Diseño de un Sistema de aprovisionamiento automático de redes.
(Zero Touch Provisioning)

Iñigo Esteban Oleaga
Máster Universitario en Ingeniería de Telecomunicación UOC-URL
Telemática

Consultor: José López Vicario
Profesor Responsable: Xavier Vilajosana Guillen

9 de junio 2019

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del Trabajo:	Diseño de un Sistema de aprovisionamiento automático de redes. (Zero Touch Provisioning)
Nombre del autor:	*Iñigo Esteban Oleaga*
Nombre del consultor/a:	*José López Vicario*
Nombre del PRA:	*Xavier Vilajosana Guillen*
Fecha de entrega:	*06/2019*
Titulación:	*Máster Universitario en Ingeniería de Telecomunicación UOC-URL*
Área del Trabajo Final:	*Telemática*
Idioma del trabajo:	*Español*
Palabras clave	*NetOps, automatización, redes, ZTP.*

Resumen del Trabajo:

Desde hace ya años, existen en el mercado varias herramientas de automatización y orquestación orientadas a los entornos de granjas de servidores, donde tiene mucho sentido poder gestionar los múltiples servidores de una manera que evite que se tenga que ir máquina por máquina haciendo configuraciones.

Sin embargo, hasta ahora no se había justificado el uso de estas herramientas para gestionar entornos de red empresariales. Lo cierto es que las redes empresariales cada vez son más complejas e implican disponer de un alto número de equipos que hay que administrar. Esta necesidad ha provocado que los ingenieros de telecomunicaciones hayan comenzado a cuestionarse el modo en el que se gestionan estos equipos y a pensar en soluciones enfocadas a la automatización de redes de telecomunicaciones.

El mundo de las redes está repleto de configuraciones repetitivas que se suelen hacer a mano cada vez que surge la necesidad. La filosofía que impulsa la automatización y la orquestación promueve la idea de que merece más la pena invertir el tiempo en hacer ingeniería que en aplicar las mismas configuraciones una y otra vez.

A día de hoy existen varias soluciones enfocadas a la automatización de entornos de servidores como pueden ser Chef, Puppet o Ansible. Algunas de estas soluciones ya han comenzado a dar ciertos pasos hacia la

automatización de redes. El objetivo principal del proyecto sería, precisamente, analizar las soluciones disponibles para la automatización de redes y diseñar alguna solución que pueda resultar útil en entornos empresariales.

Abstrac:

For years now, there are several automation and orchestration tools in the market focused on server farm environments, where it makes sense to manage multiple servers in a way that avoids having to go machine by machine making configurations.

However, until now, the use of these tools to manage enterprise network environments has not been justified. The truth is that business networks are increasingly more complex and involve having a high number of devices that must be provisioned. This need has caused the telecommunications engineers to think about the way in which these devices are managed and to think about solutions focused on the automation of telecommunications networks.

The networking enviroment is full of repetitive configurations that are usually made by hand whenever the need arises. The philosophy that drives automation and orchestration is that it is worth investing more time in engineering than in applying the same configurations over and over again.

Today there are several solutions focused on the automation of server environments such as Chef, Puppet or Ansible. Some of these solutions have already begun to take certain steps towards network automation. The main objective of the project would be, precisely, to analyze the available solutions for the automation of networks and to design a solution that could be useful in business environments.

Índice

Tabla de ilustraciones

1. Introducción

1.1 Contexto y justificación del Trabajo

Desde hace ya años, existen en el mercado varias herramientas de automatización y orquestación enfocadas a los entornos de granjas de servidores, donde tiene mucho sentido poder gestionar los múltiples servidores de una manera que se evite tener que ir máquina por máquina haciendo configuraciones.

Sin embargo, hasta ahora no se había justificado el uso de estas herramientas para gestionar entornos de red empresariales, salvo que la empresa fuera una gran compañía de telecomunicaciones. Lo cierto es que las redes empresariales cada vez son más complejas e implican disponer de un alto número de equipos que hay que administrar. Esta necesidad ha provocado que los ingenieros de telecomunicaciones hayan comenzado a cuestionarse el modo en el que se gestionan estos equipos y a pensar en soluciones enfocadas a la automatización y la orquestación de las redes de telecomunicaciones.

El mundo de las redes está repleto de configuraciones repetitivas que se suelen hacer a mano cada vez que surge la necesidad. La filosofía que impulsa la automatización y la orquestación promueve la idea de que merece más la pena invertir el tiempo en hacer ingeniería que en malgastar el tiempo en aplicar las mismas configuraciones una y otra vez.

A día de hoy existen varias soluciones enfocadas a la automatización y la orquestación de entornos de servidores como pueden ser Chef, Puppet o Ansible. Algunas de estas soluciones ya han comenzado a dar ciertos pasos hacia la automatización y orquestación de redes. El objetivo principal del proyecto sería, precisamente, analizar las soluciones disponibles para la automatización y orquestación de redes y diseñar algunas soluciones que puedan resultar útiles en entornos de redes empresariales. Los ámbitos de aplicación de dichas soluciones serían:

- Zero Touch Provisioning (ZTP): configurar una red y todos sus dispositivos desde cero en pocos pasos.
- Continous Integration / Continous Deployment (CI/CD): desarrollar configuraciones, actualizaciones o mejoras y desplegarlas sobre

una red completa consiguiendo así, la estandarización de las soluciones y la homogeneidad de la red.

- Seguridad TIC: diseñar políticas de seguridad de manera centralizada y distribuirlas a lo largo de los diferentes equipos que conforman la red.
- Software Defined Networking (SDN): mejorar el comportamiento de la red rompiendo los límites de lo que permiten los protocolos de encaminamiento actuales. Por ejemplo, balancear tráfico entre enlaces en base a la salud de los mismos (latencia, tasa de pérdida de paquetes…) o aplicar técnicas de encaminamiento como OSPF Fibbing y ExaBGP[1].

La velocidad y el aumento de la eficiencia son beneficios obvios de la automatización. Sin embargo, no se debe olvidar la consistencia y el aumento resultante en la confiabilidad. Es realmente tedioso y propenso a errores crear servicios complejos que implican cambiar múltiples configuraciones de dispositivos mediante un largo proceso manual: el mismo proceso se vuelve significativamente más confiable cuando está automatizado o al menos, en caso de contener un error, permitiría un diagnóstico más sencillo ya que la ejecución se mantendría constante.

También es más fácil hacer cumplir reglas de seguridad consistentes cuando el dispositivo o la implementación del servicio están automatizados, porque eso hará que los administradores de la red no puedan hacer ningún cambio manual ni sobre la marcha. Así, las auditorias de seguridad se vuelven más llevaderas, ya que las configuraciones demostrarán coherencia a lo largo de toda la red.

Finalmente, un proceso de implementación de servicio automatizado alivia la necesidad de ventanas de mantenimiento excesivamente largas, ya que los cambios en la configuración del dispositivo ya no representan una modificación que se deba documentar, revisar y planificar. Los únicos cambios reales que deben gestionarse con marcos como ITIL, son los despliegues de nuevas versiones del sistema de automatización.

1.2 Objetivos del Trabajo

[1] OSPF Fibbing y ExaBGP son técnicas de encaminamiento que modifican el comportamiento de los protocoloso OSPF y BGP respectivamente.

A continuación, se enumeran los objetivos ordenados atendiendo a los diferentes ámbitos de aplicación de la solución:

Objetivos generales

1. Analizar las soluciones de automatización existentes en el mercado.
2. Seleccionar la herramienta adecuada para la automatización de equipamiento de red.
3. Analizar los nuevos enfoques en la administración de redes.
4. Estudiar los fundamentos de la administración remota de equipos de red.

Objetivos del diseño

1. Diseñar una solución ZTP y su implementación en una empresa.
2. Obtener diagramas orientativos del diseño.
3. Obtener un listado de los parámetros a configurar.
4. Analizar el coste de implantación del sistema.

1.3 Alcance

En las fases previas al proyecto se pretendía abordar la Automatización de Redes de manera general, explorando diferentes ámbitos de aplicación. Sin embargo, se ha considerado que resultará más interesante enfocar los esfuerzos en un único ámbito con la intención de profundizar más en el tema. De esta forma, el proyecto ofrecerá una perspectiva de la automatización de redes a más bajo nivel del que se pretendía inicialmente. Además, de esta manera resultará más interesante y enriquecedor porque se ahondará en conceptos más técnicos.

El alcance de este proyecto es diseñar un sistema de Zero Touch Provisioning a nivel teórico. Se analizará de forma exhaustiva qué se necesita y qué enfoque hay que darle a la situación para obtener dicho sistema.

Además, se analizará la viabilidad técnica y económica del proyecto obteniendo como resultados una planificación realista y un presupuesto económico detallado para llevarlos a cabo.

Quedan fuera del alcance de este proyecto la obtención de una solución funcional. No se pretende implementar la solución a nivel técnico de tal

forma que se pueda usar, sino diseñar el sistema para su posterior implementación.

También quedan fuera del alcance de este proyecto las explicaciones técnicas sobre switching, routing y seguridad de la información, así como los procesos de instalación y configuración de las herramientas seleccionadas para el diseño de la solución.

1.4 Metodología

En primer lugar, se analizarán las diferentes herramientas de Automatización y Orquestación que existen en el mercado, evaluando el ámbito de aplicación de las mismas, sus costes, su facilidad de uso y su integración con herramientas de terceros. Posteriormente, se evaluará cuál de las soluciones es la mejor para poder llevar a cabo un proyecto de diseño de un Sistema de Automatización de Operaciones en Redes de Datos.

Después, se analizará de forma exhaustiva la rutina habitual de trabajo de un ingeniero de telecomunicaciones al gestionar las redes corporativas de una empresa. De esta forma, se podrá obtener una lista de tareas o de pasos que son necesarios para realizar una nueva red.

Una vez obtenidos los pasos, se diseñará la solución concreta para lograr un sistema Zero Touch Provisioning. Para ello, se analizarán todas las necesidades técnicas que requiere el diseño del sistema y se especificarán de manera concreta y detallada las diferentes fases del proyecto. Aparte de las reflexiones y explicaciones pertinentes, se obtendrán diagramas de flujo que orientarán la programación del sistema, en el lenguaje que ofrezca la herramienta que se seleccione. Después, se evaluará el desarrollo y la aplicación de dichas soluciones en el ámbito empresarial, tomando como referencia los recursos, las herramientas y las dimensiones correspondientes a una empresa real. Para ello, se presupuestará y se estimará la planificación y los recursos necesarios para la implantación de las soluciones diseñadas en dicha empresa.

Por último, se obtendrán las conclusiones que permitirán evaluar si las soluciones diseñadas tienen una aplicación viable y cumplen con los objetivos definidos.

1.5 Planificación del Trabajo

Las tareas a alto nivel que se llevarán a cabo dentro del alcance de este proyecto son las siguientes:

Nombre de tarea	Duración	Comienzo	Fin
Desarrollo de la memoria	67 días	jue 07/03/19	dom 09/06/19
Elaboración de la Introducción	8 días	jue 07/03/19	lun 18/03/19
Elaboración del contexto y justificación del Trabajo	8 días	jue 07/03/19	lun 18/03/19
Elaboración de los objetivos del Trabajo	8 días	jue 07/03/19	lun 18/03/19
Desarrollo de la Metodología	8 días	jue 07/03/19	lun 18/03/19
Planificación del Trabajo	8 días	jue 07/03/19	lun 18/03/19
Desarrollo Estado del Arte	9 días	mar 19/03/19	vie 29/03/19
Descripción del escenario de aplicación	3 días	mar 19/03/19	jue 21/03/19
Análisis de soluciones existentes	4 días	vie 22/03/19	mié 27/03/19
Comparativa de las herramientas	1 día	jue 28/03/19	jue 28/03/19
Elección de la herramienta	1 día	vie 29/03/19	vie 29/03/19
Desarrollo del diseño del sistema ZTP	**46 días**	**lun 01/04/19**	**lun 03/06/19**
Desarrollo de la introducción	1 día	lun 01/04/19	lun 01/04/19
Profundización en la filosofía ZTP	2 días	mar 02/04/19	mié 03/04/19
Documentación previa a la implementación del sistema ZTP	**15 días**	**jue 04/04/19**	**mié 24/04/19**
Diseño de diagramas lógicos y físicos de red estándar de una oficina	3 días	jue 04/04/19	lun 08/04/19
Análisis de la configuración de routers WAN	4 días	mar 09/04/19	vie 12/04/19
Diseño de la plantilla con datos de configuración de routers WAN	1 día	mar 09/04/19	mar 09/04/19
Análisis de la configuración de switches de CORE	4 días	lun 15/04/19	jue 18/04/19
Diseño de la plantilla con datos de configuración de switches de CORE	1 día	lun 15/04/19	lun 15/04/19
Análisis de la configuración de switches de ACCESO	4 días	vie 19/04/19	mié 24/04/19
Diseño de la plantilla con datos de configuración de switches de ACCESO	1 día	mié 24/04/19	mié 24/04/19
Entrega PEC 2	0 días	mié 24/04/19	mié 24/04/19
Diseño del sistema ZTP	**28 días**	**jue 25/04/19**	**lun 03/06/19**
Fase 1: Traducción de documentación a datos interpretables por Ansible	5 días	jue 25/04/19	mié 01/05/19
Fase 2: Proceso de configuración del router WAN mediante Ansible	**10 días**	**jue 02/05/19**	**mié 15/05/19**
Diseño del diagrama de flujo del funcionamiento de la fase 2.	10 días	jue 02/05/19	mié 15/05/19
Explicación de los procesos de la fase 2.	10 días	jue 02/05/19	mié 15/05/19

Fase 3: Proceso de configuración del switch de CORE mediante Ansible	**3 días**	**jue 16/05/19**	**lun 20/05/19**
Diseño del diagrama de flujo del funcionamiento de la fase 3.	3 días	jue 16/05/19	lun 20/05/19
Explicación de los procesos de la fase 3.	3 días	jue 16/05/19	lun 20/05/19
Fase 4: Obtención y corrección de la topología de red	**3 días**	**mar 21/05/19**	**jue 23/05/19**
Diseño del diagrama de flujo del funcionamiento de la fase 4.	3 días	mar 21/05/19	jue 23/05/19
Explicación de los procesos de la fase 4.	3 días	mar 21/05/19	jue 23/05/19
Entrega PEC3	0 días	mié 22/05/19	mié 22/05/19
Fase 5: Proceso de configuración de switches de ACCESO mediante Ansible	**3 días**	**vie 24/05/19**	**mar 28/05/19**
Diseño del diagrama de flujo del funcionamiento de la fase 5.	3 días	vie 24/05/19	mar 28/05/19
Explicación de los procesos de la fase 5.	3 días	vie 24/05/19	mar 28/05/19
Fase 6: Proceso de revisión de la topología	2 días	mié 29/05/19	jue 30/05/19
Fase 7: Proceso de reporte y documentación	2 días	vie 31/05/19	lun 03/06/19
Validación y evaluación del proyecto	**3 días**	**mar 04/06/19**	**jue 06/06/19**
Evaluación técnica de la solución	1 día	mar 04/06/19	mar 04/06/19
Evaluación económica de la solución	1 día	mié 05/06/19	mié 05/06/19
Evaluación de objetivos	1 día	jue 06/06/19	jue 06/06/19
Planificación del desarrollo	1 día	mar 04/06/19	mar 04/06/19
Diseño del diagrama de Gantt	1 día	mar 04/06/19	mar 04/06/19
Conclusiones	1 día	mié 05/06/19	mié 05/06/19
Entrega de la memoria	0 días	dom 09/06/19	dom 09/06/19
Desarrollo de la presentación del proyecto	3 días	lun 10/06/19	mié 12/06/19
Entrega de la presentación	0 días	dom 16/06/19	dom 16/06/19
Defensa del proyecto	6 días	lun 17/06/19	lun 24/06/19
Fin del Trabajo de final de máster	0 días	mar 25/06/19	mar 25/06/19

1.5.1 Planificación

Se trasladan las tareas detectadas a un diagrama de Gantt (Ilustración 1: Diagrama de Gantt de la planificación. Fragmento 1., Ilustración 2: Diagrama de Gantt de la planificación. Fragmento 2.) para poder definir el desarrollo del trabajo en el tiempo. Apréciense los comentarios que se hacen a continuación:

- Los fines de semana no se consideran días laborables, aunque no se señale el sombreado en el diagrama.
- Los días laborables implican jornadas de 6 horas de dedicación al Trabajo Final de Máster.
- Se han definido hitos que contemplan las entregas que se exigen en la asignatura.

1.5.2 Primera revisión de la planificación (24/04/2019)

Al comenzar a trabajar en el proyecto se ha visto la necesidad de reenfocar el alcance. Es más interesante centrarse en uno de los ámbitos de aplicación del proyecto que repasar por encima los cuatro que se mencionan. Por este motivo, la planificación ha sido reajustada para alcanzar los objetivos propuestos antes de la fecha límite.

Los plazos previstos se han ido cumpliendo, aunque a día de hoy se aprecia un pequeño retraso sobre lo establecido. Las plantillas de configuración de los diferentes elementos de la red deberían estar acabadas y no lo están. No obstante, habiendo detectado este retraso a falta de tantos días para la fecha de entrega final, parece que no existirá ningún problema para recuperar los tiempos a medida que se sigue avanzando con el proyecto.

1.5.3 Segunda revisión de la planificación (22/05/2019)

En esta segunda revisión se puede destacar que se ha recuperado el atraso que se apreciaba en la entrega anterior. Además, se ha realizado el esfuerzo para realizar todo el trabajo técnico del proyecto, por lo que, a fecha del 22 de mayo 2019, se han desarrollado todos los apartados de la memoria.

Esto implica que durante las próximas semanas se invertirá el tiempo en pulir los detalles de la memoria, además de ir adelantando trabajo sobre la presentación que se espera entregar antes del día 16 de junio.

1.5.4 Última revisión de la planificación (09/06/2019)

La planificación quedó completada en la última revisión. Se aprovecha para corregir pequeños detalles.

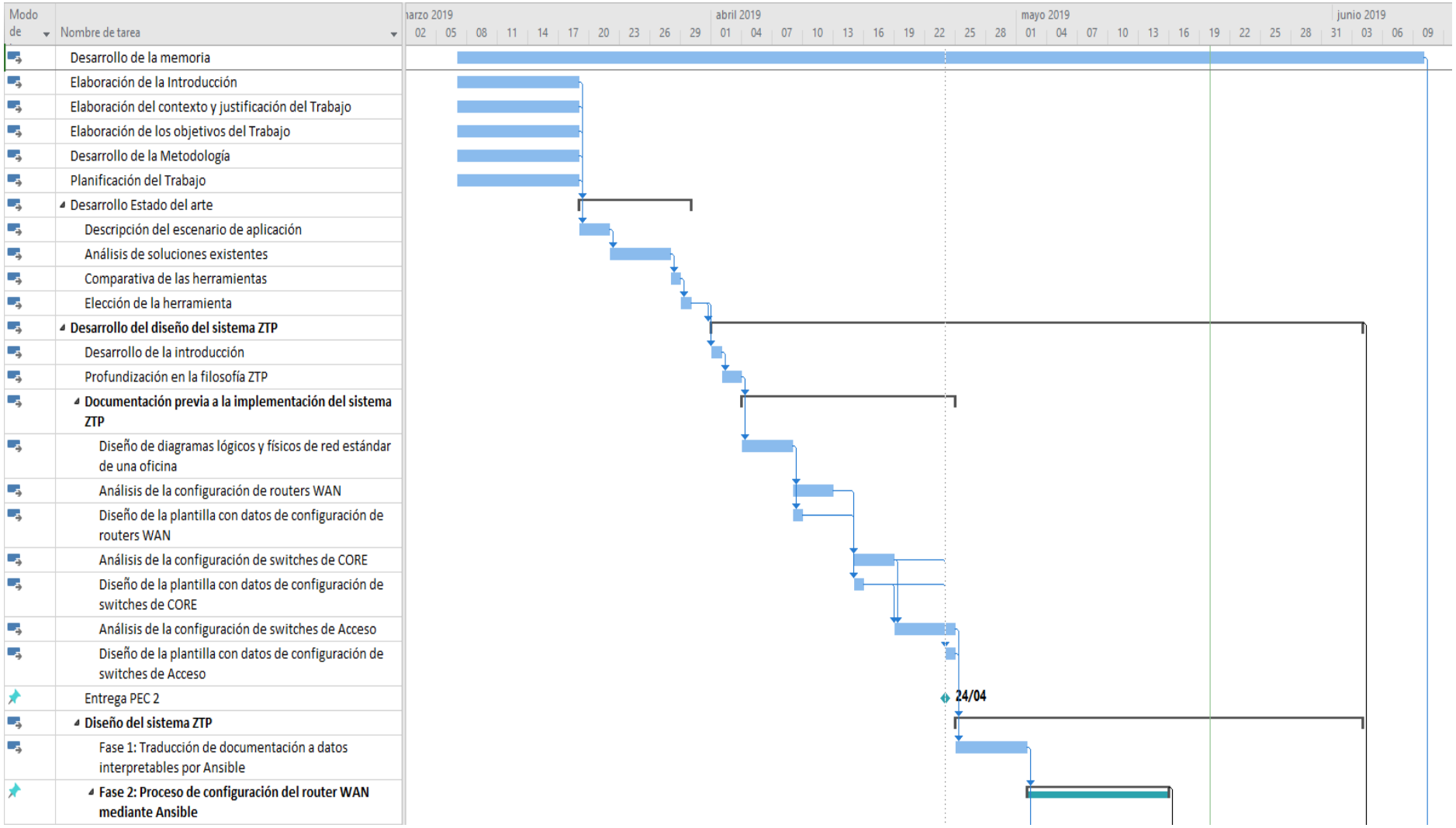

Ilustración 1: Diagrama de Gantt de la planificación. Fragmento 1.

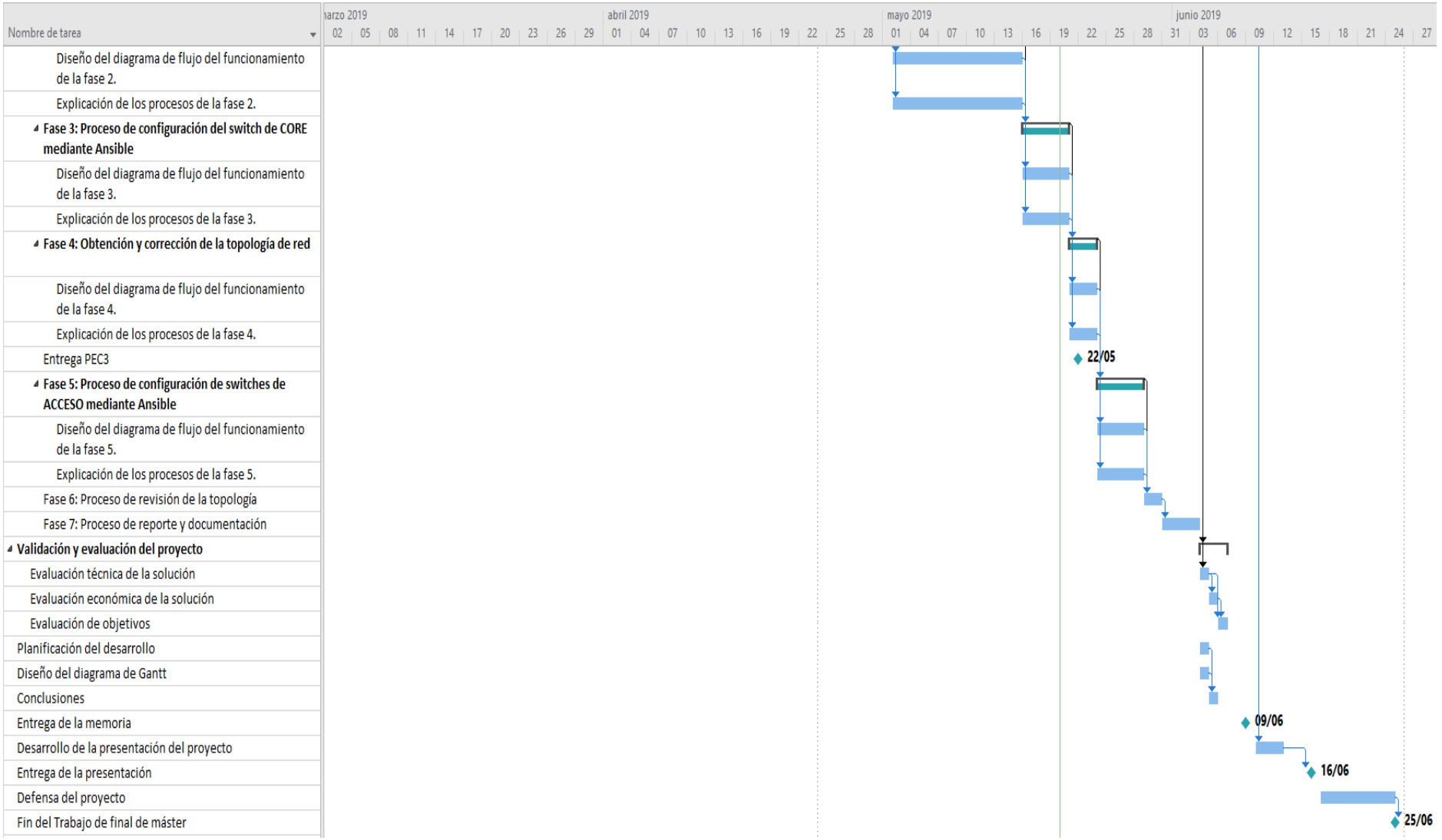

Ilustración 2: Diagrama de Gantt de la planificación. Fragmento 2.

2. Estado del arte

2.1 Análisis de soluciones existentes

En el siguiente apartado se analizarán diferentes herramientas que se pueden encontrar en el mercado para aplicar Automatización y Orquestación en una empresa.

2.1.1 Puppet

Puppet (Ilustración 3: Logo de Puppet) es una compañía que ofrece una herramienta que permite automatizar procesos informáticos mediante un software con el mismo nombre. Dicho software de automatización es "open-core", es decir, es un software de código abierto que se comercializa.

Fue fundada en 2005 por Luke Kanies y su software es capaz de gestionar configuraciones de equipos Unix y Windows (entre otros) usando lenguaje declarativo[2].

Ilustración 3: Logo de Puppet

Puppet (16. (2018). Puppet) es un sistema cliente-servidor, lo que quiere decir que el dispositivo a gestionar debe tener un agente instalado que le permita comunicarse con el servidor. Sin embargo, en algunas soluciones no es necesario este agente.

En lo que a automatización de redes se refiere, Puppet) (14. (2019). Solutions) ofrece una herramienta "multivendor" que permite aplicar de manera homogénea la configuración que se necesite. A continuación, se describen las características del software en este ámbito:

[2] El lenguaje declarativo (28. (2005). Lenguaje declarativo) es un tipo de lenguaje de programación donde no se describe cómo hay que hacer algo, sino que describe qué se quiere hacer u obtener. Define estados y se basa en el lenguaje imperativo, en el que sí se define cómo hay que hacer las cosas.

- Contiene módulos predefinidos para gestionar los equipos de red de los fabricantes más populares como Cisco (15. (2019). Cisco IOS), Lenovo, Palo Alto Networks, Arista, Huawei, y Cumulus.
- Es una solución que no necesita instalar ningún agente.
- Se integra con Puppet Enterprise, lo que permite ver los equipos de red como cualquier otro nodo del parque.

En el ámbito económico Puppet ofrece 3 productos:

- El producto Puppet Discovery permite monitorizar los nodos y se puede adquirir por 25$ por nodo al año.
- Puppet Enterprise, que es un producto de automatización de nodos y se puede probar gratuitamente para 10 nodos, aunque no se publica cuál es el precio para entornos más grandes.
- Por último, ofrece el producto Puppet Pipelines que está pensado para el desarrollo de software y éste tampoco tiene precio público.

2.1.2 Chef

Chef (Ilustración 4: Logo de Chef) es una compañía que comercializa un software de "gestión de configuraciones" con el mismo nombre. Actualmente, es un software de tipo "open-core", es decir, ofrece unas características limitadas en su versión de código abierto, pero alcanza todo su potencial en versiones de pago. En el transcurso de la redacción de esta memoria, desde abril de 2019, Chef ha comenzado a ofrecer el 100% de su software bajo una licencia open-source.

Ilustración 4: Logo de Chef

Chef (17. (2019): Chef) fue creada por Adam Jacob para cubrir ciertas necesidades de su empresa, que estaba enfocada a crear herramientas para servidores "end-to-end". Al descubrir el potencial de la herramienta, se asoció con algunos compañeros y comenzó a comercializar el software.

Todo lo relacionado con Chef gira en torno a la cocina, empezando por el nombre, Chef, donde se pueden desarrollar "cookbooks" (libros de cocina) que incluyan "recipies" (recetas), que al fin y al cabo serán los scripts con lenguaje declarativo que harán lo necesario para automatizar las configuraciones.

Es importante destacar que Chef funciona mediante la instalación de un agente en los equipos a configurar. Esto ofrece muchas ventajas, entre las que se encuentra la independencia respecto a un nodo central. Sin embargo, tener que desplegar un agente es algo bastante intrusivo, por no hablar de que habrá equipos en los que no se pueda instalar dicho agente.

En lo que a la automatización de redes se refiere, Chef tiene algún que otro "cookbook", entre los que podemos encontrar módulos de Cisco IOS, aunque parece que su apuesta son los equipos Cisco Nexus. Sin embargo, no existen muchos más módulos y no se ha encontrado nada al respecto de Fortinet, salvo algún desarrollo de usuario particular que, por tanto, no está mantenido.

En lo relativo al precio, Chef ofrece su producto en Enterprise Automation Stack por 35.000$ al año en su versión Essentials y por 150.000$ al año en su versión Enterprise.

2.1.3 Ansible

Ansible (Ilustración 5: Logo de Ansible) es una plataforma de software libre de configuración y administración de dispositivos. Fue creada por Michael DeHaan, que lanzó la primera versión en febrero de 2012.

Entre otras cosas, Ansible (10. (2019)) permite aprovisionar configuraciones, desplegar aplicaciones, desplegar medidas de seguridad y orquestar dichos despliegues.

Ilustración 5: Logo de Ansible

Al igual que Chef y Puppet, mediante Ansible se pueden desarrollar scripts (Playbooks) que utilizan lenguaje declarativo y que, tras ejecutarlos, editan simultánea y automáticamente las configuraciones de diferentes equipos.

El diseño de Ansible tiene las siguientes características:

- Mínimo por naturaleza. Los sistemas de administración automática no deben imponer dependencias adicionales. Ansible necesita pocas aplicaciones extra para funcionar.
- Consistente. Ansible solo aplica las configuraciones si no están realizadas. Si ya son correctas, no hace nada e informa.
- Seguro. Ansible no instala agentes vulnerables en los nodos. Solamente se requiere OpenSSH que es considerado crítico y altamente comprobado.
- Alta confiabilidad. El modelo de idempotencia es aplicado para las instalaciones y configuraciones, para prevenir efectos secundarios en la ejecución repetitiva de Playbooks (scripts).
- Suave curva de aprendizaje. Los Playbooks o libretos usan un lenguaje descriptivo simple, basado en YAML.

En cuanto a los módulos, podemos encontrar una gran cantidad de ellos, siendo los siguientes los que nos interesa estudiar, dado el escenario (13. (2019). Network modules):

- **cli**
  - cli_command: ejecución de comandos cli en cualquier tipo de dispositivos basado en CLI.
  - cli_config: carga de configuraciones en elementos de red compatibles con network_cli.

- **Fortimanager**
  - fmgr_provisioning: provisión de equipos vía FortiManager.
  - fmgr_script: ejecución de scripts de FortiManager.

- **Fortios**
  - fortios_address: gestión de objetos del tipo “address” en fortios.
  - fortios_config: gestión de configuraciones en fortios.
  - fortios_ipv4_policy: gestión de políticas IPv4 en firewall con Fortinet.
  - fortios_webfilter: configuración de restricciones y filtros web en equipos Fortinet.

- **IOS**
  - ios_banner: gestión de banners multilínea en equipos Cisco IOS.
  - ios_command: ejecución de comandos en equipos remotos basados en Cisco IOS.

- ios_config: gestión de partes de configuración en equipos Cisco IOS.
- ios_facts: recolección de datos de dispositivos Cisco IOS.
- ios_interface: gestión de interfaces en equipos Cisco IOS.
- ios_l2_interface: gestión de interfaces de nivel 2 en equipos Cisco IOS.
- ios_l3_interface: gestión de interfaces de nivel 3 en equipos Cisco IOS.
- ios_linkagg: gestión de interfaces de agregación en equipos Cisco IOS.
- ios_lldp: gestión del protocolo de descubrimiento basado en la capa de enlace de Cisco IOS.
- ios_logging: gestión de logs en equipos Cisco IOS.
- ios_ping: comprobación de la alcanzabillidad de equipos remotos que se basan en Cisco IOS.
- ios_static_route: gestión del encaminamiento estático en equipamiento Cisco IOS.
- ios_system: gestión de los atributos de un sistema basado en Cisco IOS.
- ios_user: gestión de los usuarios de equipos remotos basado en Cisco IOS.
- ios_vlan: gestión de VLANs en equipos Cisco IOS.
- ios_vrf: gestión de la colección de VRF[3] de equipos basados en Cisco IOS.

Por otro lado, Redhat ha desarrollado una interfaz web que permite gestionar los “Playbooks” y los servidores. Se comercializa bajo el nombre de Ansible Tower (11. (2019). Ansible Tower).

Si bien Ansible es un software gratuito, Ansible Tower es un producto que requiere una suscripción anual, ofreciendo varias modalidades:

- El servicio estándar de Ansible Tower permite gestionar hasta 100 equipos y ofrece un soporte de 8x5, teniendo un coste de 10.000$ al año.
- El servicio Premium permite gestionar el mismo número de nodos pero ofrece un soporte 24x7, teniendo un coste de 14.000$ al año.
- Si adicionalmente se quiere contratar el producto Ansible Engine, las cifras ascienden hasta los 13.000$ y 17.500$ respectivamente. Ansible Engine es la comercialización del software libre Ansible

[3] Siglas de Virtual Routing and Forwarding. Se utiliza para dividir los recursos físicos de un equipo Cisco en diferentes equipos lógicos.

bajo un contrato de soporte que ofrece a las compañías cierta tranquilidad y estabilidad en torno a la herramienta.

En paralelo, la herramienta Ansible Tower se ha liberado bajo una licencia de software libre que ha generado un proyecto llamado AWX (12. De la Cruz, J (2018)). Viene a ser lo mismo que Ansible Tower, pero en versión libre y soportada por una comunidad de desarrolladores.

2.1.4 Comparativa de las herramientas

A continuación, se comparan diferentes parámetros de las tres herramientas. Posteriormente, se analizarán cada uno de estos parámetros.

	Puppet	**Chef**	**Ansible**
Lenguaje de programación	C++, Clojure	Ruby, Erlang	Python
Método de transporte	Mcollective	RabbitMQ	SSH
Agente	Sí	Sí	No
Arquitectura	Cliente-servidor	Cliente-servidor	Servidor
GUI empresarial	Puppet Enterprise		Ansible Tower
GUI opensource	Foreman	Chef Manager	AWX
Estrellas en Github	5.243	5.742	36.431
Forks	2.076	2.313	15.050
Contribuyentes	513	563	4.330
Commits	30.004	23.428	44.372

2.1.4.1 Lenguaje de programación

En este concepto se hace referencia al lenguaje que se ha utilizado para construir la herramienta. Esto no implica, en sí mismo, ninguna ventaja o desventaja, aunque sí es cierto que C++ y Python son lenguajes más vivos o conocidos que el resto.

2.1.4.2 Método de transporte

Tanto Puppet como Chef utilizan un método de transporte específico para el software. Ansible, en cambio, utiliza Secure Shell (SSH) lo que le permite conectarse a cualquier máquina Windows (con powershell), Linux o con cualquier equipamiento que permita SSH, como, por ejemplo, Cisco IOS (9.Kashin, M (2015)).

2.1.4.3 Agente y arquitectura

En este aspecto, Puppet y Chef vuelven a coincidir. Ambas herramientas exigen que los dispositivos a gestionar tengan un agente instalado, lo que las convierte en una arquitectura cliente-servidor. Puppet requiere su versión empresarial para controlar dispositivos de red (como Cisco NX-OS), mientras que Chef permite que la versión de código abierto de su software haga lo mismo.

Sin embargo, Ansible admite la gama más amplia de agentes: como se indicó anteriormente, siempre que un cliente o dispositivo pueda ejecutar SSH (o Powershell), puede ser controlado por Ansible. Por supuesto, el software de control (en forma de módulos) para dicho dispositivo debe estar disponible o se puede diseñar desde cero.

En lo que a la red respecta, además de admitir los sistemas operativos tradicionales basados en Linux, también admite Cisco IOS y Juniper JunOS de forma nativa. Las otras dos herramientas utilizan sus propios agentes propietarios, pero dichos agentes están disponibles para todas las plataformas informáticas comunes. Puppet requiere su versión empresarial para controlar dispositivos de red (como Cisco NX-OS), mientras que Chef permite que la versión de código abierto de su software haga lo mismo.

2.1.4.4 Github

Las tres herramientas tienen su proyecto en Github, aunque solo Ansible es 100% open source. Puede que precisamente por eso sea la más apoyada, la que más contribuyentes tiene y la que más ha evolucionado desde que está publicada. En este aspecto, la diferencia respecto a las otra dos, es abismal.

2.2 Elección de la herramienta

Tras el análisis de los parámetros de las herramientas objeto de estudio, se selecciona Ansible para el desarrollo del proyecto. Los motivos de la elección se desarrollan a continuación:

Que sea una herramienta sin agente es algo excepcionalmente importante cuando se trata de automatización de red, sobre todo si hablamos de equipamiento de una red que ya existe. Si consultamos qué tipos de dispositivos tienen instalados las diferentes compañías podremos observar que la mayor parte de esos equipos no tienen una API moderna, sino que se configuran vía línea de comando (CLI). Está claro que tener una API hace que todo sea mucho más simple de gestionar, pero una plataforma “agentless” como Ansible se integra a la perfección con estos dispositivos tradicionales y también con los más modernos, por lo que la convierte en una herramienta válida para cualquier entorno de red.

A medida que los dispositivos de red como routers, switches y firewalls continúan implementando soporte para las API, también están surgiendo soluciones de redes definidas por software (SDN, software defined networks). El único aspecto común con las soluciones SDN es que todas ofrecen un punto único de integración y administración de políticas, generalmente en forma de un controlador SDN. Esto se aplica a soluciones como Cisco ACI, VMware NSX, Big Switch Big Cloud Fabric y Juniper Contrail, así como a muchas de las otras ofertas de SDN de compañías como Nuage, Plexxi, Plumgrid, Midokura y Viptela. También incluye controladores de código abierto como OpenDaylight.

Todas estas soluciones simplifican la administración de redes, porque permiten que un ingeniero de comunicaciones cambie del “workflow” de la gestión de “caja a caja” a la administración de un solo sistema en toda la red. Si bien éste es un gran paso en la dirección correcta, estas soluciones aún no eliminan los riesgos de error humano durante las ventanas de cambio. Sin embargo, la necesidad de automatizar la gestión de las redes no desaparece incluso cuando el mundo tiende a la gestión basada en controladores SDN.

Todos estos controladores modernos suelen ofrecer una API REST. Como Ansible tiene una arquitectura sin agente, hace que sea extremadamente sencillo automatizar no solo los dispositivos heredados que pueden no tener una API, sino también las soluciones de red definidas por software a través de las API REST, todo sin requerir ningún software

adicional. El resultado es poder automatizar cualquier tipo de dispositivo utilizando Ansible con o sin una API.

Otro punto fuerte de Ansible es que es un software de código abierto, lo que lo convierte en una herramienta totalmente gratuita. Más aún, es importante destacar que Ansible (Ansible, Inc, en concreto) es una compañía y ofrece el producto Ansible Tower, que agrega funciones como el control de acceso basado en roles (RBAC), informes, interfaz de usuario web, API de REST, "multitenancy" y mucho más, lo que suele ser una buena opción para empresas que buscan implementar Ansible. Aun así, como ya hemos comentado anteriormente, hace poco se liberó el código fuente de Ansible Tower que ha generado un proyecto llamado AWX que ofrece casi toda la potencia de Ansible Tower, pero de manera abierta y gratuita.

Por otro lado, es necesario recalcar el grandísimo apoyo que tiene Ansible por parte de la comunidad. En su origen, Ansible estaba pensado para administrar entornos que usaran sistemas operativos basados en Linux, aunque al de muy poco tiempo también se empezó a integrar con entornos Windows. La comunidad que hay detrás de la herramienta se ha dado cuenta de su flexibilidad y versatilidad, por lo que hace ya unos años que algunos desarrolladores independientes empezaron a enfocarse en el mundo de las redes, destacando Matt Oswalt, Kirk Byers, Elisa Jasinska, David Barroso, Michael Ben-Ami, Patrick Ogenstad, Gabriele Gerbino o Jason Edelman. También ha habido fabricantes que se han sumado al desarrollo: Arista, Juniper, Cumulus, Cisco, F5 o Palo Alto Networks, entre otros.

Asimismo, un valor que ni Puppet ni Chef parecen poder ofrecer de momento es la transversalidad de Ansible. Si bien este proyecto se enfocará en la automatización de redes, Ansible es muy competente en la administración de otros entornos lo que llegaría a permitir crear scripts (Playbooks en este caso) que configuren equipos de cualquier tipo de naturaleza (servidores, switches, routers, firewall, teléfonos, sistemas de seguridad…).

Referencias: Fuentes: 1. Suripa, K (2016), y 5. Edelman, J (2016).

2.3 Contribución del trabajo frente a soluciones existentes

Este trabajo ofrece ventajas claras respecto a las soluciones que se pueden encontrar en el mercado.

Por un lado, ofrece un planteamiento DIY[4], lo que permite un diseño a medidas según las necesidades de la compañía. Si este servicio fuera contratado a un fabricante, tendría los siguientes inconvenientes:

- El fabricante te obligaría a usar sus equipos físicos (routers, switches)
- Ante problemas muy particulares, el fabricante podría no ofrecer ninguna solución.
- Se ha de pagar por la implementación y el mantenimiento del servicio.

No obstante, diseñar el sistema por cuenta propia, tiene la desventaja de que se tienen que invertir cientos o miles de horas hasta conseguir la solución.

Por otro lado, en internet se pueden encontrar varias soluciones de automatización orientadas a la automatización de redes, pero todas están realizadas por ingenieros independientes que han personalizado su solución para su problema particular. Sin embargo, este trabajo tiene la vocación de ofrecer una solución lo más universal posible. Por supuesto, todas las aportaciones de estos ingenieros independientes servirán de orientación para el desarrollo, pero el objetivo principal de este trabajo es ofrecer un diseño que pueda cubrir las necesidades de cualquier compañía y que oriente claramente cómo se ha de implementar el sistema.

Por último, cabe destacar que se han encontrado escasas referencias académicas que hablen sobre el tema de la automatización de redes. Como se ha comentado, se pueden encontrar muchos artículos de ingenieros independientes pero muy pocos sobre ingenieros de redes que se dediquen al ámbito docente. Esto también ha sido motivo para realizar este trabajo y arrojar algo de luz sobre un tema no muy recurrente en el mundo académico.

[4] DIY (Do It Yourself), hazlo tú mismo.

3. Desarrollo del Sistema ZTP

En los siguientes capítulos se desarrollarán los diferentes aspectos del diseño de un sistema “Zero Touch Provisioning”. Se pretende diseñar, de manera teórica, un sistema que sea capaz de aprovisionar toda la configuración necesaria para crear la arquitectura de red de una oficina empresarial pequeña (sede filial) de manera automática. Todo este desarrollo se apoyará sobre los artículos de estos ingenieros de redes:

18. Miloslavsky, A (2017).
19. Dainese, A (-).
20. Pepelnjak, I (2018).
21. Pepelnjak, I (2017).
22. Ogenstad, P (-).
23. Ogenstad, P (2018).

Para ello, el trabajo se basará en un entorno corporativo real, en el cual es habitual que surja la necesidad de montar una oficina corporativa para que usuarios desplazados a una ubicación puedan trabajar con normalidad y acceder a los recursos corporativos.

El siguiente diagrama, Ilustración 6: Visión general de un Sistema ZTP, plantea de manera general el sistema que se diseñará:

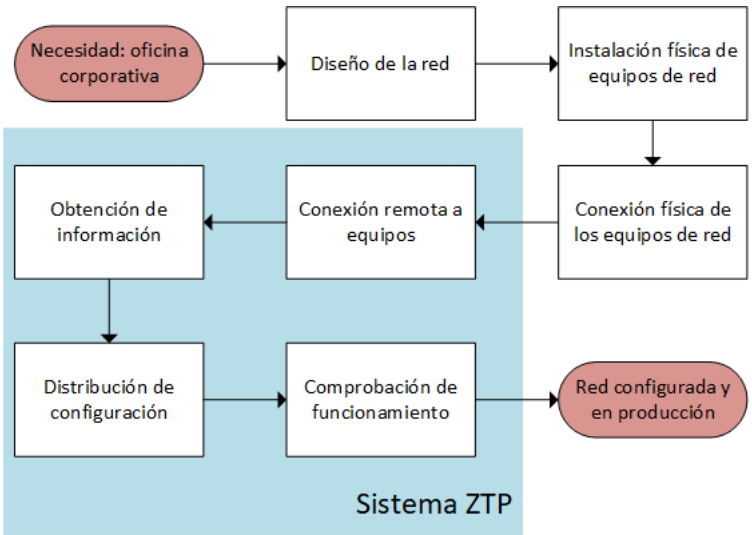

Ilustración 6: Visión general de un Sistema ZTP

3.1 Descripción del escenario

En los siguientes apartados se describe la arquitectura de red que se utilizará para orientar el diseño de la solución.

3.1.1 Perspectiva WAN

La WAN (Wide Area Network) hace referencia a un entorno de red amplio que suele abarcar ciudades, países… En el caso de este proyecto la WAN hace referencia al conjunto de redes que interconectan cada uno de las oficinas de la compañía que se propone como modelo.

A nivel de red, la compañía real en la que se basará el diseño del sistema está formada por dos sedes principales (headquarters) y varias sedes filiales. Ambas sedes centrales tienen dos routers de agregación para interconectarse mientras que las sedes filiales pueden tener dos routers de EDGE[5], un único router de EDGE, o pueden estar conectadas a la WAN[6] directamente desde el router de CORE[7]. En el siguiente diagrama, Ilustración 7: Resumen de la arquitectura WAN, se pueden observar routers de EDGE y de agregación desde una perspectiva WAN:

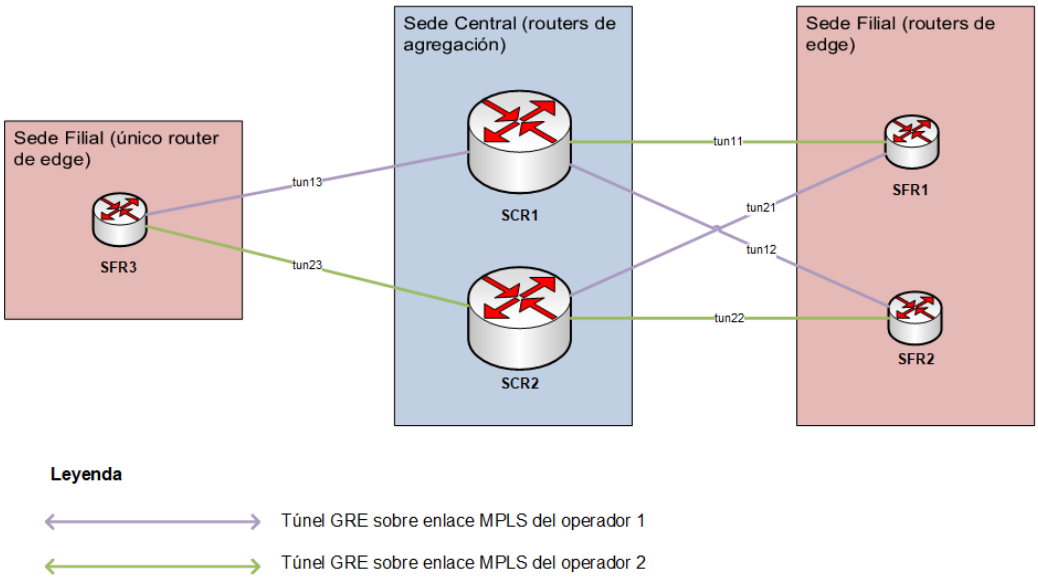

Ilustración 7: Resumen de la arquitectura WAN

Los routers de EDGE son los routers que se instalan en el límite de la red interna de una oficina. Estos routers son los que interconectan la red interna con otras redes externas (redes de operador, por ejemplo) y con

[5] Router que se encuentra en el perímetro de la red local, en el borde.
[6] Wide Area Network, hace referencia a una red que interconecta diferentes redes locales (LAN).
[7] Router central o troncal de una red local.

otras oficinas. Los routers de agregación también son routers de EDGE pero tienen una peculiaridad: son los nodos centrales de la WAN donde se concentran (o agregan) todas las conexiones con sedes filiales.

En la Ilustración 8: Routers de EDGE desde la perspectiva LAN se aprecian los elementos descritos:

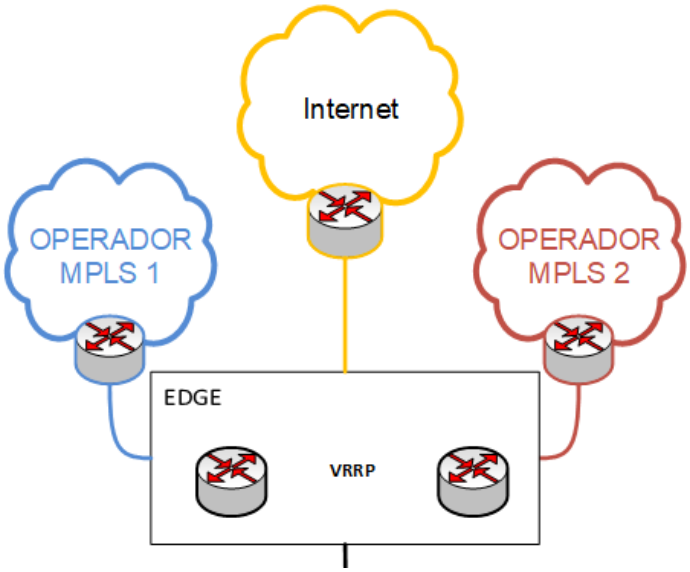

Ilustración 8: Routers de EDGE desde la perspectiva LAN

Volviendo a la perspectiva WAN de la compañía, se puede decir que está formada por dos estrellas unidas, en concreto, dos dobles estrellas. Esta puntualización se debe a que las sedes centrales tienen dos routers de agregación cada uno, por lo que las estrellas se duplican. Sin embargo, en la siguiente Ilustración 9: diagrama de red WAN se aprecia la WAN de una compañía real, aunque por simplificar solo se han dibujado las estrellas simples. Apréciense dos nodos centrales y múltiples enlaces que conectan con las diferentes oficinas filiales.

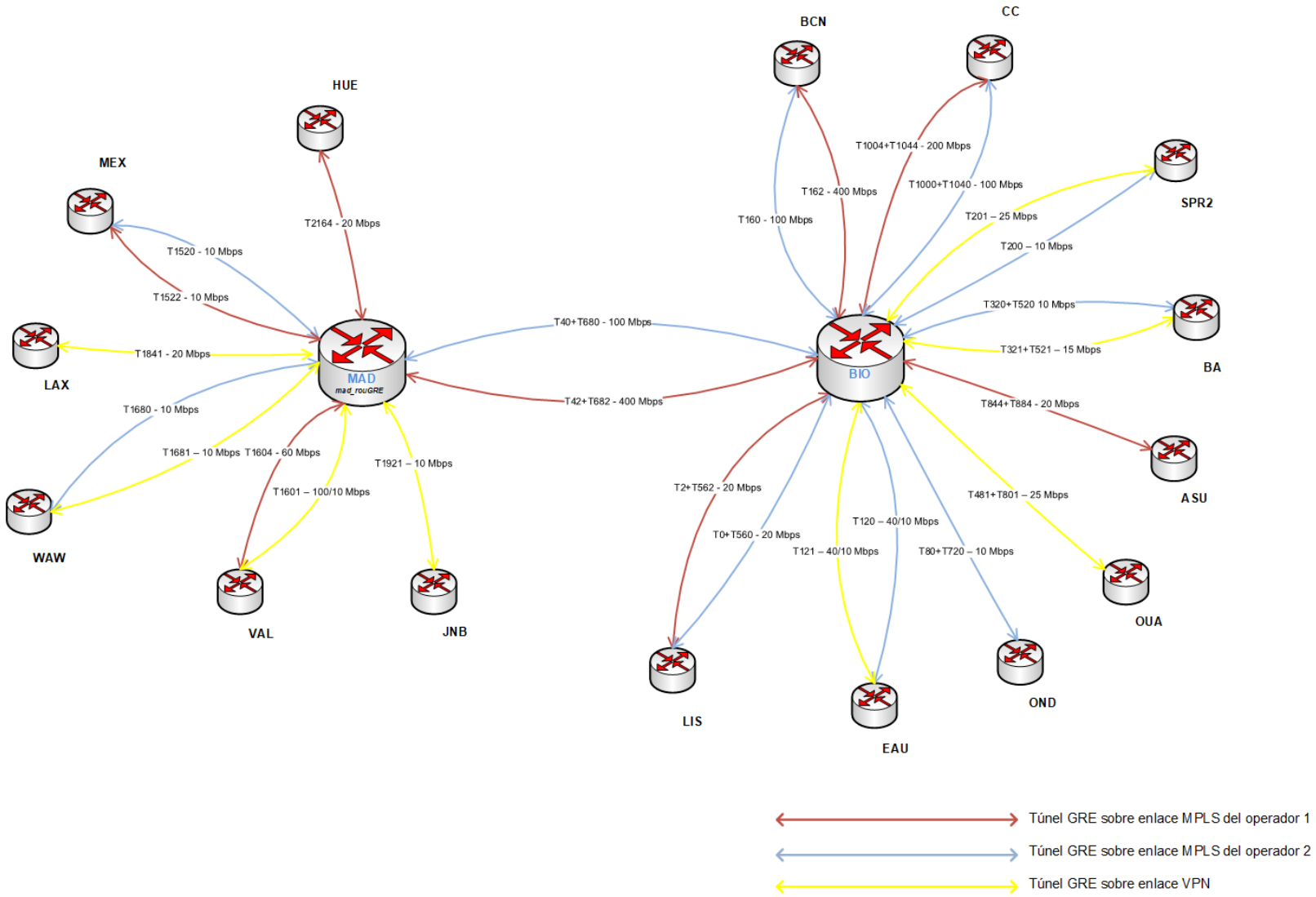

Ilustración 9: diagrama de red WAN

Las sedes filiales se conectan a través de sendos túneles GRE a las sedes centrales. Estos túneles GRE está configurados sobre redes MPLS[8] de diferentes operadores o sobre VPN site-2-site que se conectan a través de internet. La razón de estos túneles GRE es tener una capa de abstracción que permita interconectar las diferentes oficinas independientemente de la tecnología que tengan para conectarse (MPLS, internet...)

3.1.2 Perspectiva LAN

La LAN (Local Area Network) hace referencia al conjunto de redes locales de una ubicación. En el escenario propuesto, LAN hace referencia a las redes locales de las que dispone cada una de las sedes.

Los routers CORE son los routers centrales de una oficina y es donde se interconectan sus diferentes redes locales. Es una práctica habitual tener diferentes VLANs (redes) para segmentar la red (Ilustración 10: Router de CORE desde la perspectiva LAN). Es frecuente tener VLANs dedicadas a los usuarios, otras a los servidores, otras a equipos de red… Hacer esto provoca que el tráfico de una red LAN tenga que pasar por el CORE para alcanza otras redes locales.

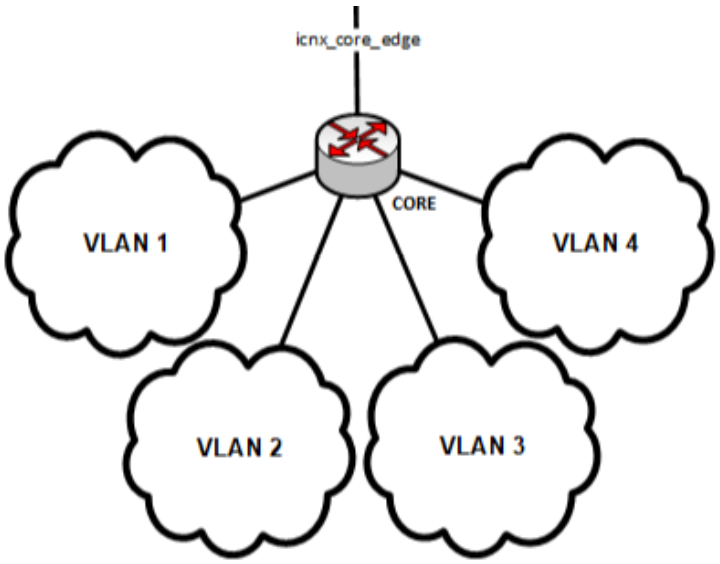

Ilustración 10: Router de CORE desde la perspectiva LAN

[8] MultiProtocol Label Switching, es un protocolo de red que ofrece un operador de red para tener un WAN privada.

Desde una perspectiva local de una red se puede observar el siguiente ejemplo de red corporativa en la Ilustración 11: Diagrama lógico de la red LAN de una sede filial.

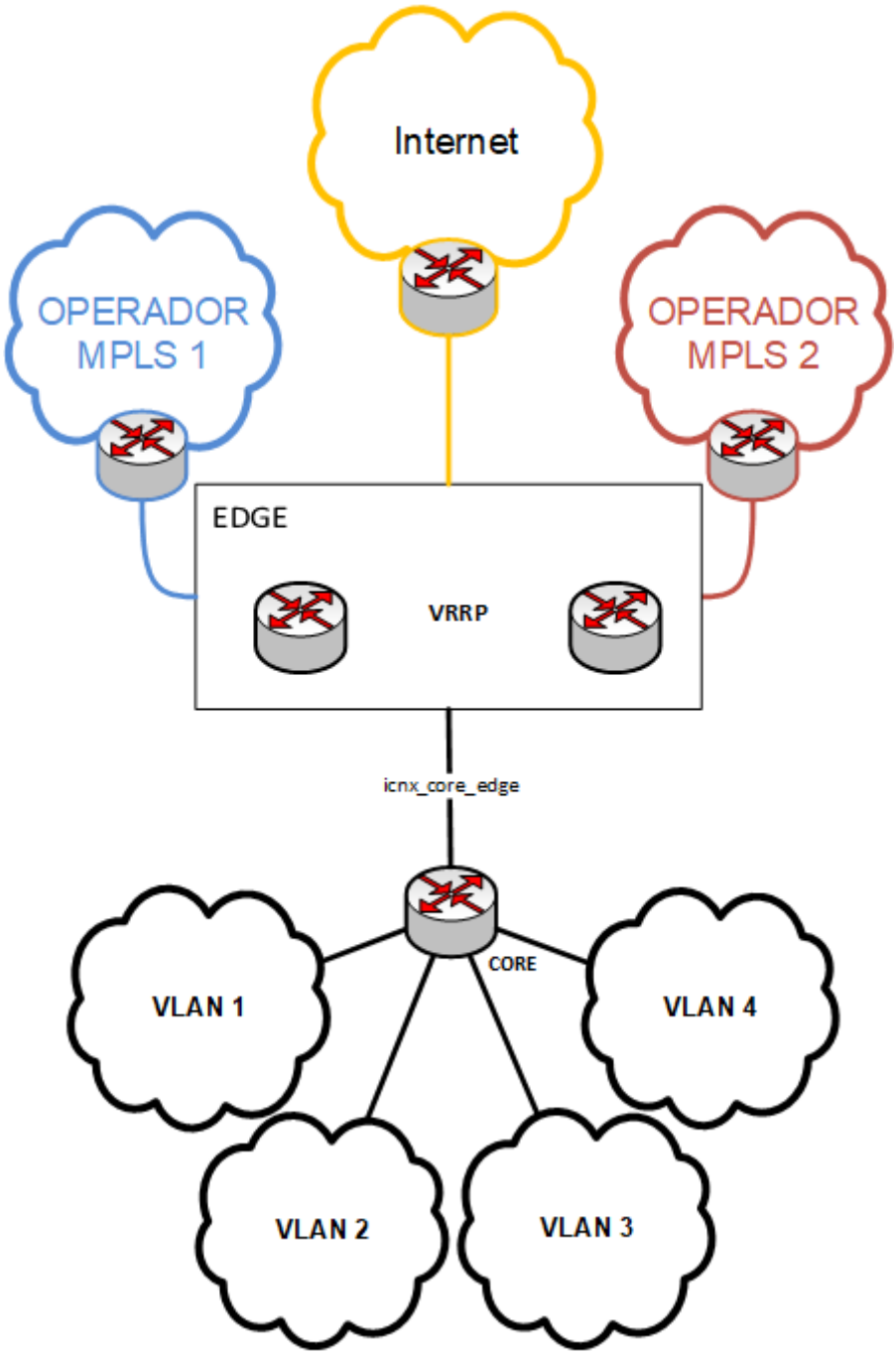

Ilustración 11: Diagrama lógico de la red LAN de una sede filial.

Lo habitual es que dependiendo del número de usuarios que albergan una sede y, por tanto, dependiendo del presupuesto de que ésta disponga, prescindan de ciertas prestaciones y de guardas por redundancia, hasta

el punto de convertirse en una filial con un router de CORE, un switch de ACCESO[9] y un router de operador. En el ejemplo anterior, se muestra la imagen lógica de una red LAN de una sede filial habitual.

Sin embargo, una sede central que albergará cientos de usuarios y de servicios tiene un aspecto mucho más complejo. Se puede observar en el anexo 7.1 un diagrama de ejemplo de una sede central donde todo está por duplicado y se pueden apreciar otros elementos relacionados con la seguridad perimetral de la red.

En lo que a la arquitectura física respecta, todas las sedes tienen una estructura similar. Todos los elementos de la red se conectan físicamente al router/switch de CORE. Entre esos elementos, se encuentran los switches de ACCESO, donde se conectarán los PCs de usuarios y otros tipos de dispositivos finales. Obsérvese la Ilustración 12: Diagrama físico de la red de una oficina.

[9] Switch donde se conectan los equipos finales (PCs, impresoras, teléfonos…) y reciben acceso a la red.

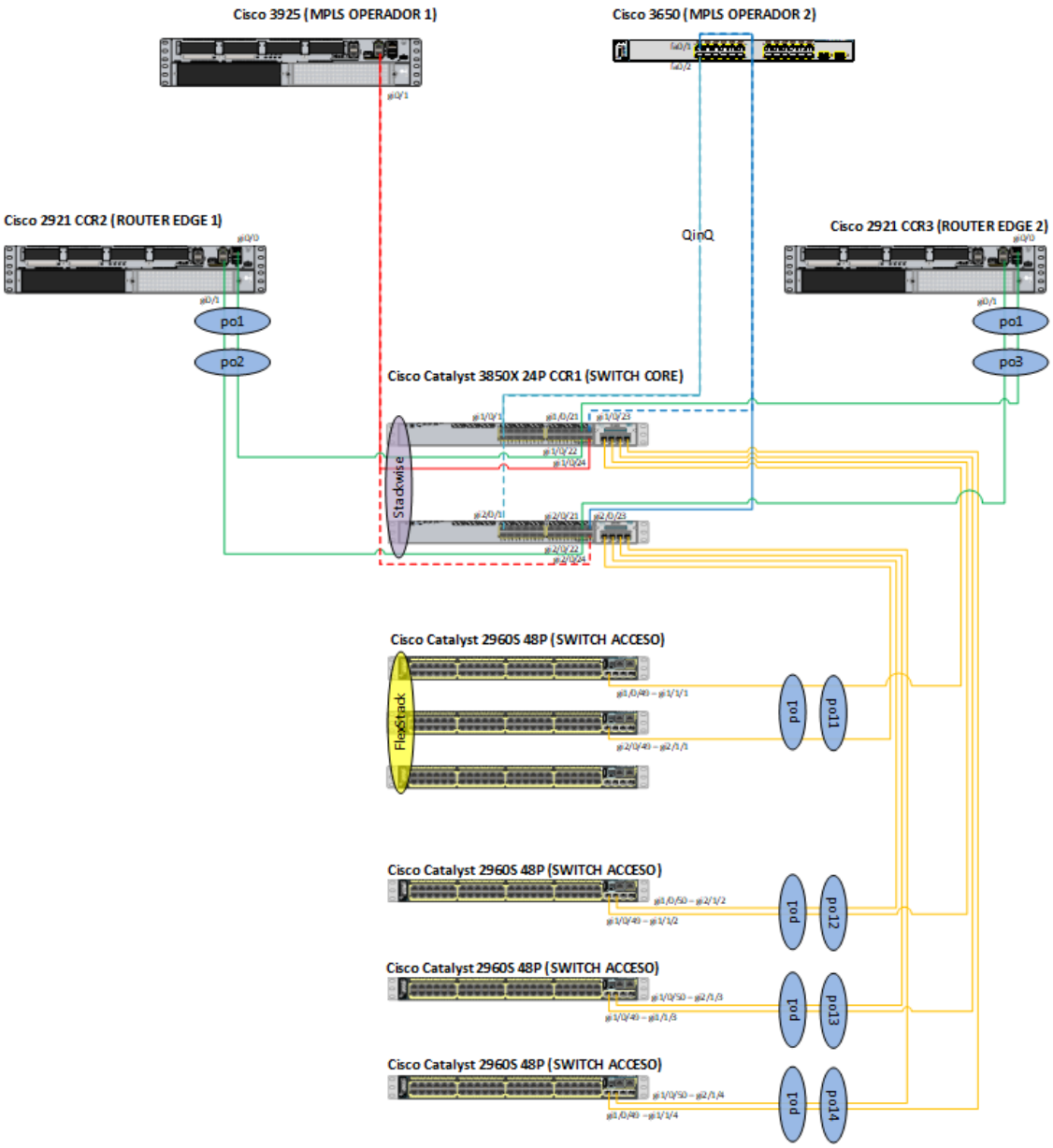

Ilustración 12: Diagrama físico de la red de una oficina

Como se aprecia en el diagrama, todos los elementos se conectan a dos switches de CORE centrales, representados por dos Cisco Catalyst 3850X. Es importante diferenciar entre un switch de CORE y un router de CORE. Un switch de CORE es un switch de nivel 3 capaz de hacer routing y que también tiene puertos para poder usarse en modo switch. Un router por el contrario, es solo un router y no puede usarse en modo switch. Es habitual usar switches de nivel 3 como elemento central de las redes locales.

Cabe destacar que se utilizan dos switches Cisco Catalyst 3850X que actúan como un único switch. Ambos equipos forman un único “stack” porque están conectados por dos cables “Stackwise”. Como ambos

switches forman uno solo, la gestión es más sencilla pero el servicio es mucho más completo:

- Ofrece redundancia del servicio si un “cae”.
- Se dispone de más puertos para conectar lo que sea necesario.
- Completa la redundacia de los equipos de acceso.

Apréciese también que los switches de acceso, representados en este caso por switches Cisco Catalyst 2960S, están conectados por dos caminos diferentes a los switches de CORE. Esto asegura la conectividad de los switches por si alguno de los enlaces falla.

Por otro lado, se puede apreciar que los switches de acceso que están inmediatamente después de los switches de CORE están apilados. Estos tres switches funcionan, al igual que el CORE, como un único switch. Este apilamiento ofrece las mismas mejoras que las comentadas para los COREs, pero se utiliza otra tecnología para unirlos. En este caso, por ser switches C2960S utilizan cables “Flexstack” para conectarse entre ellos.

3.1.3 Marcas y dispositivos.

A continuación, se listan las marcas que se utilizan en cada uno de los tipos de dispositivos que conforman la red.

- Routers: Cisco IOS.
- Switches: Cisco IOS.
- Firewalls: Fortinet o Cisco ASA.
- Puntos de Acceso WiFi: Ubiquiti.
- Aceleradores WAN: Riverbed.

3.1.4 Descripción del escenario para el diseño

Para que el diseño del sistema esté bien acotado, a continuación, se describirá el entorno real sobre el cual se planteará el diseño, ya que se han mencionado varios tipos de arquitecturas.

La oficinal filial que se utilizará como referencia para diseñar el sistema tendrá los siguientes elementos:

- Un router de operador para conectar a la WAN de la compañía.

- Dos switches de CORE Cisco 3850 en stack de 16 puertos cada uno.
- Dos pilas de acceso de dos switches Cisco 2960 en stack cada una. Esos switches serán de 48 puertos cada uno.
- El CORE y las 2 pilas de acceso estarán conectadas por 4 enlaces de fibra. Los dos enlaces que conecten con la primera pila pertenecerán a una agregación de puertos y los otros dos a otra agregación de puertos.

3.2 Introducción al Zero Touch Provisioning

ZTP (Zero Touch Provisioning) hace referencia al concepto de otorgarle a un equipo una configuración inicial para que comience a funcionar de forma automática. Evidentemente, esta es una tarea habitual entre los administradores de red, ya sea porque se monta una oficina nueva, se hace alguna ampliación o porque se instala un nuevo servicio que requiere un cambio de infraestructura.

Cuando un dispositivo de red se arranca por primera vez, no tiene ninguna configuración. Para que sea utilizable, alguien tiene que conectar un cable de consola al hierro[10] y aplicar una configuración antes de que sea alcanzable en una red.

Un proceso como éste puede requerir mucho tiempo, especialmente si el número de equipos es amplio. Incluso puede significar que alguien se tenga que desplazar a una ubicación remota para configurar los equipos y conectarlos uno a uno. En estos casos hay que contemplar lo que cuesta el viaje del técnico o el coste de contratar a algún técnico local de la zona.

Las ventajas de un ZTP automatizado son muchas y se describen a continuación:

- Elimina la necesidad de enviar un técnico a configurar “on-prem”[11] los diferentes equipos.
- Permite la configuración de todos los equipos en el mismo ejercicio de provisión.
- Evita errores humanos.

[10] Notación coloquial usada por técnicos de TI para referirse a un equipo por su cobertura (habitualmente metálica).

[11] Abreviación de “on-premises”, significa que se halla en la ubicación donde se encuentra el equipo o el usuario.

- Se despliegan configuraciones homogéneas favoreciendo la comprensión de la configuración y la estandarización de la misma.

Para desarrollar un proyecto de implementación de un sistema ZTP habrá que empezar por redactar todos y cada uno de los pasos manuales que hay que acometer para configurar un equipo. Evidentemente, no es lo mismo configurar un switch, un router, o un firewall, por lo que las posibles configuraciones serán muy diversas. Por tanto, el esfuerzo inicial de análisis será considerable y transcendental para la consecución de una solución adecuada para la empresa en la que se desee implementar el sistema.

Después, habrá que analizar una por una todas las tareas y encontrar la solución que automatice ese paso. Probablemente, Ansible ofrezca un módulo que facilite la automatización del paso y en caso contrario, el propio administrador del sistema deberá crearlo.

Por último, habrá que recoger todos esos pasos y soluciones e integrarlos en uno o varios “Playbooks” de Ansible.

3.3 Profundizando en la filosofía ZTP

Habitualmente, los administradores de redes saben cómo configurar un dispositivo y no suelen necesitar anotar ningún procedimiento para hacerlo. Por ejemplo, si se va a configurar un switch, es común copiar la configuración de un switch similar y pegar las partes que se necesitan en el nuevo y cambiar las partes que no son iguales. Además, si hay alguna configuración que se resiste, se puede acudir a internet y buscar la solución. En otras palabras, los problemas se van afrontando a medida que aparecen. Esto puede requerir mucho tiempo, pero hay que admitir que esta filosofía a la hora de configurar equipos, tiende a funcionar. Además, es habitual que los problemas sean necedades como que se ha olvidado “levantar” un interfaz o de generar las claves de SSH.

El problema de lo que se describe en el párrafo anterior es que esa mentalidad a la hora de configurar equipos no es posible en el mundo de la automatización. Los pasos correctos y precisos de la configuración deben ser pensados con anterioridad, mucho antes de hacer cualquier instalación. Esto puede ser un gran problema si no se tiene certeza de las configuraciones que se precisan o de la topología que se necesita diseñar.

De manera general, podemos decir que un sistema ZTP se puede dividir en 4 grandes fases:

- Hacer que el dispositivo sea alcanzable.
- Actualizar el equipo.
- Realizar configuraciones básicas (por ejemplo, DNS, NTP, SNMP…)
- Realizar configuraciones específicas (VLANs, interfaces, protocolos de encaminamiento…)

Arquitectos de red con gran experiencia en este tema, como Ivan Pepelnjak y Stan Strijakov, han dado unas pinceladas más específicas en este sentido, y destacan los siguientes 5 pasos a nivel general:

1) Paso de preconfiguración

Se trata de generar las configuraciones que necesitaría cada dispositivo en base a los criterios que se hayan elegido. Stan Strijakov, en concreto, ha conseguido generar ficheros YALM[12] (lenguaje de programación de Ansible) a partir de unos ficheros Excel con la información de los switches, las VLANs, las subredes IP, servidores DHCP…

2) Paso de configuración del router

Consistiría en cargar la configuración que se ha generado en el apartado anterior en el router de la oficina. Esta configuración debería contemplar un servidor DHCP temporal, que sería el encargado de otorgar una IP al resto de elementos de la red con la intención de que estos sean alcanzables.

3) Configuración del switch de CORE

Esta parte es muy relevante, ya que es habitualmente de donde cuelgan todos los elementos de una red. En realidad, por eso se llama switch de CORE. La manera ingeniosa de saber cuál es el switch de CORE sería revisando qué switch tiene como vecino CDP[13] el router (o los routers)

[12] Siglas de Yet Another Markup Language, es un lenguaje de serialización de datos legible por humanos.

[13] CDP son las siglas de Cisco Discovery Protocol, un protocolo que permite detectar qué dispositivos están conectados al otro lado de los puertos del router o el switch en el que se esté investigando.

WAN. Una vez detectado cuál sería el switch de CORE, mediante Ansible, se lanzaría la configuración prevista y se desharía la configuración DHCP para configurar un enlace extremo a extremo entre el switch de CORE y el router WAN.

4) Detección de la topología

En este paso, valiéndose del protocolo CDP se descubrirían el resto de switches y se guardaría la información en un fichero YALM.

5) Comprobación de la topología

Mediante un script PHP se dibujaría la topología de red en una pantalla con la intención de que un ingeniero revise toda la información y corrija los errores que hayan podido surgir. Es normal que, por ejemplo, los nombres de los switches que se han asignado automáticamente no estén en el switch exacto en el que debieran estar. Llegado el caso, se corrigen los errores cosméticos y se genera un nuevo fichero YALM con la topología correcta.

3.3.1 Actualización del software de los switches

El servidor de Ansible debería volver a escanear todos los switches ordenándolos por su vecindad CDP. En otras palabras, debería generar grupos de switches por su pertenencia a las diferentes ramas de la topología. En este punto, empezando por los grupos de switches de las ramas más alejadas del CORE, comenzaría a actualizar el software de los equipos hasta llegar a los grupos de switches que cuelgan directamente del CORE. Después, de manera opuesta, empezando por los switches que cuelgan directamente del CORE y terminando por los switches más alejados, iría lanzándoles la configuración prevista en el paso uno y los reiniciaría.

3.3.2 Verificaciones finales

Ansible redescubriría la topología resultante que se debe revisar para verificar que a todos los switches se les haya aplicado la configuración y se hayan creado las nuevas direcciones IP. Algunos de los pasos

anteriores se pueden volver a ejecutar si alguno de los conmutadores no pudo obtener la configuración inicial.

3.3.3 Documentación

En este paso final, Ansible escanearía la red en busca de todos los switches, recopilando los detalles de los vecinos a través de CDP / LLDP y actualizando los nombres de las interfaces del switch con el switch conectado o el nombre punto de acceso WiFi (si los hay) y el puerto del mismo nivel.

Sin embargo, como se decía al comienzo del apartado 3.2 de introducción al sistema ZTP, antes de proceder hay que definir qué es lo que se quiere hacer y qué forma tendrá. Sobre todo, esto es importante para poder proceder con el paso de preconfiguración de manera impoluta. Para guiar la definición del sistema que se desea configurar, será necesario generar la siguiente información de manera previa a comenzar con el diseño del sistema ZTP:

- Diagrama lógico de red estándar de una oficina.
- Diagrama físico de red estándar de una oficina.
- Plantilla de direccionamiento IP.
- Plantilla con datos de configuración de routers WAN.
- Plantilla con datos de configuración de switches de CORE.
- Plantilla con datos de configuración de switches de ACCESO.

3.4 Documentación previa a la implementación del sistema ZTP

En este apartado se desarrollará la documentación previa a la implementación de un sistema ZTP desde la perspectiva de un entorno empresarial real. Por este motivo, se podrán apreciar particularidades de la compañía y habrá información que se omita por motivos confidenciales. Por otro parte, cabe destacar que no se realizarán explicaciones del significado de los comandos asociados a los sistemas operativos de los dispositivos a configurar, ya que se entiende que no es materia del presente trabajo.

3.4.1 Diagramas lógicos y físicos de red de una oficina estándar

No se profundizará en el desarrollo de estos diagramas ya que se puede partir de los diagramas aportados en el desarrollo del apartado Descripción del escenario.

3.4.2 Plantilla con datos de configuración de routers WAN

El primer reto que hay que afrontar a la hora de generar estos ficheros es la necesidad de crear un fichero lo más universal posible que contemple la mayoría de las opciones que tiene un router WAN y sus particularidades dependiendo del modelo que se elija.

Previo a generar la plantilla, se considera que hay que configurar un router Cisco con la siguiente configuración, por lo que deberá aparecer en la plantilla de manera obligatoria:

- General
    - Configurar hostname.
    - Configurar usuario “admin”.
    - Configurar usuario “forescout”.
    - Configurar “no aaa new-model”.
    - Configurar cryto pki autofirmado.
    - Configuración acceso remoto.
        - line con 0
        - line vty 0 4
        - exec-timeout 20 0
        - login local
        - transport input ssh
        - line vty 5 15
        - exec-timeout 20 0
        - login local
        - transport input ssh
        - ip ssh time-out 60
        - ip ssh authentication-retries 2
        - ip ssh version 2
    - SNMP
        - snmp-server community xxxxxxxx RO
        - snmp-server community yyyyyyyy RW
        - snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart

        - snmp-server host 192.168.252.190 yyyyyyyy
        - snmp-server host 172.21.2.71 xxxxxxxx
- Configuraciones de interconexión LAN.
    - Configuración de portchannels (con subinterfaces) e interfaces uplink contra CORE.
    - Configuración de interfaces de icnx_operador y/o icnx_firewall.
- Configuraciones de interconexión WAN.
    - Configuración de interfaces túnel.
    - Aplicación de rutas estáticas hacia el operador.
        - Operador 1.
        - Operador 2.
    - Ajustar MTU y MSS.
    - Probar conectividad.
- Configuración del encaminamiento.
    - Establece router-id (loopback).
    - Establecer auto-cost reference-bandwidth 1000.
    - Establecer passive-interface default.
    - Configurar la interfaz/portchannel icnx_edge/core como “no passive”.
    - Comprobar vecindad (sh ip ospf nei).
    - Publicar redes locales (incluido loopback).
    - Comprobar que se aprenden rutas (sho ip ro ospf).
    - Configurar interfaces túneles como “no passive”.
    - Comprobar vecindad (sh ip ospf nei).
    - Publicar redes locales (incluido loopback).
    - Comprobar que se aprenden rutas (sho ip ro ospf).

3.4.3 Plantilla con datos de configuración de switches de CORE

Previo a generar la plantilla, se considera que hay que configurar un switch Cisco con la siguiente configuración, por lo que deberá aparecer en la plantilla de manera obligatoria:

- Configuración general.
    - Configurar hostname.
    - Configurar usuario “admin”.
    - Configurar usuario “forescout”.
    - Configurar “no aaa new-model”.
    - Configurar cryto pki autofirmado.
    - Configurar IP DHCP snooping para todas las VLANs.

    - Configurar spannig-tree y BPDU-guard.
        - spanning-tree mode rapid-pvst
        - spanning-tree portfast bpduguard default
        - no spanning-tree etherchannel guard misconfig
        - spanning-tree extend system-id
        - spanning-tree VLAN 1-1000 priority 0
        - errdisable recovery cause bpduguard
        - errdisable recovery cause pagp-flap
    - Configuración acceso remoto.
        - line con 0
        - line vty 0 4
        - exec-timeout 20 0
        - login local
        - transport input ssh
        - line vty 5 15
        - exec-timeout 20 0
        - login local
        - transport input ssh
        - ip ssh time-out 60
        - ip ssh authentication-retries 2
        - ip ssh version 2
    - SNMP
        - snmp-server community xxxxxxxx RO
        - snmp-server community yyyyyyyy RW
        - snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
        - snmp-server host 192.168.252.190 yyyyyyyy
        - snmp-server host 172.21.2.71 xxxxxxxx
- Configuración de interfaces VLAN.
    - Descripción.
    - Dirección IP.
    - (opcional) IP-helpers.
- Configurar interfaz loopback.
- Configuración de VLANs L2 (icnx_fw, icnx_edge, icnx_operador).
- Configuración de portchannels e interfaces uplink contra ACCESO.
- Configuración VTP (server).
- Configuración IP name-servers.
- Configuración de portchannels e interfaces uplink contra EDGE.
- Encaminamiento.
    - Establece router-id (loopback).
    - Establecer auto-cost reference-bandwidth 1000.
    - Establecer passive-interface default.

- Configurar la interfaz/portchannel icnx_edge/core como “no passive”.
- Comprobar vecindad (sh ip ospf nei).
- Publicar redes locales (incluido loopback).
- Comprobar que se aprenden rutas (sho ip ro ospf).

3.4.4 Plantilla con datos de configuración de switches de ACCESO

Previo a generar la plantilla, se considera que hay que configurar switches de ACCESO Cisco con la siguiente configuración, por lo que deberá aparecer en la plantilla de manera obligatoria:

- Configuración general.
  - Configurar hostname.
  - Configurar usuario “admin”.
  - Configurar usuario “forescout”.
  - Configurar “no aaa new-model”.
  - Configurar cryto pki autofirmado.
  - Configurar IP DHCP snooping para todas las VLANs.
  - Configurar spannig-tree y BPDU-guard.
    - spanning-tree mode rapid-pvst.
    - spanning-tree portfast bpduguard default.
    - no spanning-tree etherchannel guard misconfig.
    - spanning-tree extend system-id.
    - spanning-tree VLAN 1-1000 priority 0.
    - errdisable recovery cause bpduguard.
    - errdisable recovery cause pagp-flap.
  - Configuración acceso remoto.
    - line con 0
    - line vty 0 4
    - exec-timeout 20 0
    - login local
    - transport input ssh
    - line vty 5 15
    - exec-timeout 20 0
    - login local
    - transport input ssh
    - ip ssh time-out 60
    - ip ssh authentication-retries 2
    - ip ssh version 2
  - SNMP.

        - snmp-server community xxxxxxxx RO
        - snmp-server community yyyyyyyy RW
        - snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
        - snmp-server host 192.168.252.190 yyyyyyyy
        - snmp-server host 172.21.2.71 xxxxxxxx
- Configuración de interfaz VLAN de gestión en pila de acceso.
- Configuración de default Gateway (interfaz VLAN de gestión).
- Configuración de Port-channels e interfaces de uplink.
- Configuración VTP (client).
- Configuración de puertos de acceso.
    - spanning-tree portfast

3.5 Implementación del sistema

En los siguientes apartados se desarrollará el Sistema ZTP en sí mismo. Se parte de la base de haber recogido toda la información necesaria y se procederá con las fases que utilizan dicha información para generar el entorno de red previsto.

Cabe recordar que el entorno donde se enfoca este diseño ha sido descrito en el apartado Descripción del escenario para el diseño.

3.5.1 El entorno Ansible

Ansible será la herramienta que despliegue la información en todo el equipamiento de red. Como se comentó en apartados anteriores, Ansible es un software que ejecuta “Playbooks” que se escriben en lenguaje YAML, por lo que será necesario traducir toda la información recopilada y obtener varios ficheros YAML.

Para el escenario en el que se ha enmarcado este proyecto se tendrán que manipular tres tipos diferentes de archivos (Ilustración 13: Sistema de ficheros de Ansible):

- Playbooks: serán los scripts que trasladen la información a los diferentes equipos.
- Hosts: es un único fichero donde se definen los diferentes equipos del escenario y el rol que desarrollan en el mismo.
- Vars: serán los ficheros donde se describan las características de cada uno de los equipos. Tiene una relación directa con el fichero

host, lo que permitirá definir variables relacionadas con los diferentes equipos y con los diferentes roles. Dicho de otra manera, los ficheros de tipo “vars” otorgan la capacidad de definir variables específicas del equipo y/o variables que comparten los diferentes dispositivos que forman parte del mismo rol.

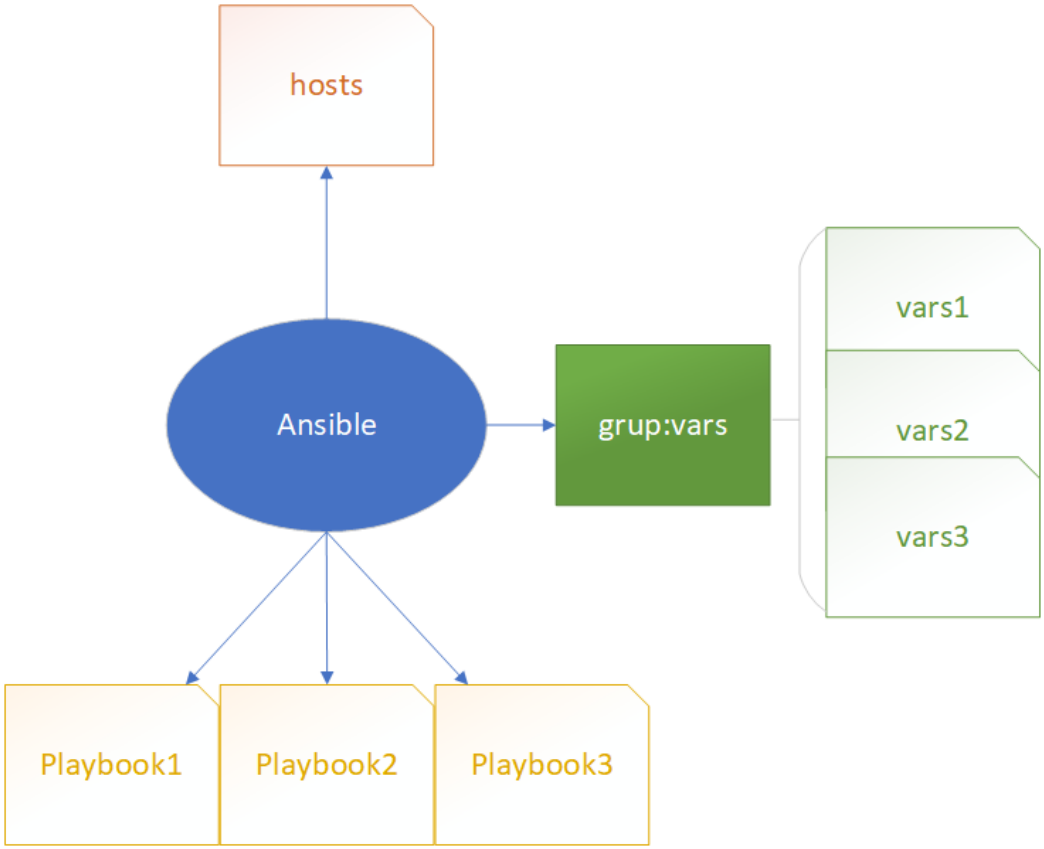

Ilustración 13: Sistema de ficheros de Ansible

3.5.1.1 Playbooks

Los Playbooks de Ansible, por estar basados en lenguaje YAML, tienen la siguiente estructura:

```
---
- name: Nombre del Playbook
  hosts: Equipos

  tasks:
    - name: Nombre de la tarea
      ios_command:
        commands:
         - show running-config interface GigabitEthernet1/0/14
```

La característica principal es que el fichero comienza con tres guiones en la primera línea. En las líneas segunda y tercera se anotan el nombre del Playbook y el entorno donde se quiere ejecutar dicho Playbook. Ese entorno debe estar definido dentro del fichero “hosts”. A continuación, se hace el desarrollo de las diferentes tareas, donde se tendrá que definir un nombre para cada una de ellas y, obviamente, lo que ha de configurarse en el equipo.

En dichas tareas, se pueden utilizar los miles de módulos que ofrece Ansible. A modo de ejemplo, se ha plasmado el módulo “ios_command” que es un módulo que permite ejecutar líneas de comando básicas en el sistema operativo IOS, que es el sistema que utilizan los equipos Cisco que se han propuesto para el escenario. En los siguientes apartados, se sugerirán algunos módulos para el desarrollo.

3.5.1.2 Fichero Hosts

El fichero Hosts es el fichero donde se definen los diferentes equipos que se configurarán y sus roles. Teniendo en cuenta lo definido en el apartado Descripción del escenario para el diseño3.1.4, el fichero hosts tendría la siguiente estructura:

```
[OFI_CORE]
OFIR1 ansible_host=10.117.252.1

[OFI_ACCESS]
OFIP10 ansible_host=10.117.252.10
OFIP11 ansible_host=10.117.252.11

[OFI:children]
ASU_CORE
ASU_ACCESS
```

Se puede observar que el fichero hosts se basa en etiquetas. Dichas etiquetas son los diferentes roles que se tienen en la arquitectura de red. Se pueden diferenciar tres diferentes roles:

- Los dos primeros, OFI_CORE y OFI_ACCESS hacen referencia a los switches que harán de CORE o ACCESO.

- Además, se ha configurado el rol OFI con el parámetro children, que signifia “hijos” en inglés. Mediante esta declaración se engloban bajo la etiqueta OFI todos los equipos que estén bajo las etiquetas OFI_CORE y OFI_ACCESS. Esto permitirá realizar configuraciones genéricas que sean iguales en todos los switches, tengan el rol que tengan.

Por otro lado, bajo las etiquetas se aprecian diferentes equipos que tienen un nombre arbitrario, una descripción que comienza por el parámetro “ansible_host=” y una dirección IPv4 arbitraria. Estos dos datos arbitrarios hacen referencia al nombre y a la dirección IP del equipo, que es la información necesaria para conectarse al mismo y hacer las configuraciones pertinentes. En este caso, se han utilizado IPs inventadas para facilitar la interpretación de la declaración.

3.5.1.3 Directorio grup:vars

El directorio vars es la carpeta donde se almacenarán cada uno de los ficheros de tipo vars. Cada fichero de tipo vars debe llamarse estrictamente igual a las etiquetas definidas en el fichero hosts o los nombres de los equipos. Por lo tanto, en base a la descripción del fichero hosts realizada en el apartado anterior, se deberían crear los siguientes ficheros:

- grup:vars
    - OFI_CORE.yml
    - OFI_ACCESS.yml
    - OFI.yml
    - OFIR1.yml
    - OFIP10.yml
    - OFIP11.yml

En cada uno de estos ficheros se escribirá la configuración específica de cada uno de los roles o de cada uno de los dispositivos. Esta estructura permite describir las configuraciones sin tener que repetir las configuraciones que compartan varios equipos.

Cabe destacar, que la redacción de estos ficheros puede resultar compleja y que implicará muchas horas de desarrollo, ya que se ha de desgranar la información obtenida en el apartado Documentación previa a la implementación del sistema ZTP.

3.5.2 Traducción de documentación a datos interpretables por Ansible

En este apartado se analizará y describirá la forma en la que se ha de redactar la configuración en Ansible y cómo hacerlo de forma eficiente. Este es un aspecto importante, ya que uno de los objetivos principales de la automatización de redes es eliminar la necesidad de construir todo de cero una y otra vez.

Por tanto, uno de los retos que plantea la automatización, es el de aplicar la configuración específica de cada equipo utilizando un Playbook genérico que se pueda aplicar sobre cualquier dispositivo, es decir, poder reutilizar Playbooks para diferentes entornos de red, sin tener que reescribirlo completamente. Para resolver este aspecto, se plantea el uso de plantillas escritas en lenguaje Jinja2 (Ilustración 14: Logo Jinja2).

Jinja2 es un motor de plantillas Python (recuérdese que Ansible está escrito en Python) que se utiliza para crear HTML, XML, YALM u otros formatos que se devuelven al usuario a través de una respuesta HTTP. Jinja resulta útil por ofrecer una sintaxis basada en etiquetas que permite redactar Playbooks limpios y reutilizables.

Ilustración 14: Logo Jinja2

Jinja será también el lenguaje que permita relacionar los Playbooks con las variables definidas en los diferentes ficheros de tipo vars. A continuación, se copia un fragmento de Playbook con etiquetas Jinja:

```
---
- name: Show PortChannels config using vars of each group of hosts
  hosts: OFI_CORE

  tasks:
    - name: Show run PO
      ios_command:
        commands:
          - show running-config interface {{ item.0 }}
      with_together:
          - "{{ Access_PortChannels }}" # PortChannels = item.0
```

Apréciense dos parámetros escritos entre llaves dobles. Estas son las etiquetas Jinja. Estas etiquetas están definidas en el fichero vars del rol

OFI_CORE o bien, en alguno de los ficheros vars de los dipositivos que tienen el roll OFI_CORE.

Es importante destacar que Ansible busca el valor de las etiquetas Jinja desde el fichero vars más específico hasta el más genérico. Esto quiere decir que Ansible buscará la variable en el siguiente orden:

1. Fichero vars de los dispositivos del rol.
2. Fichero vars del rol.
3. Fichero vars de un grupo de roles.

A continuación, se aporta un fragmento de un fichero de tipo vars, donde se puede apreciar la forma de declarar diferentes variables:

Name: OFIR1

Access_PortChannels: [Port-Channel2,Port-Channel3,Port-Channel11,Port-Channel12,Port-Channel13,Port-Channel14]

Portchannel_description: Port-Channel

VLANs: [vlan1,vlan2,vlan3]

Se puede ver que las variables se declaran sencillamente escribiendo un nombre y dos puntos. Después, se escribe el valor de la variable. Si se necesita declarar un array de valores (como en Access_PortChannels), después de los dos puntos se abren corchetes y se declaran varios valores separados por comas.

Posteriormente, para hacer referencia a estas variables desde los Playbooks, bastará con escribir el nombre de la variable entre corchetes dobles, tal y como se ha explicado en párrafos anteriores.

Se considera objeto de este trabajo plantear el diseño de un sistema que sea capaz de trasladar la información reseñada en los capítulos 3.4.2, 3.4.3 y 3.4.4 y que hacen referencia a las plantillas de datos de configuración, de ficheros de texto a ficheros en lenguaje YAML. Se plantea el siguiente proceso manual (Ilustración 15: Proceso de traducción de plantillas a ficheros YAML):

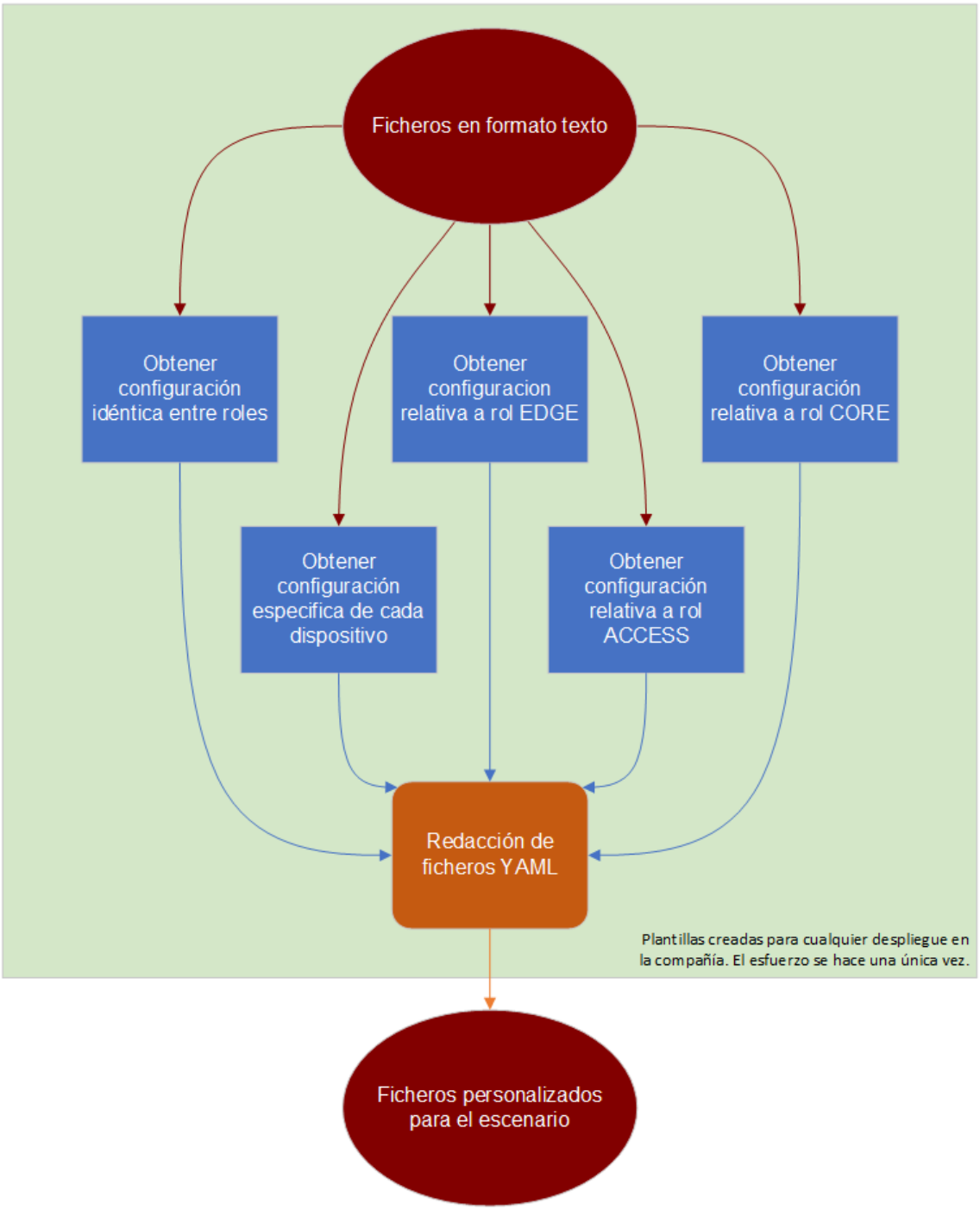

Ilustración 15: Proceso de traducción de plantillas a ficheros YAML

En primer lugar, se debe extraer toda la configuración común a los tres diferentes roles que se han planteado. Esto hace referencia a toda aquella información que es exactamente igual en los tres roles. Es importante destacar esto porque si bien los tres roles plantean un nombre de equipo, por ejemplo, este será diferente según el rol. Sin embargo, se puede apreciar que los tres roles necesitan la configuración de usuario admin. Esto es estrictamente igual en los 3, por lo que sería un parámetro seleccionable en esta fase.

En segundo lugar, se debe extraer la información común a los diferentes equipos de un mismo rol. Dicho de otra manera, se debe destacar la información que compartan todos los equipos que sean EDGE, los que sean CORE, y los que sean ACCESO. Esto quiere decir que se obtendrán 3 conjuntos de informaciones.

Hecho esto, la información que no haya sido seleccionada en los dos procesos anteriores, se puede considerar como información específica de cada dispositivo.

Una vez realizada la diferenciación de la información, se deberían obtener unas plantillas Jinja genéricas. Todos estos procesos se realizan una única vez, mientras que los que se explican a continuación, son ajustes que hay que hacer cada vez que haya que aplicar el sistema de ZTP sobre una nueva oficina.

La siguiente fase consiste en copiar las plantillas y ajustar todos los parámetros a la implantación de la oficina. En este proceso hay que especificar direccionamiento IP, nombres, configuraciones… relativas a la oficina que se desplegará. Esto quiere decir que se obtendrán los siguientes ficheros:

- Un fichero vars por cada dispositivo.
- Un fichero vars por cada rol.
- Un fichero vars genérico.

Dentro del marco definido en el apartado 3.1.4 Descripción del escenario para el diseño, obtendríamos:

- 3 ficheros vars de dispositivos (1 CORE y 2 ACCESOs),
- 2 ficheros de roles (1 de CORE y otro de ACCESO)
- y un fichero de vars genérico.

En total 6 ficheros del tipo vars.

Los últimos pasos de este proceso serían la carga de los ficheros en el servidor de automatización y la adecuación del fichero hosts.

3.5.3 Visión general del sistema de despliegue de configuraciones

En este apartado, se dibujará a groso modo la secuencia de procesos que seguirá el Sistema ZTP para hacer las configuraciones pertinentes. A continuación, se dibuja la Ilustración 16: Visión general de los procesos del Sistema ZTP.

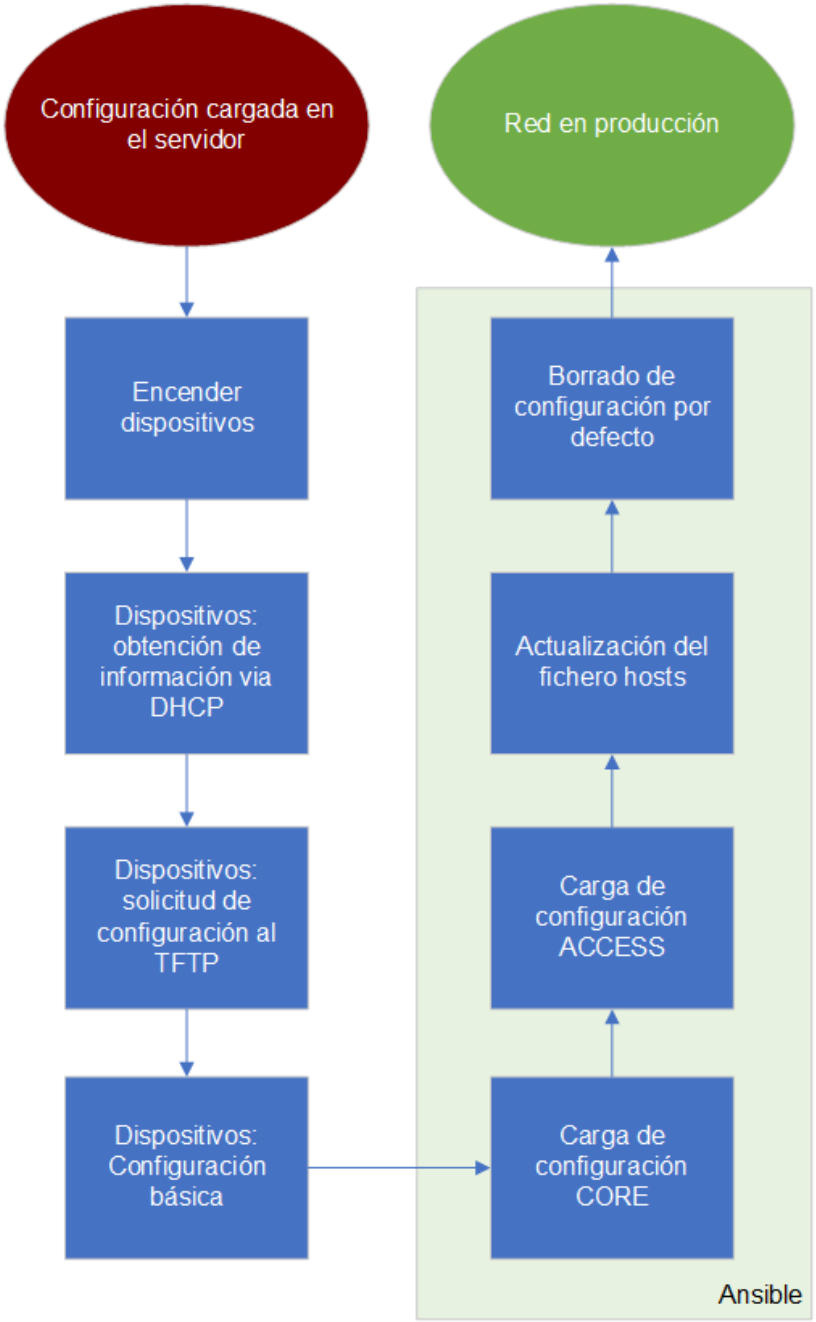

Ilustración 16: Visión general de los procesos del Sistema ZTP

El proceso de Sistema ZTP arranca en cuanto toda la información y las plantillas han sido cargadas en el servidor de automatización.

Evidentemente, para continuar, los equipos a configurar tienen que estar conectados y encendidos.

En cuanto los equipos arranquen recibirán cierta información por DHCP lo que les permitirá poder conectar con el servidor de automatización y descargar una configuración mínima para funcionar.

En este punto, Ansible ya podrá contactar con los dispositivos de red que han sido recién instalados y podrá comenzar con la configuración de los mismos. En primera instancia configurará el rol de CORE, y por último los dispositivos asociados al rol de ACCESO.

Una vez toda la configuración ha sido desplegada hay que actualizar el fichero hosts. Los equipos han funcionado hasta ahora con la configuración por defecto que les otorgó el servidor DHCP, pero esa información ya no es necesaria y se debe eliminar. Es importante actualizar el fichero hosts con la nueva IP de cada dispositivo con la que se podrá contactar. Por último, Ansible se conectará con los dispositivos en la nueva IP, borrará la información antigua y deshabilitará las interfaces por defecto.

3.5.4 Proceso de conexión a los equipos de red

Este proceso es el primero en el que el servidor de automatización y los dispositivos interactuarán.

Antes de comenzar este proceso se han de cumplir los siguientes requisitos:

- Los dispositivos de red están instalados.
- Los dispositivos de red están encendidos.
- Los dispositivos de red están cableados tal y como se define en el diagrama físico.
- El router de operador está instalado y operativo.
- El router de operador tiene un servidor DHCP configurado.

Todos estos requisitos resultarán fáciles de cubrir, sin embargo, la demanda al proveedor de servicios de un servidor DHCP exigirá una interlocución clara y precisa.

En concreto, al operador de red se le tiene que pedir un servidor DHCP que tenga las siguientes características:

- Tener un scope[14] de red temporal, el contemplado en el fichero hosts.
- Publicación de la red en la MPLS.
- Tener una reserva de IPs asociada a la direcciones MAC[15] de cada dispositivo.
- Configurar como puerta de enlace el router del operador.
- Configurar un servidor TFTP[16] que apunta al servidor de automatización (Ansible).
- Configurar un IP secundaria que será la IP definitiva que conectará el equipo con la red MPLS de la compañía.

El motivo de esta solicitud tan específica es que la configuración automática que permitirá la conexión con los dispositivos debe ser la misma que se ha determinado en el fichero hosts de Ansible y debe poder relacionarse con exactitud. En el siguiente diagrama (Ilustración 17: Proceso de conexión inicial de los equipos) se explica el proceso que proporciona conectividad a los equipos inmediatamente después a que hayan sido encendidos.

[14] Hace referencia a un grupo limitado de direccionamiento IP.
[15] Dirección física de un dispositivo. Esta dirección es única, estática (no cambia) e identifica el dispositivo inequívocamente.
[16] Protocolo de transporte de ficheros.

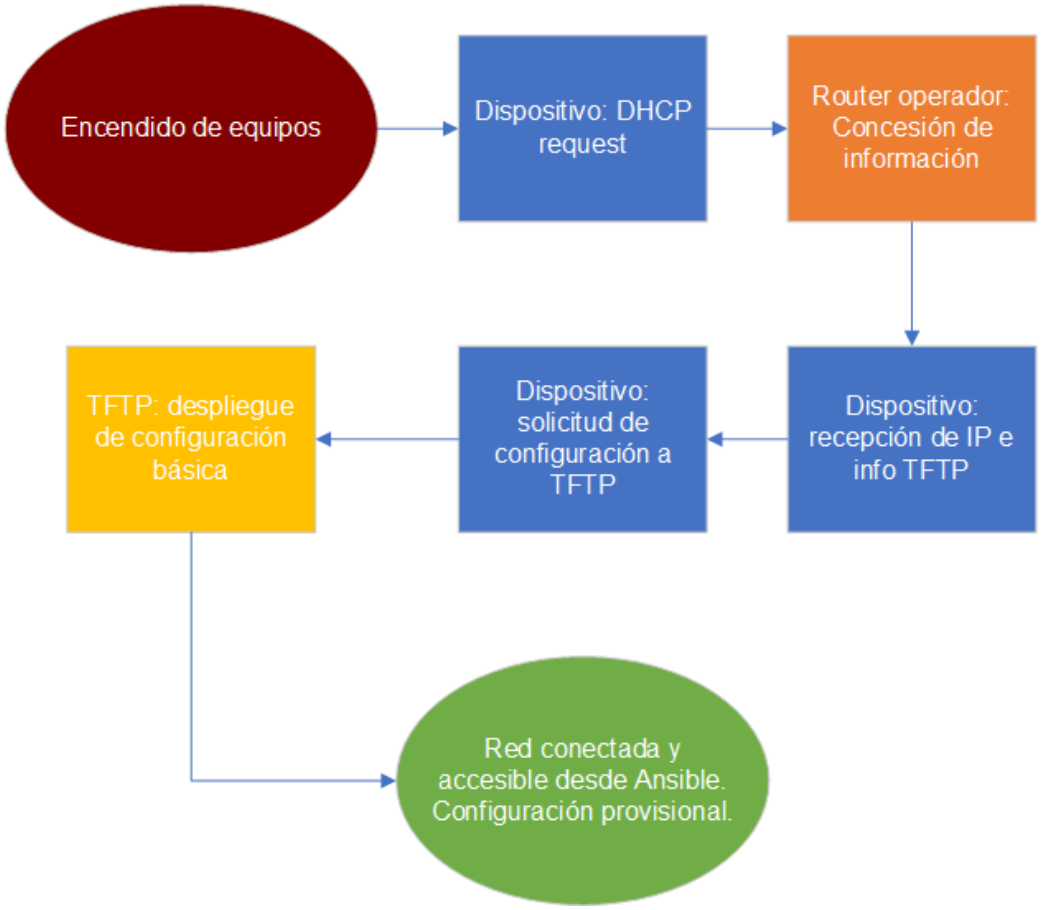

Ilustración 17: Proceso de conexión inicial de los equipos

Hay que prestar mucha atención al orden de inicio de los equipos. Para el escenario propuesto, habría que arrancar los equipos en el siguiente orden y esperando a hayan arrancado por completo para proceder con el siguiente:

- Router de operador
- Router de CORE
- Switches de ACCESO

Además, si hay switches “stackados”, se deberán arrancar en el orden de stack. Esto quiere decir que primero hay que arrancar el primer switch del stack, después el segundo y así, sucesivamente. Es importante hacerlo así, sobre todo, con el primer switch, ya que la información física del stack (S/N, MAC…) será la del switch que arranque primero.

En cuanto los dispositivos hayan arrancado (pueden tardar varios minutos) realizarán un DHCP request, lo que les proporcionará a cada uno de ellos una dirección IP concreta (la asociada a la dirección MAC), la IP de la puerta de enlace, y la IP del servidor TFTP.

El servicio de TFTP es un servicio necesario y puede estar alojado en un servidor diferente al servidor de automatización, aunque tiene sentido que se aloje en el mismo servidor. El servidor TFTP será el responsable de cargar la mínima configuración para que el dispositivo pueda ser gestionado por Ansible.

En primera instancia, un dispositivo virgen no es gestionable por Ansible ya que Ansible utiliza el protocolo SSH para conectarse al dispositivo. El servicio SSH viene deshabilitado por defecto en los equipos Cisco por lo que será una de las configuraciones que se delegará en el servidor TFTP.

Un switch de marca Cisco nuevo viene con cierta configuración por defecto. Todas las bocas del switch vienen configuradas en la VLAN 1, también conocida como la VLAN por defecto. Esto quiere decir que todas las interfaces están configuradas en la misma red. Además, el switch tendrá configurada una interfaz VLAN 1, que es una interfaz virtual por defecto. Será esta interfaz la encargada de realizar un DHCP request que le provea de la configuración necesaria para comenzar.

Cabe mencionar, que esta VLAN e interfaz VLAN deberán ser deshabilitadas o borradas una vez acabe todo el proceso de ZTP. Solo se utilizarán para poder realizar las primeras conexiones y volcados de información.

Por otro lado, para conectarse mediante SSH habrá que crear un usuario que permita loguearse al usuario administrador. Esta configuración, será responsabilidad del servidor TFTP.

En resumen, el servidor TFTP tiene que configurar las siguientes opciones:

- SSH (debe autogenerar las claves)
- Un usuario de acceso

No obstante, se debe destacar que este proceso es un proceso muy crítico. La carga de las configuraciones ha de hacerse en un orden concreto, ya que hacerlo en un orden equivocado puede dejar desconectado algún equipo y sería necesaria la asistencia de un técnico que reseteara la configuración o reiniciara el equipo. El orden de aplicación de configuraciones desde el TFTP sería el siguiente:

1. Equipos finales (no planteado en el escenario).
2. Equipos de ACCESOs (switches de ACCESO).
3. Equipos de CORE (switch de CORE).
4. Equipos de EDGE (router de EDGE, no planteado en el escenario).

Una vez realizada la carga de la configuración mínima desde el servidor TFTP, los dispositivos estarían conectados a la red corporativa con el direccionamiento definitivo.

En este punto del proceso, se podría plantear la cuestión de por qué no aplicar toda la configuración mediante el servidor TFTP. Las razones que defienden no proceder de esta manera, son las siguientes:

1. El servidor TFTP cambia el direccionamiento asociado, por lo que se pierde la asociación de IP-dispositivo.
2. El TFTP carga la configuración tal y como haya sido escrita. Si se atasca o hay errores no hay forma de saber qué ha pasado.
3. Si se necesita aplicar alguna modificación en la configuración, Ansible solo aplicará los cambios, mientras que el TFTP lo reescribirá todo de nuevo, con el peligro que eso conlleva (posible desconexión).
4. Ansible facilita la programación, porque es un lenguaje declarativo. En cambio, el TFTP exige una configuración ultraprecisa, ordenada y estática.
5. La comunicación mediante Ansible proporciona feedback, mientras que el proceso mediante TFTP es totalmente opaco.

3.5.5 Obtención y corrección de la topología de red

Aprovechando los recursos que ofrece un software tan completo como Ansible, el siguiente eslabón del proceso sería comprobar que la topología de red coincide con la diseñada.

Este proceso es necesario para comprobar que el técnico que ha instalado y conectado los equipos lo haya hecho siguiendo los diagramas que se le han proporcionado (7. Erikruiter2 (2019)).

Cisco ofrece un protocolo llamado Cisco Discovery Protocol (CDP). Este protocolo ofrece información sobre los equipos que se han conectado en las interfaces de un equipo Cisco. A continuación, se copia un ejemplo

real del resultado de la ejecución del comando “sh cdp neighbors” en un equipo Cisco:

```
#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
P10              Gig 1/1/1         175           S I   WS-C2960X Gig 1/0/49
P10              Gig 2/1/1         179           S I   WS-C2960X Gig 2/0/49
```

Se puede apreciar que la información ofrecida por el comando en muy completa, ya que ofrece los siguientes campos, entre otros:

- ID del dispositivo (si es Cisco, aparece el nombre del dispositivo).
- Interfaz del equipo donde se ha localizado el equipo remoto.
- Tipo de dispositivo detectado (Capability). Suele estar representado con una letra que se puede descifrar usando la leyenda.
- Plataforma, que en otras palabras, es el modelo del dispositivo detectado.
- ID del puerto remoto, que ofrece la información del puerto con el que se ha conectado el dispositvo local.

Esta información es tan completa porque toda la infraestructura se basa en equipos de la marca Cisco. Si no fuera así, la información obtenida sería más limitada.

En base a la información que ofrece este comando se deberá desarrollar el primer Playbook, que será el encargado de plantear la arquitectura física de la topología de red. Dicho Playbook, deberá seguir el siguiente esquema funcional (Ilustración 18: Proceso de comprobación de la topología):

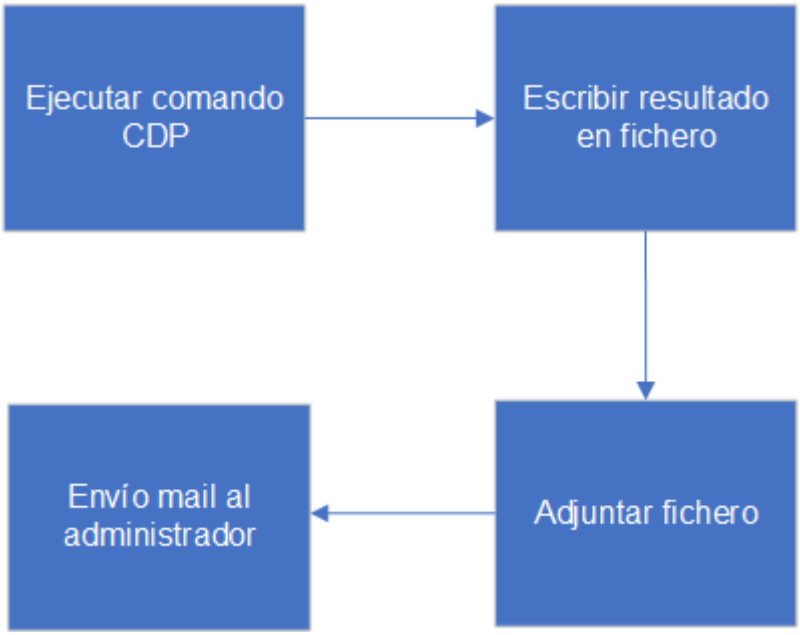

Ilustración 18: Proceso de comprobación de la topología

Este Playbook se ejecutará a voluntad del administrador de la red. Cabe destacar que este Playbook deberá contemplar el entorno más genérico posible, que será el entorno que englobe todos los dispositivos previstos en la arquitectura. En el ejemplo que se ha propuesto para las explicaciones de este trabajo, el grupo OFI del fichero hosts sería el candidato adecuado, ya que cubre todos los dispositivos de la red que se está desplegando.

Una primera tarea será la encargada de ejecutar el comando en cada uno de los switches y otra diferente la que cogerá el resultado del comando y lo copiará a un fichero de texto. Ese fichero de texto se irá poblando con la información de todos los switches.

Al finalizar la recopilación de la información, dicho texto se deberá adjuntar en un correo y se enviaráal administrador de la red. El administrador, con dicho fichero podrá detectar cualquier tipo de anomalía en la red. Los parámetros que se deberán comprobar son los siguientes:

- Aparecen todos los elementos de la red.
- El switch de CORE reporta conectividad contra todos los demás dispositivos de manera duplicada. Esto permite comprobar que los caminos están redundados.
- El resto de elementos reporta duplicidad de caminos respecto al switch de CORE.

- Las interfaces documentadas en el fichero coinciden con exactitud con las planificadas.

Si se detectara algún tipo de error de conexión habría que informar a un técnico para que se desplazase al lugar y así corregir el error. Si todo es correcto, se puede validar la arquitectura física de la red y se puede proceder con el resto de configuraciones.

Nota: Para este proceso, el módulo de Ansible ios_command es de gran utilidad.

3.5.6 Proceso de configuración de switches de CORE

Los switches de CORE son el nudo central de cualquier red local. Su configuración es muy importante y los switches de ACCESO se basan en el switch de CORE para ciertas configuraciones.

Para la configuración del switch de CORE se plantearán diferentes Playbooks, cada uno de los cuales configurará una parte de la configuración general. Se podría resumir todo en un único Playbook, pero hacerlo de manera diferenciada facilita la profundización y la comprensión del procedimiento.

3.5.6.1 Configuración específica del rol de CORE

En primera instancia se desplegará la configuración especifica del switch de CORE. Todos estos parámetros deberán estar definidos en los ficheros de tipo vars relacionados con el CORE. El Playbook debería seguir el flujo planteado en la Ilustración 19: Proceso de configuración especifica del rol de CORE:

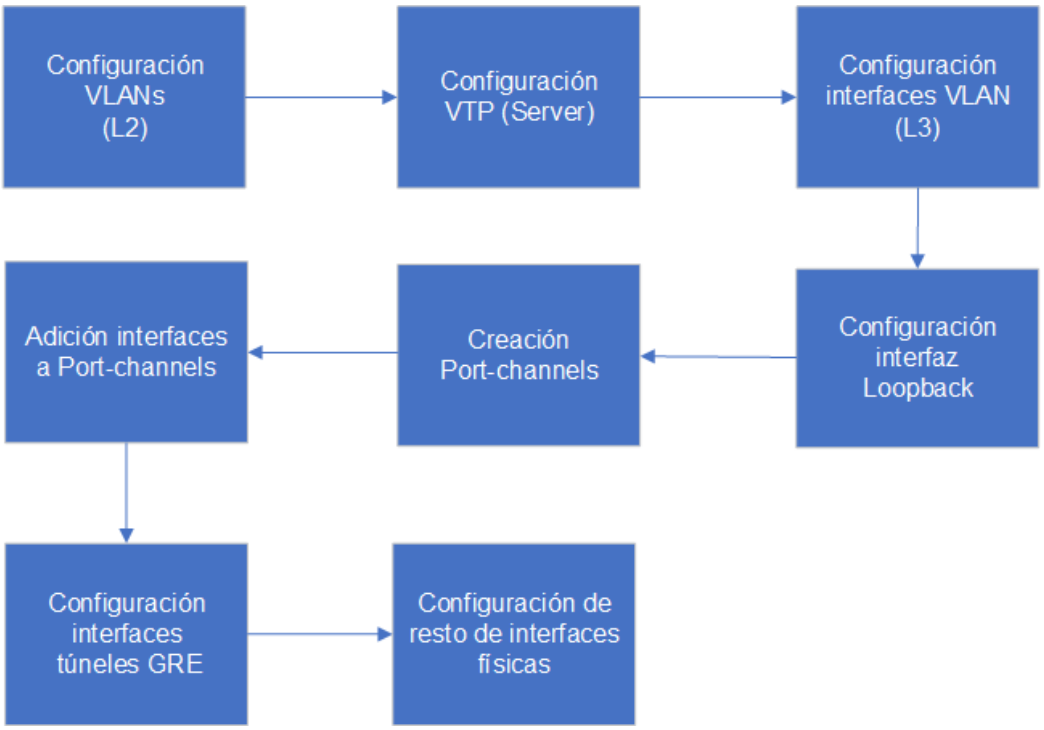

Ilustración 19: Proceso de configuración especifica del rol de CORE

El primer paso será la configuración de las diferentes VLANs de la red de la oficina. Ansible tiene un módulo específico de configuración de VLANs de nivel 2 que facilita enormemente esta tarea (8. Kuchenski, D (2017)).

Posteriormente se realizará la configuración VTP (VLAN Trunking Protocol). Dicha configuración se hará en modo servidor. De esta manera, cuando se configuren más adelante los switches de ACCESO en modo cliente, estos aprenderán las diferentes VLANs.

Tras la creación de las VLANs será necesario crear las interfaces VLAN. Las interfaces VLAN son interfaces virtuales de nivel 3, es decir, interfaces que tienen configurada una IP. La configuración de las interfaces VLAN creará las diferentes puertas de enlace por defecto (default gateways) de la red. Esta configuración es necesaria si se quiere tener la posibilidad de alcanzar equipos que están configurados en diferentes VLANs.

También se configurará la interfaz de loopback. Es otro tipo de interfaz virtual. La característica que tiene es que es una interfaz que siempre está en estado UP (levantada). Es una configuración muy recomendable para realizar la resolución de problemas (troubleshooting).

El siguiente proceso será la creación de las agregaciones de puertos (Port-channels). Las agregaciones de puertos se suelen realizar por dos motivos prácticos:

- Aumento del ancho de banda de las interfaces.
- Redundancia de caminos.

En una primera tarea se creará la agregación de puertos y en una segunda tarea se añadirán las diferentes interfaces físicas a esa agregación. Dichas interfaces físicas serán las que conecten el switch de CORE con la diferentes pilas de ACCESO. En los Port-channels se deben configurar todas las VLANs que tienen que ser accesibles desde los switches de ACCESO, incluida la VLAN 1, para que no queden desconectados.

Por último, se procederá con la configuración de las interfaces del tipo túnel GRE (Generic Routing Encapsulation). Nuevamente, serán unas interfaces virtuales que conectarán el switch de CORE con los nodos centrales de la red WAN: los routers de agregación. Como se comentó en apartados anteriores, están ubicados en las sedes centrales. Estas interfaces virtuales abstraerán a la red local de la oficina de la red WAN y la conectarán directamente a la oficina central. No es una configuración estrictamente necesaria, pero simplifica mucho la configuración de la red WAN de la compañía.

Por último, se deben configurar el resto de puertos físicos del switch de CORE. En el escenario planteado solo se configuran las interfaces que se conectan con el ACCESO y con el router de operador, pero puede haber entornos (de hecho, es bastante habitual), donde en el CORE se conectan servidores, cabinas de almacenamiento… No obstante, esta posible necesidad no es de relevancia en el presente trabajo. En este punto, solo quedaría configurar el puerto que interconecta en switch de CORE con el router de operador. Por defecto, ese router está conectado en una interfaz configurada en la VLAN 1. Habrá que configurar en la interfaz en modo trunk y permitiendo la VLAN 1 y también la VLAN de interconexión que se haya previsto.

Al acabar este proceso, el switch de CORE será accesible desde la IP que le otorgó el operador de red y también desde la IP de la interfaz VLAN de interconexión.

Nota: Para este proceso, se recomienda la utilización de los siguientes módulos de Ansible:

- ios_vlan
- ios_l2_interface
- ios_l3_interface
- ios_linkagg

3.5.6.2 Configuración genérica del rol de CORE

En un segundo Playbook sería interesante configurar toda la información genérica del switch de CORE. El siguiente diagrama de bloques (Ilustración 20: Proceso de configuración genérica del rol de CORE) dibuja cómo debería ser la distribución de esta información:

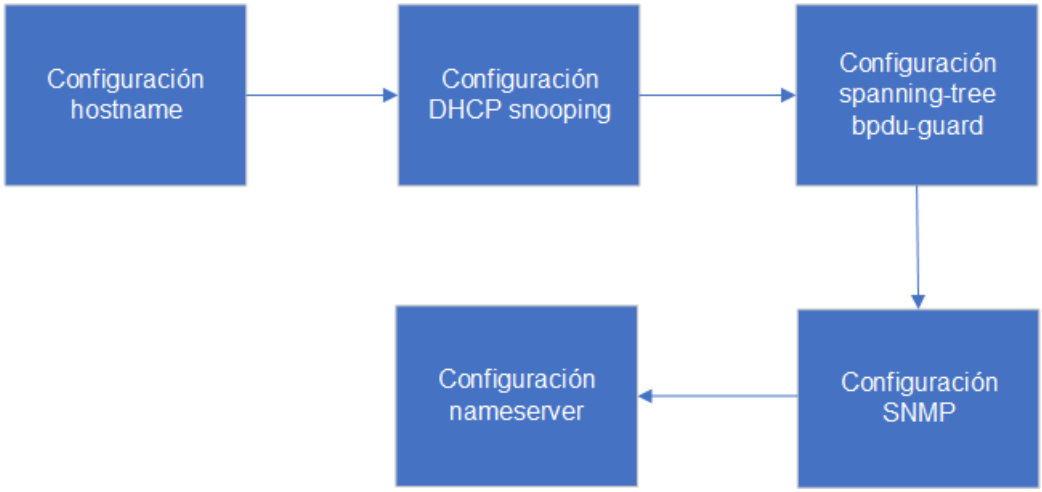

Ilustración 20: Proceso de configuración genérica del rol de CORE

Como primer paso es importante configurar el nombre (hostname) del switch CORE. Esta sencilla configuración nos ayudará a identificar el equipo, además de ubicarlo en la arquitectura, siempre y cuando se use un nombre conciso pero descriptivo. Por ejemplo, OFIR1, nos dirá que es router (R) principal (1) de la oficina (OFI).

Será importante también configurar el parámetro “DHCP snooping” para todas las VLANs por defecto salvo para la VLAN 1. Esto nos permitirá bloquear servidores DHCP clandestinos, y al hacer la salvedad para la VLAN 1, no tendremos problemas con el servidor DHCP configurado en el router de operador.

También se deben habilitar las guardas de “spanning-tree” y “bpdu-guard” para evitar bucles en la red. Cabe destacar, que es necesario hacer la salvedad en las interfaces port-channel (apartado anterior), para que la

guarda de BPDUs no nos bloquee los puertos donde están conectados los switches de ACCESO.

Una configuración que no es necesaria para el funcionamiento, pero que es muy recomendable hacer, es la habilitación de comunidades SNMP (Simple Network Management Protocol). Esto permitirá a los servidores de monitorización obtener datos del equipo y poder detectar de manera anticipada posibles fallos físicos.

Por último, una práctica habitual es configurar servidores DNS. Esto le permitirá al switch de CORE poder traducir nombres en IPs y viceversa.

Nota: Para este proceso, se recomienda la utilización de los siguientes módulos de Ansible:

- ios_config
- ios_command
- ios_user

3.5.6.3 Proceso de configuración del encaminamiento

Probablemente éste sea el proceso más sensible de la configuración. Un error en este proceso de configuración puede dejar totalmente desconectada la oficina, obligando a la compañía a desplazar a un técnico para hacer correcciones en local. La clave del éxito es proceder de manera ordenada, tal y como se plantea en la Ilustración 21: Proceso de configuración del encaminamiento:

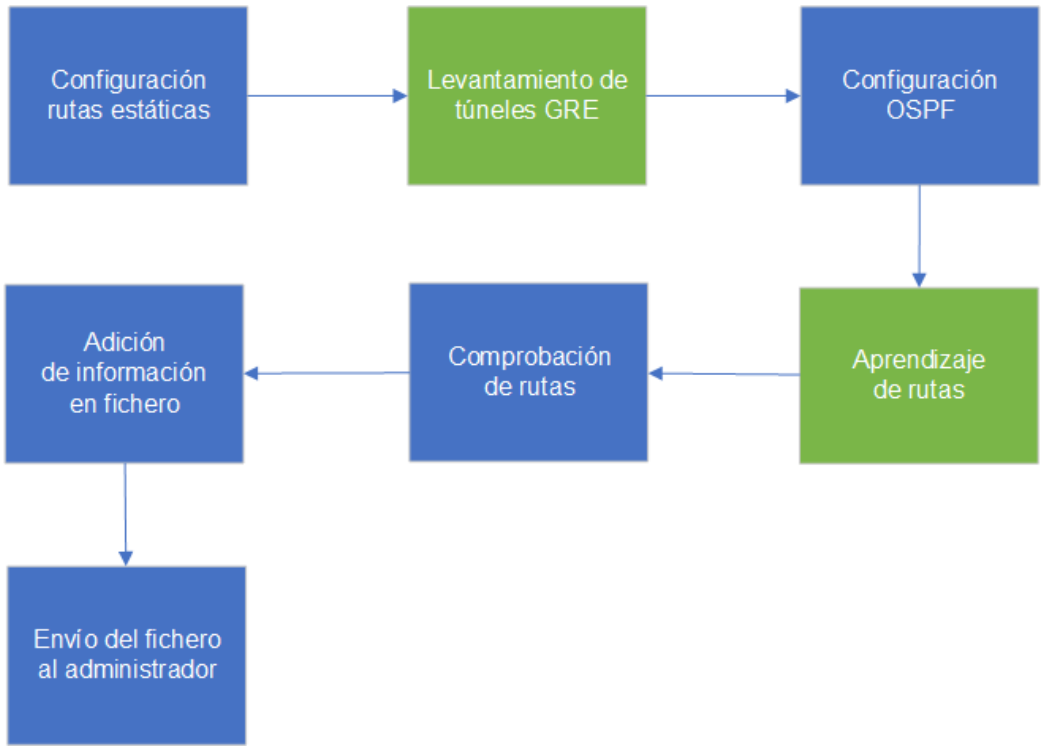

Ilustración 21: Proceso de configuración del encaminamiento

Los primeros parámetros a configurar serán las rutas estáticas que señalarán dónde están el resto de extremos de la MPLS. Esta configuración es necesaria para poder levantar los túneles GRE contra la sede central. Como se puede intuir, el extremo remoto de los túneles debe estar publicado en la MPLS.

La acción anterior levantará los túneles (si previamente también han sido configurados en la sede central) y deberemos realizar la configuración de OSPF (Open Shortest Path First). Es un protocolo de encaminamiento dinámico que permite la publicación de redes privadas. Dicho de otra manera, es el protocolo que permite hablar con otros equipos de la red para intercambiar información sobre la ubicación de las redes. Al saber la ubicación de las redes, el tráfico sabrá a dónde debe ir de manera dinámica. La configuración de OSPF debe contemplar las siguientes características y deben ser configuradas en el orden en el que se listan:

- Configuración de router-id.
- Ajuste del auto-coste.
- Las interfaces deben estar en modo pasivo por defecto.
- Las interfaces de tipo túnel deben estar en modo no pasivo.
- Se publican las redes locales de la oficina (incluida la red de la interfaz de loopback).

La configuración anterior debe provocar un intercambio de rutas y todos los rincones de WAN deberían comenzar a ser alcanzables.

Es interesante configurar una tarea de comprobación que valide que se están aprendiendo las rutas de la red. La idea sería ejecutar un comando que muestre las rutas aprendidas, copiarlo en un fichero y enviárselo al administrador de redes.

Si la publicación de rutas es correcta, se puede determinar que la configuración del CORE de la oficina es completa. En ese caso, el administrador de redes podrá desencadenar la siguiente ejecución de Playbooks.

Nota: Para este proceso, se recomienda la utilización de los siguientes módulos de Ansible:

- ios_command
- ios_static_route
- ios_bgp

3.5.7 Proceso de configuración de switches de ACCESO

Una vez configurado el switch de CORE, se puede comenzar con la configuración del resto de switches, en concreto, los switches de ACCESO. Igual que se ha procedido con el CORE, la configuración de los switches de ACCESO se puede dividir en tres subprocesos independientes.

3.5.7.1 Configuración específica de switches de ACCESO

Al igual que con los switches de CORE, hay que comenzar por la configuración específica del rol de ACCESO. Cabe recordar, que toda la información debe estar anotada en los ficheros de tipo vars. El siguiente diagrama (Ilustración 22: Proceso de configuración específica del rol de ACCESO) explica la función de este primer Playbook:

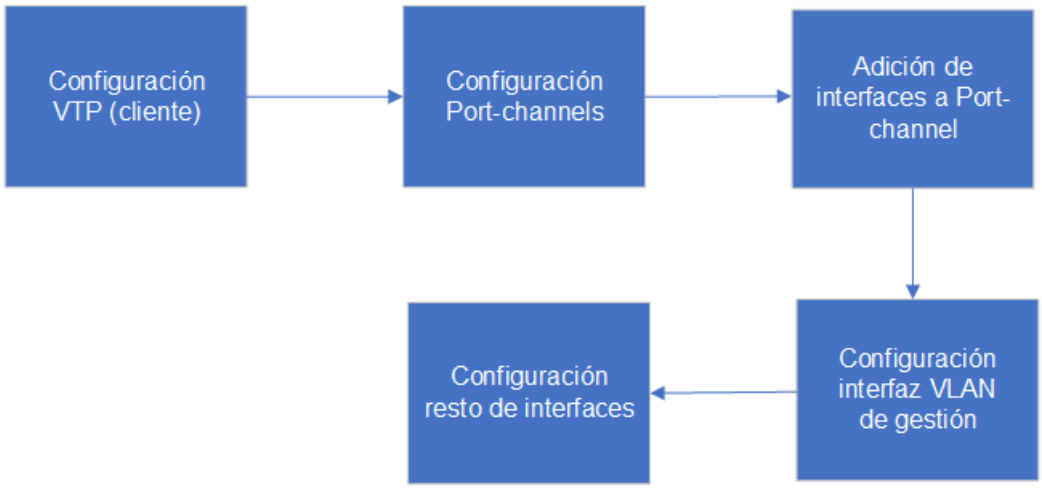

Ilustración 22: Proceso de configuración específica del rol de ACCESO

La primera tarea del Playbook debe ser la configuración de VTP (VLAN Trunking Protocol) en modo cliente. Hacer esto hará que los switches de ACCESO "aprendan" las VLANs configuradas en el CORE de manera automática. Sin embargo, las VLANs no estarán disponibles hasta realizar el siguiente paso.

En segunda instancia es necesaria la configuración de las agregaciones de puertos (Port-channels) y sus interfaces físicas asociadas. Como se destacó anteriormente, esto aumentará el caudal para el tráfico entre los switches de ACCESO y el CORE y también ofrecerá caminos redundados para protegerse ante fallos en los enlaces físicos. Para que el aprendizaje de VLANs funcione, la configuración de los port-channels tiene que permitir todas la VLANs necesarias en modo "trunk". No se debe olvidar la VLAN 1.

Después, se ha de configurar una interfaz VLAN de gestión. Será la interfaz que sustituirá a la VLAN 1 que viene configurada por defecto. Esta interfaz de nivel 3 es la que permitirá que el sistema de automatización o el administrador del sistema se conecten directamente al equipo sin tener que hacer saltos intermedios. Por el momento, el servidor de Ansible seguirá conectándose a la interfaz VLAN 1.

Por último, habrá que configurar las interfaces físicas de acceso. Aquí es donde se conectarán los equipamientos de usuario (ordenadores, impresoras…) y deberán dejarse listas para su uso.

Nota: Para este proceso, se recomienda la utilización de los siguientes módulos de Ansible:

- ios_l2_interface

- ios_l3_interface
- ios_linkagg

3.5.7.2 Configuración genérica de switches de ACCESO

Por último, se deben realizar el resto de configuraciones del switch de ACCESO. Estas configuraciones son muy similares a las realizadas en el switch de CORE. El siguiente diagrama (Ilustración 23: Proceso de configuración genérica del rol de ACCESO) plantea el flujo de trabajo a seguir:

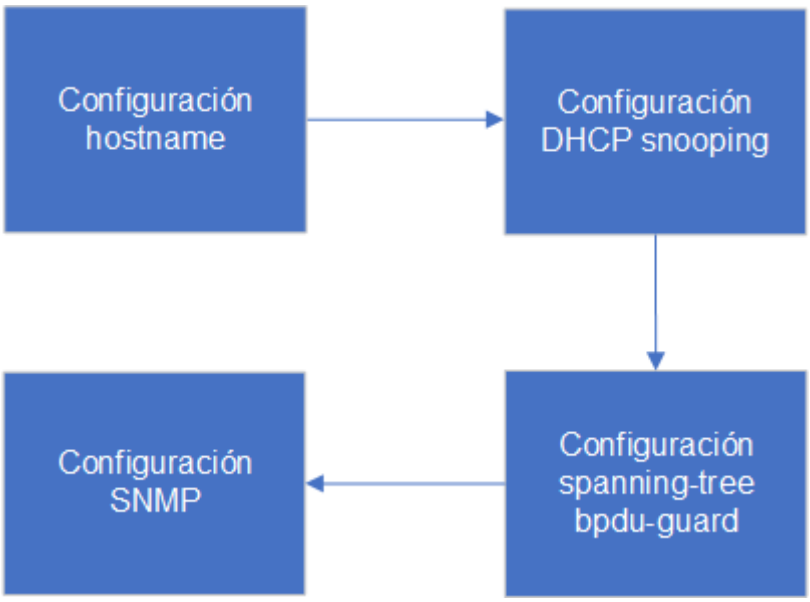

Ilustración 23: Proceso de configuración genérica del rol de ACCESO

Como primer paso, es importante configurar el nombre (hostname) de los switches de ACCESO. Esta sencilla configuración nos ayudará a identificar el equipo, además de ubicarlo en la arquitectura, siempre y cuando se use un nombre conciso pero descriptivo. Por ejemplo, OFIP10, nos dirá que es una pila (P) 10 de acceso de la oficina (OFI).

Será importante también configurar el parámetro “DHCP snooping” para todas las VLANs por defecto salvo para la VLAN 1. Esto nos permitirá bloquear servidores DHCP clandestinos, y al hacer la salvedad para la VLAN 1, no tendremos problemas con el servidor DHCP configurado en el router de operador.

También se deben habilitar las guardas de “spanning-tree” y “bpdu-guard” para evitar bucles en la red. Cabe destacar, que es necesario hacer la salvedad en las interfaces port-channel (apartado anterior), para que la guarda de BPDUs no nos bloquee los puertos donde están conectados los switches de CORE.

Una configuración, que no es necesaria para el funcionamiento pero que es muy recomendable hacer, es la habilitación de comunidades SNMP (Simple Network Management Protocol). Esto permitirá a los servidores de monitorización obtener datos del equipo y poder detectar de manera anticipada posibles fallos físicos.

Nota: Para este proceso, se recomienda la utilización de los siguientes módulos de Ansible:

- ios_config
- ios_command
- ios_user

3.5.8 Proceso de actualización de la conexión

Para realizar la primera conexión en los equipos es necesaria la asistencia del operador de red, que mediante un servidor DHCP otorga IPs a los diferentes dispositivos. Esta configuración es temporal y en este punto del proceso ya puede ser actualizada. Ahora, se comenzará a utilizar el direccionamiento de red definitivo. La Ilustración 24: Proceso de actualización de conexión explica de forma gráfica el proceso:

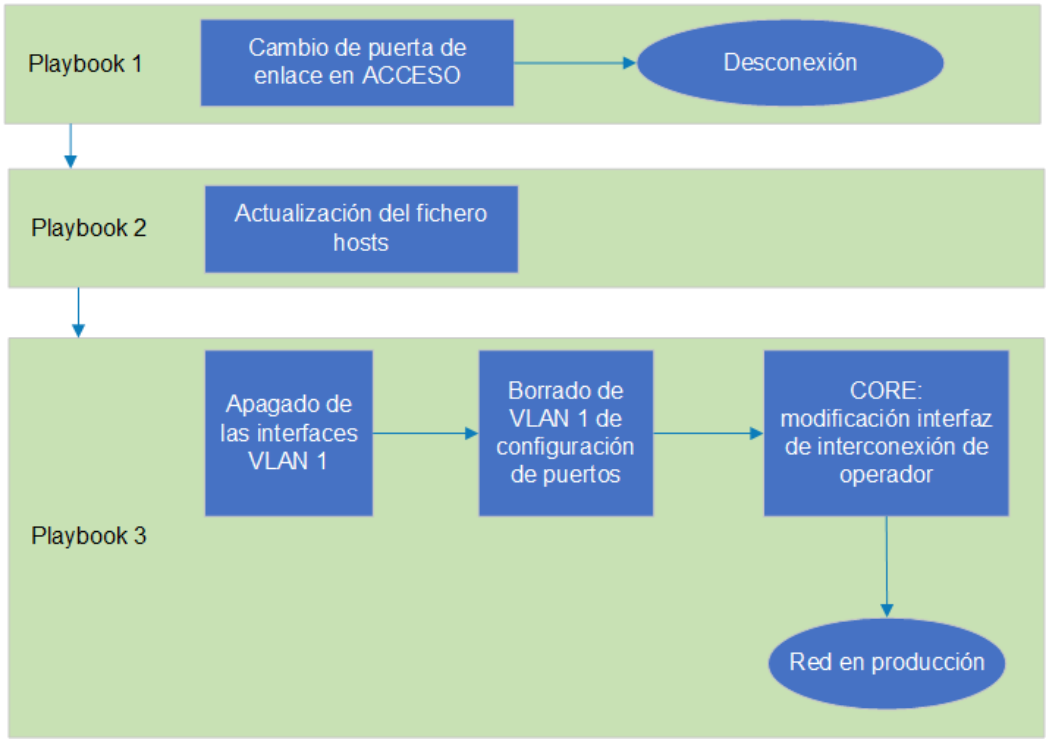

Ilustración 24: Proceso de actualización de conexión

En primer lugar, se deberá configurar la puerta de enlace por defecto (default Gateway) de los switches de ACCESO. La puerta de enlace por defecto deberá ser la interfaz equivalente en el CORE. Ésta es una condición necesaria para que mediante routing se puedan gestionar los switches de ACCESO a través de la nueva interfaz VLAN.

Al ejecutar esta tarea, el servidor de automatización reportará una desconexión de los equipos. Esto quiere decir que los switches de ACCESO han dejado de ser alcanzables desde la IP que se configuró en el servidor de automatización. Por esta razón, es momento de actualizar el direccionamiento de los equipos en el fichero de hosts. Se deberán anotar las IPs previstas para las interfaces VLAN de gestión.

El proceso de actualización del fichero hosts puede ser manual o automático (mediante Ansible), pero se realizará aparte de las siguientes tareas, ya que cuando se ejecuta un Playbook de Ansible y se modifica el fichero hosts, hasta que no se ejecute un nuevo Playbook, no se usará la versión actualizada del mismo.

Una vez actualizadas las IPs, se apagarán (shutdown) las interfaces VLAN 1 de los switches de ACCESO y de los switches de CORE. No existe riesgo de desconexión porque éstos ya se conectan por la nueva interfaz.

Después, se puede eliminar la VLAN 1 de la configuración de los Port-channels porque ya no es necesaria. Ahora se usa la VLAN dedicada a la gestión.

Por último, en el switch de CORE se debe realizar la actualización del puerto que interconecta el switch con el router de operador. Deberá pasar de ser una interfaz en modo trunk a una en modo ACCESO en la VLAN prevista para la interconexión del operador.

Una vez se llegado a este punto, se puede decir que la red definitiva está en producción.

Nota: para los procesos que se plantean en este apartado, se recomienda la utilización de los siguientes módulos de Ansible:

- ios_command
- ios_l2_interface
- ios_l3_interface

3.5.9 Proceso de reporte y documentación

Es altamente recomendable que tras el despliegue automático de toda la configuración se genere un documento que resuma y demuestre que todo está correctamente configurado. Existe la posibilidad de revisar toda la línea de comandos de cada uno de los dispositivos, pero sería poco eficiente. A continuación, en la Ilustración 25: Proceso de documentación y reporte se propone un Playbook que realice un reporte resumido:

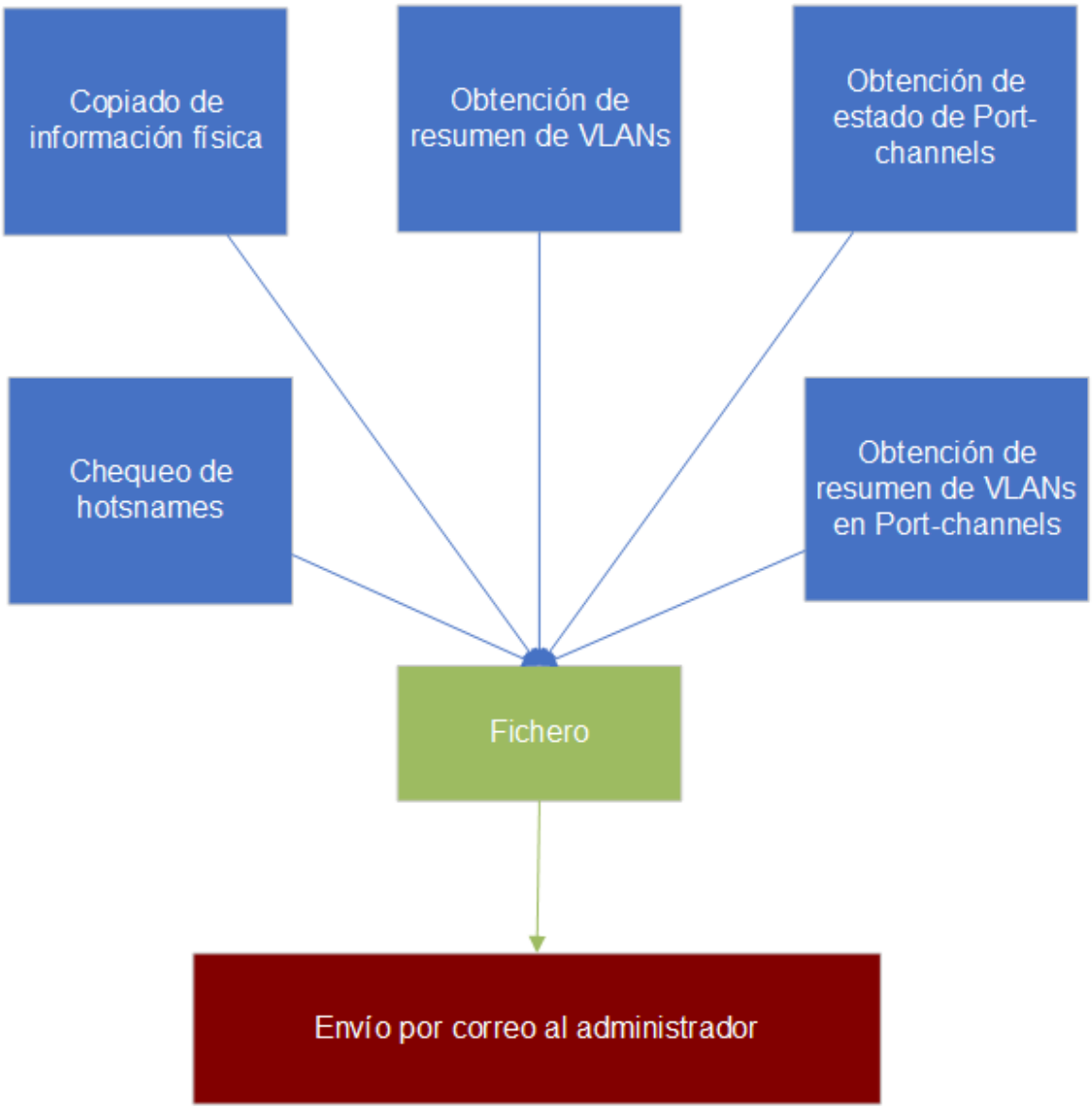

Ilustración 25: Proceso de documentación y reporte

Primeramente, se realizará una comprobación de los nombres de los equipos (hostnames) para compararlos con los descritos en el fichero hosts.

Después, bajo cada uno de los nombres de los equipos se copiará la información física que incluye el "serial number", la dirección MAC principal, el modelo del dispositivo y la versión del software.

A continuación, se obtendrá un resumen de todas las VLANs configuradas en los equipos, que se copiarán en el fichero de manera correlativa, con la finalidad de que sea fácilmente identificable la ausencia de alguna.

Por último, se realizará un recorrido por todas las interfaces físicas que estén incluidas en los Port-Channels. De dichas interfaces se debe

obtener el estado físico (up o down), además del resumen de todas las VLANs que se permiten propagar.

Toda esta información será guardada en un fichero de texto que se adjuntará en un correo electrónico y se enviará al administrador de la red.

Nota: Para este proceso, se recomienda la utilización de los siguientes módulos de Ansible:

- ios_command
- ios_facts

4. Validación y evaluación del proyecto

4.1 Evaluación técnica de la solución

El diseño del Sistema ZTP (Zero Touch Provisioning) ha sido completado. En futuras fases, el diseño debe ser desarrollado hasta obtener un producto válido para su uso. Es probable que se encuentren fallos en el diseño que deban ser corregidos, sin embargo, una futura implementación del sistema propuesto en este trabajo aportará la depuración de los posibles fallos, así como, seguramente, abrirá nuevas posibilidades que no es posible observar desde el plano teórico.

El diseño, dentro de su complejidad técnica, permite cierto margen de mejora. Se pueden rediseñar ciertas partes que harían del Sistema un diseño más completo, eficaz y automático. En el apartado 4.1.2, se desarrollan algunas propuestas de mejora.

4.1.1 Pruebas de validación

A continuación, se plantean una serie de pruebas a realizar tras el desarrollo del diseño que permitirán verificar el correcto funcionamiento del sistema:

1) El sistema funciona correctamente sobre un laboratorio virtualizado.
2) El sistema funciona independientemente de la versión del software de los equipos.
3) Si hay una conexión incorrecta en el cableado de los equipos el sistema alerta al administrador.
4) El sistema obliga a seguir un flujo de procesos de configuración que el administrador no puede obviar.

4.1.2 Propuestas de mejora

4.1.2.1 Usar el entorno gráfico AWX

Como bien se analizó en el apartado 2.1 Análisis de soluciones existentes, parte del código fuente de Ansible Tower fue liberado dando pie a la aparición de AWX (4. (2019). GitHub), que es un entorno gráfico basado en Ansible (3. Ward, B (2018)).

Un entorno gráfico facilitaría mucho las cosas porque abstrae al programador del sistema de ficheros de sistema operativo (Linux en este caso). Además, AWX ofrece una funcionalidad llamada “Callback”, lo que permite a los diferentes equipos solicitar al servidor de manera autónoma la aplicación de diferentes Playbooks. Esta funcionalidad le daría al sistema más independencia de la supervisión humana, si cabe.

Por otro lado, AWX permite el uso de flujos de trabajo como en la Ilustración 26: Captura del editor de flujos de trabajo de AWX. Los flujos de trabajo son agrupaciones de Playbooks que se van ejecutando en la secuencia establecida, pudiendo ser secuencias líneales, circulares, con bucles, con cajas de decisión…

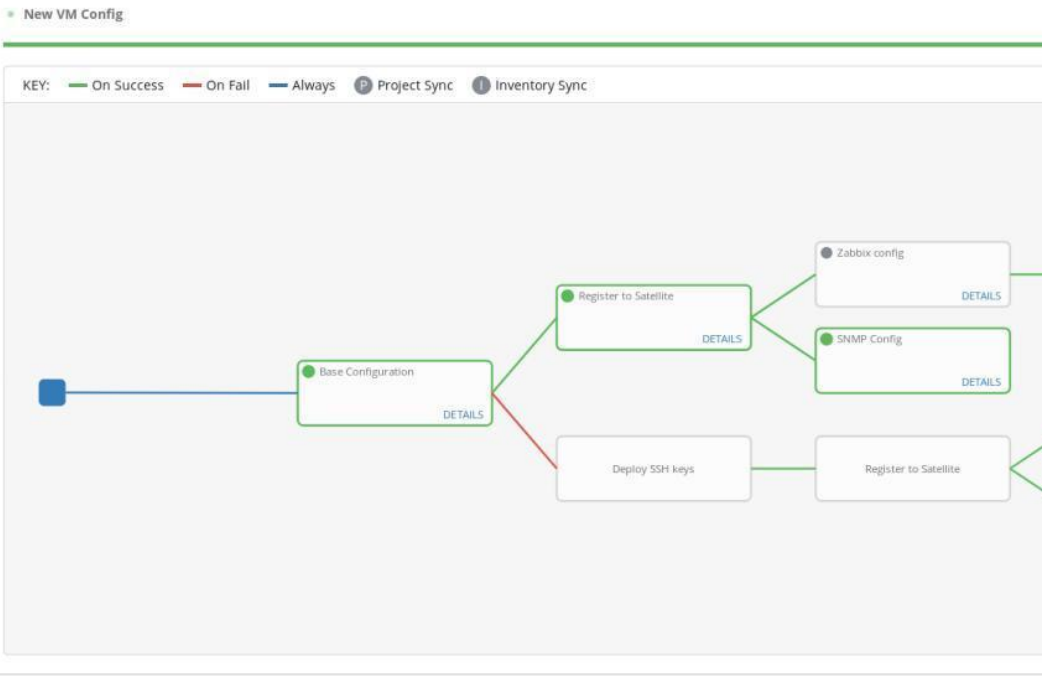

Ilustración 26: Captura del editor de flujos de trabajo de AWX

4.1.2.2 Usar scripts de traducción de Excel a YAML

Uno de los procesos más tediosos y todavía manual de este Sistema es la necesidad de trasladar las plantillas (probablemente hechas en Excel) a ficheros YAML. Hay ingenieros que ya han ideado un método para crear scripts que traducen la información de manera automática. A continuación, se proporciona el enlace de la página de GitHub del ingeniero, cuyo alias es Mamullen (6. Mamullen (2019)), que ha diseñado

el programa que traduce un Excel a ficheros de configuración de Ansible (obviamente, en YAML).

4.1.2.3 Implementar la autogeneración de la configuración

Como última mejora se plantea una vuelta de tuerca más en la automatización de las implementaciones de redes: que la información de la configuración se genere sola.

Es una mejora ambiciosa, pero encaja a la perfección en un Sistema ZTP. Un entorno de red tiene cientos de configuraciones, pero muchas de esas configuraciones son parametrizables. Por ejemplo, los nombres de los equipos son todos diferentes, pero siguen patrones según su rol en la red, por lo que sabiendo la ubicación del dispositivo en la red y el tipo de nombre que se quiere poner, es posible automatizar la generación de todos los nombres que se puedan necesitar.

Otro ejemplo claro es la generación de la IPs de los equipos. Sabiendo la red que se utilizará para esos equipos, se le puede delegar a un script o Playbook la asignación de IPs a los dispositivos según su rol.

4.2 Evaluación económica de la solución

4.2.1 Recursos humanos

Para llevar a cabo el desarrollo de la solución es necesario disponer de diferentes perfiles de ingeniería. En concreto, se necesitarán los siguientes:

- Ingeniero Técnico de Informática (ITI).
- Ingeniero Técnico de Telecomunicaciones (ITT).
- Ingeniero Superior de Telecomunicaciones (IST).

El perfil de ingeniero técnico de Informática es necesario para el montaje y la configuración del entorno del servidor de automatización. Deberá crear la máquina en un entorno de virtualización (o se deberá comprar un servidor), instalarle el sistema operativo e instalar las aplicaciones de Ansible y el TFTP.

El ingeniero técnico de telecomunicaciones, por su parte, será quien desarrolle los Playbooks y configure los ficheros necesarios del entorno Ansible y el TFTP.

Por último, el ingeniero superior de telecomunicaciones será quien lidere el desarrollo del diseño y oriente al ingeniero técnico. Además, deberá afrontar los inconvenientes técnicos que vayan surgiendo y rediseñar las partes del diseño que tengan errores o desajustes.

4.2.2 Coste de personal

Se procede a calcular el coste/hora de cada uno de estos perfiles, partiendo de la estimación de que los perfiles técnicos tienen un salario bruto medio de 30.000€ y que el perfil superior tiene uno de 50.000€.

También se asume que la compañía tiene que añadir a estos salarios la aportación del 31,05% correspondiente a los diferentes gastos imputables a la empresa y que quedan explicados en detalle en el anexo 7.2 del presente trabajo. Aplicando este incremento resulta que el gasto anual correspondiente al personal de perfil técnico asciende a 39.315€ y el de perfil superior sería de 65.525€.

Por otro lado, se prevén jornadas de trabajo de 8 horas, y que en un año laboral medio se realizan unas 1800 horas de trabajo.

Por último, se calcula el importe que corresponderá a cada uno de los perfiles por hora dedicada al proyecto. Tras aplicar un redondeo hacia la unidad superior, resulta el coste de la hora de trabajo de un ingeniero técnico, que se presupuesta en 22€ y la hora del ingeniero superior en 37€. Se tendrán en cuenta estos valores en apartados posteriores para calcular el coste del desarrollo del diseño.

4.2.3 Coste de la infraestructura

Por suerte, la infraestructura necesaria para el desarrollo del diseño es sencilla y no muy costosa.

Los desarrolladores del sistema necesitarán un ordenador de perfil ofimático que se presupone que ya dispondrán. Por este motivo, el coste de esta infraestructura no se contemplará.

Por otro lado, se necesitará un servidor que será el encargado de albergar el sistema Ansible y el servicio TFTP. Puede ser un servidor físico o virtual. Se opta por analizar la opción virtual, porque cualquier empresa en la que sea habitual que vayan surgiendo nuevas oficinas y se justifique tener un sistema ZTP, será una empresa con las dimensiones suficientes como para tener entornos de virtualización.

4.2.4 Planificación del desarrollo del diseño

A continuación, se listarán todas las tareas que se deben acometer para poder obtener el sistema ZTP como producto útil y válido. También se estimarán las horas necesarias para llevar a cabo las tareas, así como el perfil del ingeniero que debe acometerlas, calculando así el coste de los recursos humanos:

Tarea	Horas	Perfil	Coste
Creación de máquina virtual	1	ITI	22,00 €
Instalación de SO en servidor	3	ITI	66,00 €
Instalación de Ansible en servidor	3	ITI	66,00 €
Instalación de TFTP en servidor	1	ITI	22,00 €
Creación de plantilla diagrama físico	8	ITT	176,00 €
Creación de plantilla diagrama lógico	8	ITT	176,00 €
Creación de plantilla datos de EDGE	8	ITT	176,00 €
Creación de plantilla datos de CORE	8	ITT	176,00 €
Creación de plantilla datos de ACCESO	8	ITT	176,00 €
Traducción de plantilla a YAML	32	ITT	704,00 €
Diseño del direccionamiento temporal	1	IST	37,00 €
Interlocución con el operador de red	16	IST	592,00 €
Configuración del servidor TFTP	8	ITT	176,00 €
Desarrollo de Playbook: reporte topología de red	90	ITT	1.980,00 €
Desarrollo de Playbook: configuración específica de CORE	120	ITT	2.640,00 €
Desarrollo de Playbook: configuración general de CORE	120	ITT	2.640,00 €
Desarrollo de Playbook: configuración de routing	90	ITT	1.980,00 €
Desarrollo de Playbook: configuración específica de ACCESO	120	ITT	2.640,00 €
Desarrollo de Playbook: configuración general de ACCESO	120	ITT	2.640,00 €

Desarrollo de Playbook: cambio de puerta de enlace ACCESO	40	ITT	880,00 €
Desarrollo de Playbook: modificación del fichero "hosts"	48	ITT	1.056,00 €
Desarrollo de Playbook: Limpieza de configuración temporal	60	ITT	1.320,00 €
Desarrollo de Playbook: Reporte final	40	ITT	880,00 €
Montaje de entorno de pruebas	8	ITT	176,00 €
Desarrollo de ficheros y plantillas para pruebas	16	ITT	352,00 €
Pruebas sobre equipamiento real	60	ITT	1.320,00 €
Resolución de errores	20	ITT	440,00 €
Resolución de errores	20	IST	740,00 €
Ajuste del diseño	90	IST	3.330,00 €
Documentación de los ajustes del diseño	40	IST	1.480,00 €
Total horas:	1207	Total coste:	29.059,00 €

4.2.5 Valoración económica

Cabe mencionar que el coste total obtenido (a falta de sumarle un coste mínimo de la infraestructura) resulta muy asequible para una compañía multinacional. No obstante, la dificultad de defender económicamente el proyecto no residirá estrictamente en las cifras presupuestadas, sino en convencer a la Dirección de la compañía de que esta inversión en horas de ingeniería reportará un gran ahorro de horas de ingeniería en futuros despliegues de redes.

5. Conclusiones

En las siguientes líneas se reflexionará sobre el trabajo realizado valiéndose de los objetivos planteados al inicio y analizando su grado de consecución.

Se han analizado varias de las soluciones de automatización existentes en el mercado, comparando sus diferentes características técnicas y también su coste por lo que el objetivo está conseguido.

La selección de la herramienta fue fruto de la comparativa mencionada en el párrafo anterior. Si bien todas las herramientas presentaban facilidades para la automatización y orquestación de redes, Ansible era la más avanzada de todas. Además, Ansible es una herramienta “open-source” gratuita y muy respalda por una muy amplia comunidad de desarrolladores.

También se han analizado los nuevos enfoques en la administración de redes. Se han mencionado varios ámbitos diferentes de aplicación de la automatización, pero finalmente se optó por el ámbito del despliegue de redes desde cero (Zero Touch Provisioning).

Por supuesto, se considera claramente alcanzado el objetivo de conseguir diseñar una solución y su implementación en una empresa. Todo este trabajo gira en torno a este objetivo. Se ha profundizado en los diferentes aspectos del sistema y se ha estudiado su implantación en una empresa.

Para que esta implantación en empresa sea lo más orientada posible, se han desarrollado diagramas que explican de forma nítida como ha de ser el comportamiento del sistema, tal y como se había previsto en los objetivos.

También se ha logrado detallar un listado de los parámetros a configurar en los diferentes tipos de dispositivos que se enmarcan en este proyecto. No obstante, habría sido considerado más exitoso lograr unas plantillas de Excel polivalentes para facilitar la integración del sistema en un entorno corporativo.

También se han estudiado los fundamentos de la administración remota de equipos de red. No hay ningún apartado que profundice en este aspecto en concreto, pero en todas las fases del sistema se dan

explicaciones claras de cómo proceder. Se hace especial hincapié en el orden que ha de seguir el proceso y de la criticidad que tiene esto para no perder la conexión con el equipo remoto.

Por otro lado, se ha analizado el coste de implantación del sistema en una empresa. El objetivo está claramente conseguido y se ofrece una panorámica clara de la inversión que tendría que hacer una compañía para desarrollar el Sistema.

Por último, es relevante mencionar que, durante el transcurso de este trabajo, la compañía que se ha tomado como referencia para plantear el escenario ha mostrado un gran interés por el proyecto.

6. Bibliografía

A continuación, se listan las diferentes fuentes que se han consultado para la realización este trabajo, ordenadas por tema:

Productos de automatización

1. Suripa, K (2016). Chef, Puppet & Ansible Network Automation: https://www.anutanetworks.com/network-automation-with-chef-puppet-ansible/

AWX

2. Ward, B (2018). Install Ansible on Ubuntu 18.04 with AWX: https://www.admintome.com/blog/install-ansible-on-ubuntu-18-04-with-awx/
3. (2019). GitHub – ansible/awx: AWX Project: https://github.com/ansible/awx

Ansible

4. Edelman, J (2016). 6 reasons to use Ansible for network automation: https://www.oreilly.com/ideas/6-reasons-to-use-ansible-for-network-automation
5. Mamullen (2019). GitHub – mamullen13316/ansible_xls_to_facts: https://github.com/mamullen13316/ansible_xls_to_facts
6. Erikruiter2 (2019) ansible_lab/roles/cisco_lldp_topo at master: https://github.com/erikruiter2/ansible_lab/tree/master/roles/cisco_lldp_topo
7. Kuchenski, D (2017). Ansible – VLAN Provisionig: https://thenetworkstack.com/ansible-vlan-provisioning/
8. Kashin, M (2015). Getting Started With Ansible for Cisco IOS: https://networkop.co.uk/blog/2015/06/24/ansible-intro/
9. (2019). Ansible is Simpel IT Automation: https://www.ansible.com/
10. (2019). Ansible Tower | Ansible.com: https://www.ansible.com/products/tower
11. De la Cruz, J (2018). Ansible: Qué es AWX, instalación, configuración, PlayBooks para Windows y Linux y ¡mucho más!: https://www.jorgedelacruz.es/2018/08/15/ansible-que-es-awx-instalacion-configuracion-playbooks-para-windows-y-linux-y-mucho-mas/
12. (2019). Network modules – Ansible Documentation: https://docs.ansible.com/ansible/latest/modules/list_of_network_modules.html

Puppet

13. (2019). Solutions – Deliver better software faster with Puppet: https://puppet.com/solutions
14. (2019). Cisco IOS Catalyst Module – Thank you: https://puppet.com/resources/solution-brief/cisco-ios-catalyst-module/thank-you
15. (2018). Puppet (software): https://en.wikipedia.org/wiki/Puppet_(software)

Chef

16. (2019): Chef – Deploy new code faster and more frecuently: https://www.chef.io/

Automatización

17. Hans-vvv (2018). GitHub – hans-vvv/NetAutLab: https://github.com/hans-vvv/NetAutLab
18. Miloslavsky, A (2017). Infrastructure As Code For The Network – Part 1 – Introduction And Day 0 Provisioning: https://packetpushers.net/infrastructure-as-code-for-the-network-stack/

ZTP

19. Dainese, A (-). Zero Touch Provisioning: https://www.ipspace.net/kb/CiscoAutomation/030-ztp.html
20. Pepelnjak, I (2018). Automation Win: Zero-Touch Provisioning: https://blog.ipspace.net/2018/05/automation-win-zero-touch-provisioning.html
21. Pepelnjak, I (2017). Automate Remote Site Hardware Refresh Process: https://blog.ipspace.net/2017/12/automate-remote-site-hardware-refresh.html
22. Ogenstad, P (-). Zero Touch Provisioning DIY Tutorial: https://networklore.com/ztp-tutorial/
23. Ogenstad, P (2018). GitHub – networklore/ztp-tutorial: https://github.com/networklore/ztp-tutorial

Aspectos relacionados con los trabajadores de una compañía

24. (2016). Caso práctico: Cómputo anual de la jornada de trabajo cuando no se especifique nada en convenio colectivo:

https://www.iberley.es/practicos/caso-practico-computo-anual-jornada-trabajo-no-especifique-convenio-colectivo-3231
25. (2016). Sencillo método para calcular el coste de un trabajador: http://www.eurogabinete.com/sencillo-metodo-para-calcular-el-coste-de-un-trabajador/
26. Santos Pascualena, J (2017). ¿Cuánto cuesta contratat un trabajador?: https://infoautonomos.eleconomista.es/blog/cuanto-cuesta-contratar-un-trabajador/

Otros

27. (2005). Lenguaje declarativo: http://enciclopedia.us.es/index.php/Lenguaje_declarativo

7. Anexos

7.1 Diagrama lógico de red de oficina central

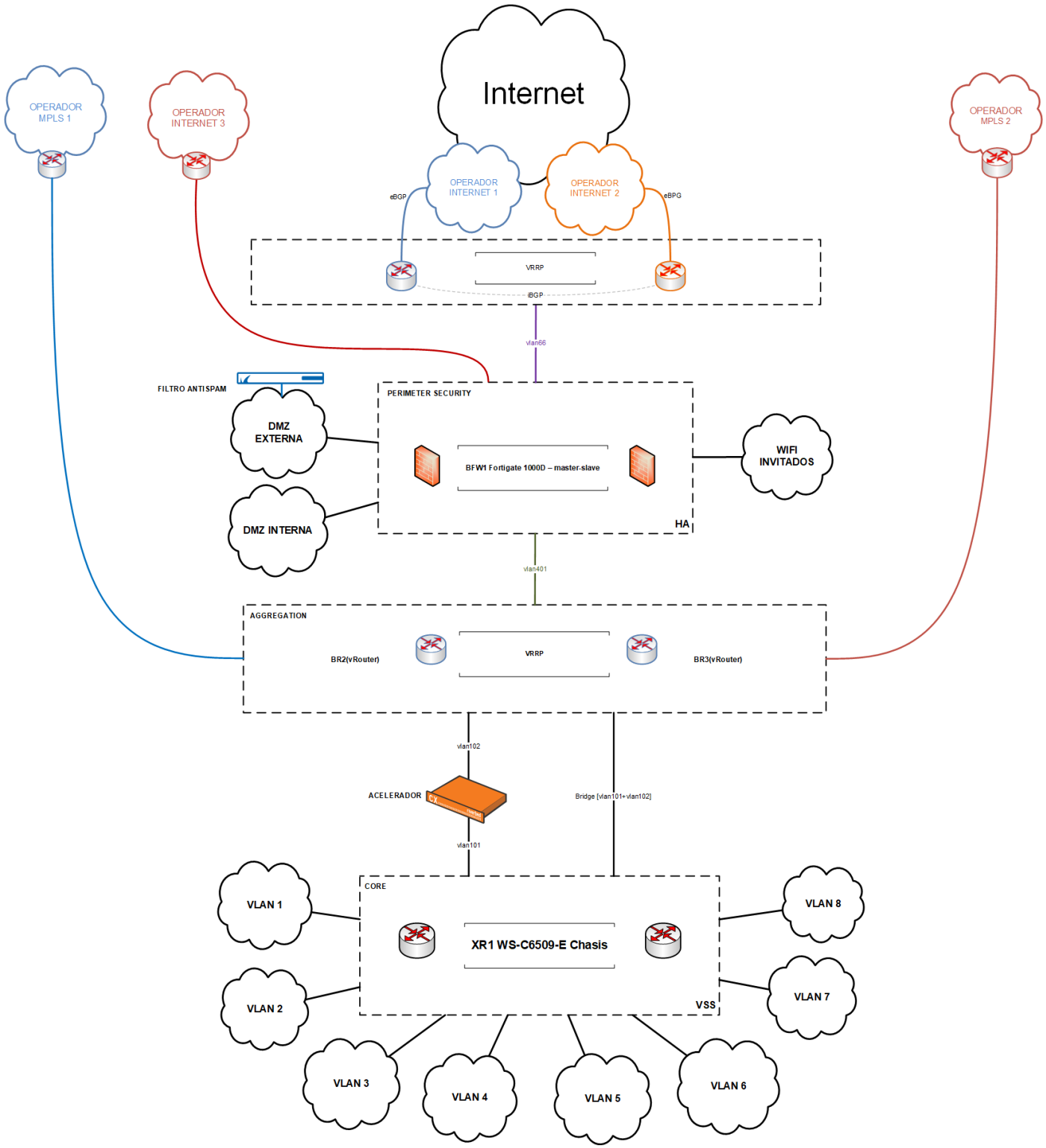

Ilustración 27: Diagrama lógico de una sede central

7.2 Justificación del cálculo de coste empresarial en relación al salario que percibe el trabajador/a.

Se utiliza como referencia la información contenida en el artículo que se menciona en el enlace. Dicha información ha sido contrastada con personal de administración en activo. Así mismo, el contenido se ha reelaborado para adecuarlo al interés de este trabajo y para concentrar la información pertinente.

https://www.pymesyautonomos.com/fiscalidad-y-contabilidad/cuanto-paga-tu-empresa-por-ti-quiza-mas-de-lo-que-piensas

CONCEPTO	A CARGO DEL TRABAJADOR	A CARGO DE LA EMPRESA	TOTAL
CONTINGENCIAS COMUNES	4,70%	23,60%	28,30%
AT Y EP	0%	1,35%	1,35%
DESEMPLEO	1,55%	5,50%	7,05%
FORMACIÓN	0,10%	0,60%	0,70%
TOTAL	6,35%	31,05%	37,4%

Estos tipos se aplican sobre la base de cotización, que se calcula teniendo en cuenta todas las remuneraciones obtenidas por el trabajador, incluido el salario en especie, y prorrateando la parte proporcional de las pagas extras.

Ejemplo práctico:

Supongamos una persona con un sueldo mensual de unos 1.300 euros. La base de cotización de esta persona es de 2.000 euros al mes, una vez hemos tenido en cuenta todas las remuneraciones y la parte proporcional de las pagas extras. La empresa tendrá que pagar por ese empleado según los siguientes conceptos:

- **IRPF**: supongamos un 12,89 por ciento sobre la base del IRPF (no tenemos en cuenta las pagas extras, que serían de 270 euros); en total, 223 euros (12,89 * 1.730).
- **Descuentos por contingencias comunes**: 94 euros a cargo del empleado y 472 a cargo de la empresa. En total, 566 euros.
- **AT y EP**: 27 euros a cargo de la empresa.

- **Desempleo**: 31 euros a cargo del trabajador y 110 euros a cargo de la empresa. En total, 141 euros.
- **Formación profesional**: 2 euros a cargo del trabajador y 12 euros a cargo de la empresa.
- **FOGASA**: 4 euros a cargo de la empresa.

El coste laboral unitario de un empleado con un sueldo neto de 1.300 euros es, sumando todos estos conceptos, de 2.275 euros. Esta cantidad se reparte de la siguiente manera:

Sueldo neto que recibe el trabajador/a	1.300 €
Aportación de trabajador a SS	127 €
Retención IRPF	223 €
Aportación de la empresa a SS	625 €
Total	2.275 €

Otros artículos de interés sobre el tema:

- http://www.eurogabinete.com/sencillo-metodo-para-calcular-el-coste-de-un-trabajador/
- https://infoautonomos.eleconomista.es/blog/cuanto-cuesta-contratar-un-trabajador/