net/documentation/en\\_US/junos-pyez/information-products/pathway-pages/junos-pyez-developer-guide.html](https://www.juniper.net/documentation/en_US/junos-pyez/information-products/pathway-pages/junos-pyez-developer-guide.html). [Último acceso: 5 junio 2019].
- [14] Armin Ronacher, <<*Jinja2 Documentation*>> [En línea]. Disponible en: <http://jinja.pocoo.org/docs/2.10/>. [Último acceso: 5 junio 2019].
- [15] Paul Vixie, <<*CRONTAB*>> 2012. [En línea]. Disponible en: <http://man7.org/linux/man-pages/man5/crontab.5.html>. [Último acceso: 5 junio 2019].
- [16] Quaizer Vohra, <<*BGP support for Four-octet AS Number Space*>> 2007. [En línea]. Disponible en: <https://tools.ietf.org/html/rfc4893>. [Último acceso: 5 junio 2019].

- [17] <<*LACNIC*>> [En línea]. Disponible en: <https://www.lacnic.net/544/1/lacnic/>. [Último acceso: 5 junio 2019].
- [18] Yakov Rekhter, Tony Li y Susan Hares<<*A Border Gateway Protocol 4 (BGP-4)*>> 2006. [En línea]. Disponible en: <https://tools.ietf.org/html/rfc4271>. [Último acceso: 5 junio 2019].
- [19] <<*RIPE NCC*>> [En línea]. Disponible en: <https://www.ripe.net/manage-ips-and-asns/>. [Último acceso: 5 junio 2019].
- [20] Alexandre Snarskii <<*bgpq3*>> 2018. [En línea]. Disponible en: <https://github.com/snar/bgpq3>. [Último acceso: 5 junio 2019].
- [21] Sean Sawtell, (2018). *Day One: Automating Junos with Ansible*, 2nd Edition
- [22] Rob Enns, Martin Bjorklund, Juergen Schoenwaelder y Andy Bierman, <<*Network Configuration Protocol (NETCONF)*>> 2011. [En línea]. Disponible en: <https://tools.ietf.org/html/rfc6241>. [Último acceso: 5 junio 2019].
- [23] Karim Okasha <<*Network Automation and the Rise of NETCONF*>> 2017. [En línea]. Disponible en: <https://medium.com/@k.okasha/network-automation-and-the-rise-of-netconf-e96cc33fe28>. [Último acceso: 5 junio 2019].

# ANEXO I: FUNDAMENTOS TEÓRICOS

## 1 Estructura de Internet y Sistemas Autónomos

El rutado de Internet está estructurado en dos niveles jerárquicos. Por un lado, está dividido en dominios cuyo rutado se establece mediante protocolos de rutado interior (IGP) los cuales tienen total conocimiento del dominio. Ejemplos de este tipo de protocolos son Open Shortest Path First (OSPF), Routing Information Protocol (RIP) y Enhanced Interior Gateway Routing Protocol (EIGRP).

Por otro lado, existen los protocolos interdominio o de rutado exterior (EGP), los cuales describen la comunicación entre diferentes dominios. Hoy en día prácticamente el único que se utiliza es BGP (Border Gateway Protocol).

Cada dominio de rutado es un único dominio administrativo, el cual es independiente uno de otro y trabaja con políticas diferentes. Se puede decir que el dominio es autónomo y es por eso por lo que se denomina Sistema Autónomo (AS). Un Sistema Autónomo se define como un grupo de redes IP que es gestionado por uno o más operadores de red que poseen una clara y propia política de ruteo. Cada sistema autónomo tiene un número como identificador que lo utilizará para intercambiar información con otros ASs. Este número fue de 16 bits hasta el año 2007 pero debido al aumento de la demanda, en la RFC 4893 [16] se extendió a 32 bits.

Número ASN	Bits	Descripción	Referencia
0	16	Reservado	[RFC1930]
1 - 23455	16	ASNs público	
23456	16	Reservado para transición de pool AS	[RFC6793]
23457 - 64534	16	ASNs público	
64000 - 64495	16	Reservado por IANA	
64496 - 64511	16	Reservado para uso en documentación y código de ejemplo	[RFC5398]
64512 - 65534	16	ASN para uso privado	
65535	16	Reservado	
65536 - 65551	32	Reservado para uso en documentación y código de ejemplo	[RFC4893][RFC5398]
65552 - 131071	32	Reservado	
131072 - 4199999999	32	ASNs públicos de 32-bit	
4200000000 - 4294967294	32	ASNs para uso privado	[RFC6996]
4294967295	32	Reservado	

Tabla 11: Lista de ASNs

Los números de los Sistemas Autónomos son distribuidos en bloques por la IANA (Internet Assigned Numbers Authority) [17] a los RIRs (Registros Regionales de Internet). Estos últimos asignarán un número a cada Sistema Autónomo conforme vayan enviando solicitudes. Existen cinco Registros Regionales de Internet:

- **AFRINIC:** África
- **APNIC:** Asia y Pacífico
- **ARIN:** Norte América
- **LACNIC:** Latino América y Caribe
- **RIPE NCC:** Europa, Oriente Medio y Asia Central



*Figura 37: Registros Regionales de Internet*

## 1.1 Tipos de Sistemas Autónomos

Entre los tipos de Sistemas Autónomos se suelen categorizar en diversos tipos. A continuación, se describen brevemente los tres tipos más significativos.

- **Stub:** AS que alcanza redes de otros ASs a través de un solo punto de salida.
- **Multi-homed non-transit:** AS que alcanza redes de otros ASs a través de más de un punto de salida y no admiten tránsito de tráfico entre ellos.
- **Multi-homed transit:** AS que alcanza redes de otros ASs a través de más de un punto de salida y permite que se comuniquen entre ellos.

Como la mayoría de los Sistemas Autónomos no están conectados unos a otros, necesitan transmitir el tráfico por diferentes ASs hasta llegar al destino. Para ello se utilizan diferentes métodos según el escenario, como pueden ser comunicaciones de tránsito y de peering.

## 2 Protocolo BGP

Debido a la necesidad de interconectar Sistemas Autónomos y a que cada uno de ellos tiene protocolos de rutado interior y políticas diferentes, se desarrolló el protocolo de rutado exterior BGP. Es un protocolo simple, de tipo "path vector" que se centra en optimizar la escalabilidad de la red en vez de en el tiempo de convergencia.

Trabaja sobre TCP en el puerto 179 para conseguir una comunicación fiable y además, ofrece retransmisión de paquetes, mecanismos de checksum y no duplica los paquetes. El socket de TCP siempre se dirige de un router a otro, es decir, no existen broadcast ni multicast.



La función de BGP es recoger información sobre los ASs vecinos, difundir las redes alcanzables del propio AS a sus AS vecinos y decidir la ruta más óptima. Desde 2006 se utiliza la versión 4 de BGP que se encuentra recogida en el RFC 4271 [18].

## 2.1 Tipos de mensajes

El protocolo BGP hace uso de cinco tipos de mensajes para llevar a cabo su funcionalidad. Concretamente estos mensajes son:

- **OPEN:** se envía para iniciar una sesión BGP entre dos routers y para negociar los parámetros que caracterizarán la sesión como la versión BGP, el número AS del emisor y el tiempo de duración de la sesión. En caso de recibir este tipo de mensaje y no querer aceptarlo, basta con enviar un mensaje de NOTIFICATION.
- **KEEPALIVE:** es la confirmación del mensaje OPEN. Cuando el tiempo de duración de la sesión es limitado, es necesario enviar este tipo de mensajes cada cierto tiempo (20 segundos normalmente) para mantener la sesión. Mientras no haya cambios en la tabla de rutado, sólo se enviarán este tipo de mensajes.
- **UPDATE:** sirve para enviar información de encaminamiento como los atributos de la ruta, las rutas a eliminar, NLRI (Network Layer Reachability Information) y la longitud de ruta.
- **NOTIFICATION:** se utiliza para finalizar o rechazar la sesión BGP junto con la de TCP. También se envía un código para indicar el error en caso de haberlo, como por ejemplo, que la versión BGP no sea compatible.
- **ROUTE\_REFRESH:** se usa para solicitar información de encaminamiento sin eliminar la conexión TCP.

## 2.2 Estados

En la figura siguiente se muestra el diagrama de estados en la que se basa el funcionamiento de este protocolo. A continuación, se mencionan dichos estados y se hace una breve descripción

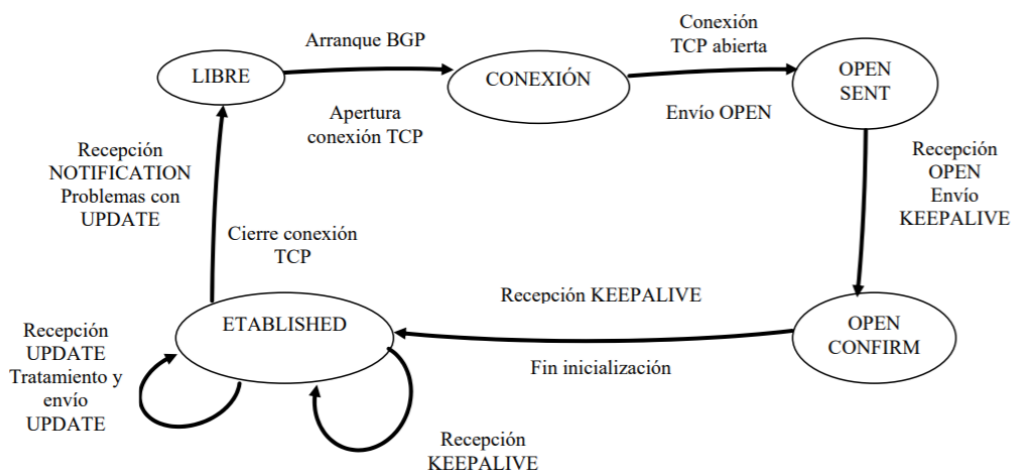


Figura 38: Estados de BGP

- **Libre:** el proceso BGP acaba de comenzar.
- **Conexión:** Inicio de comunicación TCP.
- **Open sent:** el TCP socket está establecido y un mensaje OPEN ha sido enviado.
- **Open confirm:** se recibe la respuesta al mensaje OPEN.
- **Established:** recibido un mensaje KEEPALIVE o UPDATE y estamos intercambiando información.

## 2.3 Sesiones BGP

En una sesión BGP participan dos routers (peers). Como en BGP existen dos tipos de comunicaciones (una intra-AS y otra inter-AS) también hay dos tipos de sesiones. En la versión 4 de BGP se definen los protocolos IBGP, que hace referencia a la sesión establecida dentro de un Sistema Autónomo y EBGP, que comunica los routers de borde de dos Sistemas Autónomos diferentes.

Existe una diferencia entre ambas que reside fundamentalmente en la forma de propagar los mensajes. Si un router aprende una ruta por EBGP, lo propagará a todos sus peers, tanto por IBGP como por EBGP. Sin embargo, si lo aprende por IBGP, solo lo anunciará a los peers por EBGP. De esta manera, se consiguen evitar bucles dentro un AS. Además, para propagar la información de encaminamiento a todos los routers intra-AS se utiliza una conexión de mallado total. La malla suele ser virtual ya que no hace falta conectividad física entre todos los peers; no como en EBGP que la conectividad física entre routers es necesaria.

Para evitar bucles entre Sistemas Autónomos, se utiliza el AS-PATH. Es la ruta que debe seguir el paquete para llegar al destino. Cada vez que un AS reciba un paquete comprobará si su id está en la lista y si no es así, lo encaminará (basándose en sus propias preferencias) añadiendo su propio id al AS-PATH. Si su id está en la lista, el AS rechazará el paquete.

## 2.4 Atributos

Los routers BGP anuncian rutas que consisten en un prefijo de red y unos atributos. Los atributos son principalmente utilizados para seleccionar la mejor ruta y para aplicar reglas de filtrado. En la siguiente tabla se pueden observar los más importantes:

ATRIBUTO	TIPO	DESCRIPCIÓN
ORIGIN	Well-known mandatory	Indica el origen de la información de ruta: IGP o EGP
AS_PATH	Well-known mandatory	Lista de ASs por los que ha pasado
NEXT_HOP	Well-known mandatory	Dirección IP del router que ha anunciado el paquete
LOCAL_PREFERENCE	Well-known discretionary	Métrica para seleccionar rutas de tráfico saliente. Gana el que tenga mayor valor
ATOMIC_AGGREGATE	Well-known discretionary	Indica si ha habido agregación
MULTI_EXIT_DISC	Optional non-transitive	Informa al peer exterior qué punto de entrada del AS utilizar. Gana el que tenga menor valor
ORIGINATOR_ID	Optional non-transitive	ID del router reflector
CLUSTER_LIST	Optional non-transitive	Indica el cluster origen
AGGREGATOR	Optional transitive	ID del router y AS que generó la ruta agregada
COMMUNITY	Optional transitive	Tag de routado de agrupación
WEIGHT	Optional not communicated to peers	Propiedad de Cisco. Gana el que tenga mayor valor

Tabla 12: Atributos BGP

## 2.5 Toma de decisiones

Cuando un router recibe un mensaje UPDATE, para alcanzar un destino específico tiene que elegir entre dos o más rutas. El proceso de decisión es el siguiente:

1. Si no se puede acceder al NEXT\_HOP, no se considera la ruta.
2. Se elige la ruta con mayor valor WEIGHT.
3. Se elige la ruta con mayor valor LOCAL\_PREF.
4. En el caso de tener el mismo valor LOCAL\_PREF, se elige una ruta originada en el propio router.
5. AS\_PATH más corto.
6. Menor ORIGIN (IGP<EGP<INCOMPLETE).
7. Menor MED.
8. Se elige una ruta aprendida por E-BGP antes que por I-BGP.
9. Se elige la ruta con el NEXT\_HOP más próximo, es decir, el vecino IGP más próximo.
10. Se elige la ruta hacia el router con ID más pequeño.

Para almacenar la información de encaminamiento, cada router BGP tiene 3 tipos de tablas de encaminamiento que se llaman RIB (Routing Information Basis).

- **Adj-RIB-in:** se almacenan los prefijos aprendidos por un vecino en concreto. Hay tantas tablas como vecinos.
- **Loc-RIB:** guarda las mejores rutas seleccionadas tras aplicar los filtros correspondientes. Sólo hay una por AS.
- **Adj-RIB-out:** almacena los prefijos que serán anunciados a otros vecinos. La construcción de la tabla proviene de la tabla Loc-RIB. Sólo hay una tabla de este tipo por cada par BGP.

## 3 Peering

El peering es una interconexión voluntaria de redes de Internet separadas administrativamente con el fin de intercambiar tráfico entre los usuarios de cada red. En una relación peering ninguna de las partes paga a la otra, cada uno deriva y retiene los ingresos de sus propios clientes.

Un acuerdo entre dos o más redes se lleva a cabo instalando un enlace físico entre las dos redes, un intercambio de información mediante el protocolo BGP y un documento contractual formalizado. Cuando esto sucede, las dos redes pasan a llamarse peers.

Como se ha explicado anteriormente, Internet está dividida en Sistemas Autónomos. Estas deben crear relaciones entre ellas para lograr una comunicación global. Las relaciones entre estas redes generalmente se describen mediante alguna de estas tres categorías:

- **Cliente (venta):** la red destino vende el servicio de acceso directo a su red. Es decir, la red del usuario toma el rol de cliente de la red a la que se quiere conectar.
- **Tránsito (pago):** la red del usuario compra el servicio de tránsito a una red, donde esta otra red a su vez también ha comprado el acceso directo a la red destino.
- **Peering:** las dos redes intercambian tráfico sin costes por el beneficio mutuo.

Internet se basa en el principio de accesibilidad total, lo que significa que cualquier usuario de Internet puede comunicarse con otro. Por lo tanto, para que una red alcance cualquier otra red mediante Internet, la red debe:

- Vender servicio de tránsito a esa red destino.
- Tener una relación de peering directamente con esa red o con otra red que a su vez venda servicio de tránsito para llegar a la red destino.
- Comprar servicio de tránsito a otra red el cual venda, haga peering o compre el acceso a esa red destino.

La razón principal por la que se están fomentando las relaciones peering es la reducción de costes de los servicios de tránsito. Como se puede apreciar en la imagen, si el cliente A quisiera comunicarse con el cliente B, el AS4 debería pagar por todo el servicio de tránsito a AS1 y lo mismo le sucede al AS5, que tendría que pagar por dicho servicio al AS2. En cambio, con una relación de peering, estos costes desaparecerían y sólo se encargarían del gasto del enlace físico entre el AS4 y AS5.

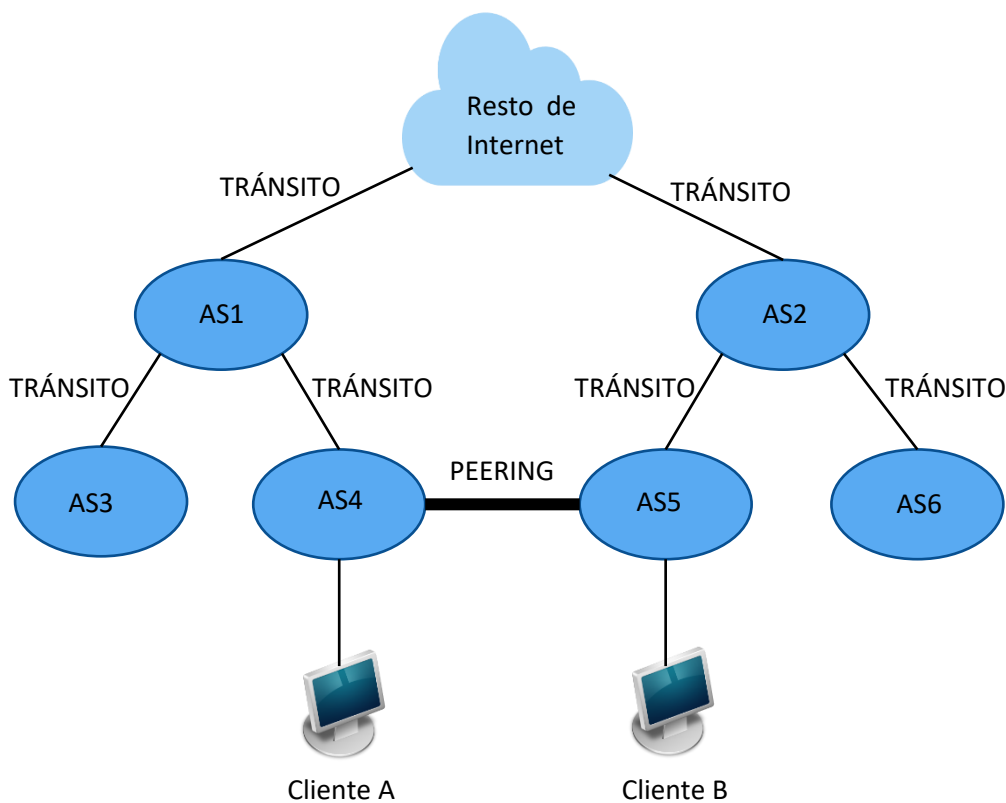


Figura 39: Peering vs tránsito

Asimismo, el peering también aporta otros beneficios:

- Mayor redundancia: al reducir la dependencia de uno o más proveedores de tránsito.
- Mayor control de rutado sobre el propio tráfico.
- Mejora del rendimiento: se pueden evitar posibles cuellos de botellas con una ruta directa.
- Mejor percepción de la red de uno mismo.

### 3.1 Tipos de peering

A día de hoy, a pesar de que existan otros tipos de peering, los más destacables y los más utilizados son los siguientes:

- **Peering público:** es una interconexión que utiliza un switch compartido por múltiples partes, como puede ser un switch Ethernet. Esta interconexión se logra a través de una tecnología de acceso de la Capa 2, generalmente llamada estructura compartida. En estas ubicaciones, los operadores se interconectan con otros a través de un único puerto físico. Hoy en día a estas ubicaciones se les denomina exchange points (IXP). La mayoría de los IXPs que hay en el mundo pueden tener cientos de participantes y sus instalaciones abarcan varios edificios por toda la ciudad.  
Dado que el peering público permite interconectar a cualquier red interesada a cientos de red a través de un único puerto, se debe considerar que ofrecen una menor capacidad que el peering privado. Muchas de las redes pequeñas o redes que acaban de empezar a hacer peering, encuentran que los IXPs son una excelente manera de interconectar con otras redes.
- **Peering privado:** es la interconexión directa entre solo dos redes, a través de un medio de la Capa 1 o 2 que ofrece una capacidad dedicada que no es compartida por otros. La mayor parte del tráfico de Internet, especialmente entre las redes más grandes, se produce entre peering privado. Sin embargo, muchas redes no están interesadas a proporcionar este tipo de conexiones a redes más pequeñas o nuevas que aún no han demostrado que puedan proporcionar un beneficio considerable al otro peer.

## 4 Internet Assigned Numbers Authority

Internet Assigned Numbers Authority o IANA, es responsable de mantener la colección crítica de registros que aseguran una coordinación global de las zonas DNS root, direcciones IP y otras fuentes de protocolos de Internet. Los registros IANA pueden formar parte de tres categorías en función de su funcionalidad en Internet:

### DIRECCIONES IP

IANA incluye el registro global para direcciones IPv4 e IPv6 y para ASNs. Estas listas contienen entradas para todos los rangos IP y bloques ASN que estén habilitados para el uso en Internet. IANA delega la responsabilidad de asignar direcciones IP y ASNs a los Registros Regionales de Internet. Sin embargo, IANA tendrá la capacidad de realizar cualquier cambio en los registros globales de acuerdo a las políticas acordadas por la comunidad global.

### DNS ROOT ZONE

DNS es una base de datos jerárquica que enlaza nombres de dominios como `www.ehu.es` a una dirección IP, la cual es utilizada para enviar información entre ordenadores.

IANA es la organización encargada de mantener el nivel más alto de esta jerarquía, el DNS root. Esta, contiene punteros a bases de datos donde se encuentran los dominios de un nivel inferior como, por ejemplo, `.com`.

## PARÁMETROS DE PROTOCOLOS

Con el objetivo de estar seguros de que un ordenador entiende a otro cuando se comunican, ciertos números usados en los protocolos de red deben tener un único significado global. Estos parámetros de protocolos vienen definidos por la IETF. IANA mantiene y publica estos registros para que luego sean utilizados por los desarrolladores de software para asegurar la estabilidad de las comunicaciones.

### 4.1 RIPE NCC

Como se ha mencionado anteriormente, IANA delega la responsabilidad de asignar direcciones IP y ASNs a los Registros Regionales de Internet. Los RIR actualmente consisten en varias bases de datos donde los operadores publican sus anuncios de red y políticas de rutado. De esta manera otras operadoras podrán consultar y hacer uso de esta información. El RIR que corresponde a Europa se llama RIPE NCC [19]. Es una organización sin ánimos de lucro que apoya la infraestructura de Internet a través de la coordinación técnica en la región de servicio. Su actividad más destacada es actuar como el RIR que proporciona recursos globales de Internet y servicios relacionados (IPv4, IPv6 y AS Number) a los miembros en la región de servicio. Por lo tanto, esta organización forma parte del backbone de internet.

Asimismo, beneficia a la administración de la red en varias formas:

- **Filtrado de rutas:** el tráfico puede ser filtrado en base a los objetos "route" registrados, previniendo problemas de red causados por anuncios de red maliciosos o malintencionados.  
El filtrado puede utilizarse en enlaces de clientes de tránsito o peering donde los peers acuerdan los filtros basados únicamente en las rutas registradas. Si un peer anuncia un route no registrado, este será filtrado.
- **Configuración de routers:** herramientas como IRRToolset facilitan la creación de configuraciones. Sugiere agregación de redes, verifica las rutas del objeto y corrige la sintaxis del RPSL.
- **Visión global del rutado de Internet:** si todas las redes son registradas en el IRR, se podría obtener una imagen global del rutado y de su política, mejorando la integridad global.
- **Solucionar problemas de red:** el registro de rutado hace más fácil la identificación de un posible fallo de red proveniente de fuera de tu red. Se puede contactar con el Sistema Autónomo asociado al problema para intentar arreglarlo.

La base de datos RIPE contiene registros de:

- Asignaciones de espacio de direcciones IP
- Asignaciones de números de Sistema Autónomos
- Registros de DNS reversos
- Información del contacto
- Información de la política de enrutamiento (en el RIR)

RIPE NCC ingresa parte de esta información, como las asignaciones de direcciones IP y asignaciones de ASN que se dieron a un determinado titular de recursos. Otra información es añadida por los propios propietarios de recursos, tales como asignaciones de clientes, DNS reverso, enrutamiento e información de contacto.

Las personas que desean consultar esta información tienen varias opciones disponibles, que varían desde una interfaz web a una herramienta de línea de comandos, así como una API RESTful. Hay varias formas que pueden influir el alcance de la búsqueda. Por ejemplo, se puede consultar un identificador de registro específico, realizar una búsqueda de texto libre o realizar una consulta que incluya resultados de las bases de datos de whois ejecutadas por otros Registros Regionales de Internet.

La base de datos RIPE almacena toda la información en registros conocidos como objetos. Estos son bloques de texto en notación estándar llamada Routing Policy Specification Language (RPSL) y definida en la RFC 2622<sup>3</sup>. Un objeto tiene varios campos llamados atributos.

Las políticas de rutado más simples pueden ser creadas utilizando bases de datos de objetos route. Es un formato muy común que utilizan los operadores de red para configurar los routers de borde. Además de este tipo de objeto, también destacan:

- **Inetnum:** Un objeto inetnum contiene información sobre asignaciones de recursos de espacio de direcciones IPv4. Este es uno de los elementos principales de RIPE.
- **Aut-num:** El objeto aut-num tiene un doble propósito en la base de datos. Por un lado, contiene los detalles de registro de un Número de Sistema Autónomo (ASN) asignado por RIPE NCC. Por otro lado, permite publicar políticas de rutado. Este es el único objeto que se cruza entre estos dos registros en la base de datos RIPE.
- **Domain:** El objeto Domain sirve principalmente para registrar delegaciones inversas (traducciones de número a nombre) en la base de datos RIPE. Las delegaciones DNS inversas permiten que las aplicaciones mapeen un nombre de dominio desde una dirección IP.
- **Route-set:** Un objeto route-set es un conjunto de prefijos de ruta y no un conjunto de objetos de route de la base de datos. Los sets pueden construirse con nombres jerárquicos y también pueden incluir referencias directas a otros sets.
- **Route6:** un objeto route6 contiene información de rutado para los recursos de espacio de direcciones IPv6. Cada ruta interAS originada por un Sistema Autónomo puede especificarse usando un objeto route6 para direcciones IPv6.
- **AS-set:** un objeto AS-set forma un grupo de ASNs que puede ser referenciados en lugares donde un objeto AS-set puede ser usado. Los sets pueden ser contruidos con nombre jerárquicos y también pueden tener referencias a otros sets. Estas referencias también pueden ser hechas directamente usando el atributo "mbrs-by-ref".

## 4.2 Objeto route

Un objeto route contiene información de rutado para recursos de direcciones IPv4. Cada ruta entre dos Sistemas Autónomos puede ser definida por un objeto route. Este es uno de los principales elementos de RIPE.

La autorización para crear un objeto route puede ser complicada. Diferentes tipos de objetos de la base de datos requieren diferentes niveles de protección. El servidor utiliza

---

<sup>3</sup> <https://tools.ietf.org/html/rfc2622>

múltiples métodos de autorización. Estos están basados en el RPSS (Routing Policy System Security) y explicados detalladamente en la RFC 2725<sup>4</sup>.

Se puede observar abajo una tabla sobre el objeto route donde aparecen listados todos los posibles atributos.

Atributo	Presencia	Repetición	Indexed
route	mandatory	single	primary / lookup key
descr	mandatory	multiple	
origin	mandatory	single	primary / lookup key
pingable	optional	multiple	
ping-hd1	optional	multiple	inverse key
holes	optional	multiple	
org	optional	multiple	inverse key
member-of	optional	multiple	inverse key
inject	optional	multiple	
aggr-mtd	optional	single	
aggr-bndry	optional	single	
export-comps	optional	single	
components	optional	single	
remarks	optional	multiple	
notify	optional	multiple	inverse key
mnt-lower	optional	multiple	inverse key
mnt-routes	optional	multiple	inverse key
mnt-by	mandatory	multiple	inverse key
changed	optional	multiple	
created	generated	single	
last-modified	generated	single	
source	mandatory	single	

Tabla 13: Atributos objeto route

- **Route:** especifica el prefijo IPv4 del router. Junto con el atributo ORIGIN, constituye una pareja de claves primarias del objeto route. La dirección solo puede ser definida como un prefijo y puede incluir más de una dirección.
- **Descr:** una breve descripción del objeto.
- **Origin:** el AS Number del Sistema Autónomo que originó la ruta al sistema de rutado interdominio.
- **Pingable:** informa al operador de red sobre una dirección IP la cual debería de ser alcanzable desde fuera de la red. Se utiliza para diagnósticos de red.
- **Holes:** forma una lista de partes del prefijo de direcciones que no son alcanzables por la ruta de agregación.
- **Member-of:** identifica el grupo de objetos que el actual objeto quiere formar parte.
- **Inject:** informa qué routers han aplicado la agregación y cuándo.
- **Aggr-mtd:** explica cómo fue generada la agregación.
- **Mnt-lower:** incluye tokens de autorización para la creación del objeto route. Son utilizados para la creación de un objeto más específico (objeto hijo).
- **Mnt-routers:** incluye tokens de autorización para el emparejamiento exacto o la creación de objetos route más específicos. También puede incluir una lista de rangos de prefijos.

<sup>4</sup> <https://tools.ietf.org/html/rfc2725>



## ANEXO II: HERRAMIENTAS

En este anexo se describen diferentes herramientas que se han utilizado o que en un principio se valoró la posibilidad de hacer uso de ella. En primer lugar, se hace un análisis de la herramienta bgpq3 y del software Ansible. Por último, se describe el protocolo NETCONF que como ha sido mencionado en el módulo Comunicación en el apartado de 6.2 de este documento, se ha utilizado para la comunicación entre el terminal y los routers.

### 1 Herramienta bgpq3

La herramienta bgpq3 se utiliza crear filtros automatizados en routers Cisco y Juniper [20]. También sirve para generar listas de prefijos, listas de accesos extendidas, términos de políticas y listas de AS-PATH basadas en información RADb (Routing Assets Database). RADb es una base de datos de consulta pública dirigida por *Merit Networks* que almacena información fundamental sobre el rutado.

Este comando tiene las siguientes opciones:

OPCIÓN	DESCRIPCIÓN
-2	Permite a los routers registrarse en as23456 (AS de transición)
-3	Asume que el dispositivo es compatible con asn32
-4	Genera una lista de prefijos/accesos de IPv4
-6	Genera una lista de prefijos/accesos de IPv6
-A	Trata de agregar los filtros generados lo máximo posible
-a asn	ASN específico que debe ser denegado en caso de que la lista de prefijos esté vacía
-B	Generará el output en formato OpenBGPD (default: Cisco)
-b	Generará el output en formato BIRD (default: Cisco)
-d	Habilita el output del debug
-D	Usa la notación asdot para la lista de acceso del AS-PATH de Cisco
-E	Genera una lista de acceso extendida (Cisco) o un establecimiento de política usando filtros de rutado (Juniper)
-f AS Number	Genera el input de la lista de acceso del AS-PATH para el AS
-F fmt	Genera el output en el formato definido por el usuario
-G number	Genera output de la lista de acceso del AS-PATH
-h host[:port]	Host ejecutando a base de datos IRRD (default: whois.radb.net)
-J	Genera la config para Juniper (default: Cisco)
-j	Genera el output en formato JSON (default: Cisco)
-m length	La máxima longitud del prefijo aceptada (default: 32 para IPv4, 128 para IPv6)
-M match	Condiciones de emparejamiento extras para los filtros de los routers Juniper
-n	Genera la config para Nokia SR OS MD-CLI
-N	Genera la config para Nokia SR OS classic CLI
-l name	Nombre de la configuración generada
-L limit	Limita la profundidad de la recursión cuando se expande. Esto enlentecerá a bgpq3
-p	Habilita el uso de ASN privados
-P	Genera la lista de prefijos

-r length	Permite redes con longitud de máscara mayor que la indicada
-R length	Permite redes con longitud de máscara menor que la indicada
-s	Genera la secuencia de números de lista de prefijos en estilo IOS
-S sources	Usa solamente fuentes específicas (RADb, RIPE, APNIC...)
-t	Genera AS-sets en formatos OpenBGPD. BIRD, JASON
-T	Deshabilita e pipelining
-U	Genera el output en formato Huawei
-W length	Genera strings de AS-PATH de la longitud indicada como máximo
-X	Genera la config para los dispositivos Cisco IOS XR
-z	Genera la lista de los filtros de rutado de Juniper

Tabla 14: Herramienta bgpq3

## 2 Herramienta Ansible

Ansible es una plataforma de automatización, un framework para ejecutar una serie de operaciones que realizan tareas definidas [21]. Se utiliza comúnmente para aprovisionar, desplegar, y administrar la infraestructura en la nube, en los entornos virtuales y físicos. Ansible fue escrito por Michael DeHaan e inicialmente lanzado en 2012. Fue comprado por Red Hat en 2015.

### 2.2 Requerimientos del sistema

El ordenador que ejecuta Ansible y ejecuta playbooks se denomina máquina de control. Los sistemas gestionados por una máquina de control de Ansible se denominan nodos gestionados. La máquina de control se comunicará con los dispositivos Junos utilizando el protocolo NETCONF que se ejecuta sobre SSH. Por defecto, el servicio NETCONF en Junos utiliza el puerto TCP 830. Una máquina de control de Ansible que administrará los dispositivos Junos requiere:

- Un sistema operativo que no sea Windows. MacOS, Linux y otros sistemas operativos de tipo UNIX funcionan bien.
- Python 2.6 o 2.7.
- Un cliente SSH, típicamente OpenSSH. Esto generalmente viene instalado por defecto en Linux /Sistemas UNIX y MacOS.
- Los módulos Galaxy de Juniper y la biblioteca PyEZ Python de Juniper.

### 2.2 Características

La creación de la automatización con Ansible requiere un poco de conocimiento de programación: la programación requerida para muchas operaciones comunes ya está hecha y puesto a disposición en forma de módulos. A riesgo de simplificar demasiado, uno crea un playbook que describe la automatización que necesita al unir una serie de módulos.

Ansible incluye una gran selección de módulos que los diferencia en tres categorías: núcleo, curación y comunidad. También hay módulos desarrollados por la comunidad de usuarios de Ansible y disponibles a través de Ansible Galaxy<sup>5</sup>.

A partir de 2014, Juniper Networks publicó los denominados módulos Galaxy que permiten la administración de dispositivos Junos<sup>6</sup>. Las operaciones admitidas incluyen

<sup>5</sup> <https://galaxy.ansible.com/>

<sup>6</sup> <http://junos-ansible-modules.readthedocs.io/> o <https://galaxy.ansible.com/Juniper/junos/>

ejecutar comandos, descargar la configuración, realizar cambios de configuración, revertir los cambios de configuración y actualizar Junos.

A partir de 2016 con la versión 2.1, Ansible agregó módulos principales que funcionan con dispositivos Junos<sup>7</sup>. Las operaciones admitidas son muy similares a los módulos Galaxy de Juniper, pero con diferencias en detalles como los nombres de los módulos y en cómo usar los módulos (lo que significa que los playbooks escritos para un conjunto de módulos deben ser reescritos para utilizar el otro conjunto de módulos). Juniper recomienda sus módulos Galaxy sobre los módulos principales de Ansible.

## 2.3 Conceptos

A continuación, se describen brevemente algunos de los términos y conceptos básicos<sup>8</sup> que utiliza Ansible.

**Playbook:** archivo que define el proceso de automatización deseado mediante la llamada a una serie de módulos. Ansible ejecuta el playbook, llamando a los módulos que implementan las tareas necesarias para realizar la automatización.

**Módulo:** programa que realiza una tarea específica, como copiar un archivo, instalar software, o reiniciar un dispositivo.

**Tarea:** dentro de un playbook, una tarea es una llamada para ejecutar un módulo que realiza un trabajo, como copiar un archivo o configurar un dispositivo. Las tareas generalmente incluyen uno o más argumentos, datos que añaden información a lo que debe hacer el módulo, como el nombre del archivo a copiar, o la dirección IP del dispositivo a configurar.

**Play:** dentro de un playbook, un juego es una colección de tareas. Un playbook tendrá una o más plays. Si un playbook contiene varias plays, es probable que las plays tengan diferentes requisitos: por ejemplo, pueden ejecutarse en diferentes hosts.

**Fact:** mientras se ejecuta un playbook, Ansible obtiene información sobre los hosts involucrados. Lo aprendido se denomina tarea y puede ser referenciado por su nombre en el playbook.

**Variable:** Datos sobre un host o grupo declarado por el usuario. Al igual que los fact, las variables pueden ser referenciadas por su nombre. La diferencia es que las variables son declaradas por el usuario.

**Rol:** Una forma de organizar el comportamiento deseado en unidades reutilizables.

**Plantilla:** archivo que contiene texto estático, como comandos de configuración del dispositivo. Las plantillas son escritas usando el lenguaje Jinja2.

**Inventario:** lista de dispositivos que Ansible conoce, posiblemente con algunas variables predeterminadas, como la dirección IP de administración del dispositivo. El inventario normalmente se almacena en un archivo o directorio llamado inventario.

---

<sup>7</sup> [http://docs.ansible.com/ansible/list\\_of\\_network\\_modules.html#junos](http://docs.ansible.com/ansible/list_of_network_modules.html#junos)

<sup>8</sup> [https://docs.ansible.com/ansible/latest/reference\\_appendices/glossary.html](https://docs.ansible.com/ansible/latest/reference_appendices/glossary.html)

### 3 NETCONF

Network Configuration Protocol (NETCONF) es un protocolo de administración de red desarrollado y estandarizado por el IETF. Fue desarrollado en el grupo de trabajo de NETCONF y publicado en diciembre de 2006 en la RFC 4741 y luego revisado en junio de 2011 y publicado como RFC 6241 [22].

NETCONF proporciona mecanismos para instalar, manipular y eliminar la configuración de dispositivos de red. Sus operaciones se realizan sobre la capa Remote Procedure Call (RPC). El protocolo NETCONF utiliza una codificación de datos basada en el lenguaje Extensible Markup Language (XML) para los datos de configuración, así como para los mensajes de protocolo.

Las tecnologías previas a este protocolo, como pueden ser SNMP y telnet no abastecían todas las necesidades actuales. NETCONF fue diseñado para cubrir ese vacío:

- Distinción entre la configuración y los datos con estado.
- Múltiples almacenes de datos de configuración (ejecución, inicio).
- Recuperación selectiva de datos con filtrado (XPath y filtrado de subárbol),
- Configuración de transacciones de cambio.

#### 3.1 Características

Las siguientes son las características principales de NETCONF:

- Utiliza SSH como protocolo de transporte.
- Está orientado a la conexión y orientado a la sesión.
- Utiliza XML para codificar y representar datos. XML es legible y estructurado por humanos, por lo que puede ser consumido fácilmente por la máquina (Scripts y programas).
- Define múltiples operaciones para interactuar con el dispositivo de red.
- Utiliza métodos RPC para invocar estas operaciones en el dispositivo de red remoto.

El siguiente diagrama describe la pila de protocolos para NETCONF [23].

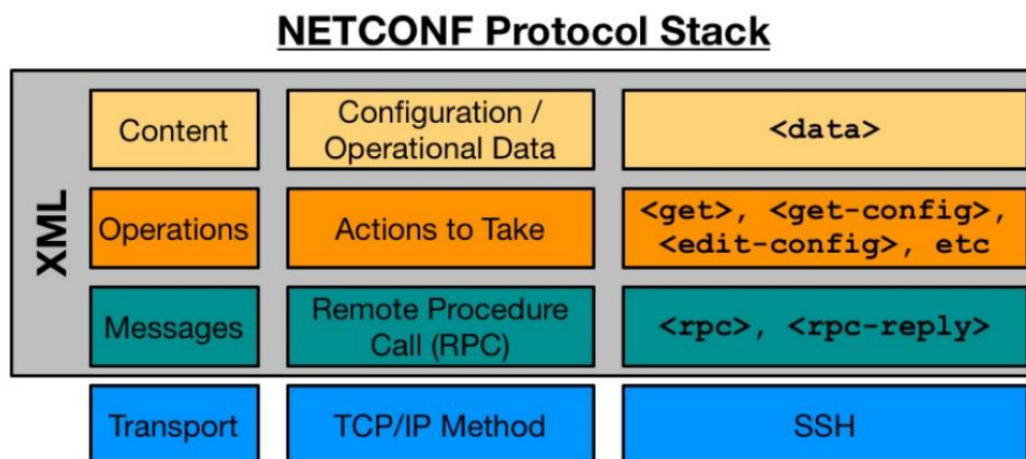


Figura 40: NETCONF

## 3.2 Sesión

Los procedimientos que tienen lugar para que se establezca una sesión de NETCONF son los siguientes:

1.- Esta comunicación siempre comienza con el cliente utilizando una de las dos opciones: ya sea abrir una sesión SSH en el puerto 830 (NETCONF PORT) o abrir una sesión SSH y solicitar el subsistema NETCONF.

2.- El Servidor / Agente responde por las Capacidades (Commit, Validate, XPATH, etc.) que admite (y en caso de que los datos estén modelados en YANG<sup>9</sup>, por los módulos de YANG admitidos también).

3.- Después de que el Cliente / Administrador responda con un mensaje HELLO para declarar sus capacidades, entonces comienza a emitir llamadas RPC hacia el servidor para recuperar la información o para poner una nueva configuración en el dispositivo.

## 3.3 Comandos

El estándar NETCONF (RFC 4741) definió las siguientes operaciones que se pueden ejecutar:

COMANDO	DESCRIPCIÓN
get	Recupera la configuración de ejecución y la información de estado del dispositivo
get-config	Recupera toda o parte de la configuración de un específico almacén de datos de configuración
edit-config	Carga toda o parte de la configuración de un específico almacén de datos
copy-config	Copia un almacén de datos de configuración completo a otro almacén de datos de configuración
delete-config	Elimina la configuración de un almacén de datos
commit	copia el almacén de datos candidato al almacén de datos en ejecución
lock / unlock	Bloquea/desbloquea la configuración completa de un almacén de datos de un dispositivo
close-session	Solicita de terminación de una sesión de NETCONF
kill-session	Fuerza la terminación de una sesión de NETCONF

Tabla 15: Comandos NETCONF

<sup>9</sup> Lenguaje de modelado de datos para la definición de datos enviados a través de protocolos de administración de red como NETCONF.