

InfraRed
SOLUTIONS

PROYECTO ASIR 24-25

Edgar Díaz Martínez
72193631N | IES MIGUEL HERRERO PEREDA.

ÍNDICE:

INTRODUCCIÓN	3
OBJETIVOS:.....	4
ANÁLISIS DEL CONTEXTO Y ESTADO DEL ARTE	5
1. ¿Cuál es mi idea?	5
2. Otras soluciones al mismo problema	6
3. Empresas y soluciones que utilizan estas tecnologías	10
4. Innovación y ventajas competitivas de este proyecto	10
5. Aplicación y difusión del proyecto en plataformas profesionales.....	11
REQUISITOS	13
DISEÑO	16
1. Diagrama de flujo funcional del proceso de despliegue.....	17
¿Por qué no se implementa una base de datos en el proyecto?.....	17
2. Diagrama lógico de red	18
3. Infraestructura desplegada.....	19
PLANIFICACIÓN.....	21
1. Definición de recursos y logística necesarios.....	21
2. Procedimientos por actividad	22
3. Identificación de riesgos y plan de prevención	24
IMPLEMENTACIÓN, PRUEBAS Y PUESTA EN MARCHA.....	25
Instalación de VMware Workstation Pro	26
Instalación de VMware ESXI.....	27
Instalación de Packer.....	36
Instalación de Terraform.....	40
Creación de imagen base con Packer en VMware Workstation.....	42
Preparación del entorno	42
Automatización del despliegue con Terraform sobre vSphere.	47
Despliegue con Terraform.....	51
Explicación detallada del archivo main.tf.....	51
Declaración de variables – variables.tf.....	54
Valores concretos - terraform.tfvars	55

Ejecución del código	55
Visualización en vSphere	57
Entorno completamente desplegado	58
Pruebas.....	59
Creación de la imagen base con Packer	59
Despliegue con Terraform desde imagen creada en VMware Workstation Pro.....	60
Elección de VMware vSphere en lugar de ESXi Free para el despliegue con Terraform	61
Limitaciones de Packer y Terraform en la creación de imágenes	63
Repositorio en GitHub del proyecto	64
PARTE ECONÓMICA PLAN DE EMPRESA	65
Datos del emprendedor.....	65
Nombre de la idea y logo	67
Responsabilidad social	67
Estudio de validación interna del proyecto InfraRed.....	68
Análisis de competencia interna y externa.....	71
Competencia interna (soluciones actualmente en uso).....	72
Competencia externa (herramientas del mercado)	72
Equipo responsable del desarrollo y mantenimiento del proyecto InfraRed	75
Coste salarial total anual de los trabajadores para la empresa:.....	78
Mutua de accidentes y servicios de prevención de riesgos laborales	79
Inventario de los elementos necesarios.....	83
Localización dentro de la empresa y relación con otros departamentos o áreas.....	85
CONCLUSIONES Y VALORACIÓN PERSONAL	87
ANEXOS Y BIBLIOGRAFIA:.....	89

INTRODUCCIÓN

En la actualidad, muchas empresas dedicadas a la consultoría de software, desarrollo y despliegue de soluciones informáticas, trabajan a diario con entornos virtuales. Estas empresas necesitan realizar pruebas, montar laboratorios de desarrollo y preproducción, o simular infraestructuras de clientes antes de desplegarlas en entornos reales.

El problema aparece cuando estos entornos se tienen que crear repetidamente de forma manual, lo cual genera una pérdida de tiempo significativa, inconsistencias entre entornos y una carga adicional sobre los administradores de sistemas. A medida que los proyectos crecen, este proceso se vuelve insostenible.

El proyecto parte de un supuesto realista: formar parte del departamento de sistemas de una empresa de consultoría de software que trabaja con máquinas virtuales en VMware. Ante la necesidad de optimizar recursos y estandarizar despliegues, surge la propuesta de automatizar la creación de entornos virtuales utilizando herramientas como Packer y Terraform, con el objetivo de simplificar tareas repetitivas y reducir el tiempo dedicado a la preparación de entornos, sin necesidad de aplicar configuraciones mediante código.

Mi proyecto tiene como objetivo automatizar el proceso de creación y despliegue de máquinas virtuales en VMware, utilizando Packer para generar imágenes personalizadas con configuraciones predefinidas, y Terraform para desplegarlas fácilmente mediante código. Este enfoque permite reducir tareas manuales, mejorar la eficiencia del administrador de sistemas y estandarizar entornos virtuales de forma sencilla y práctica.

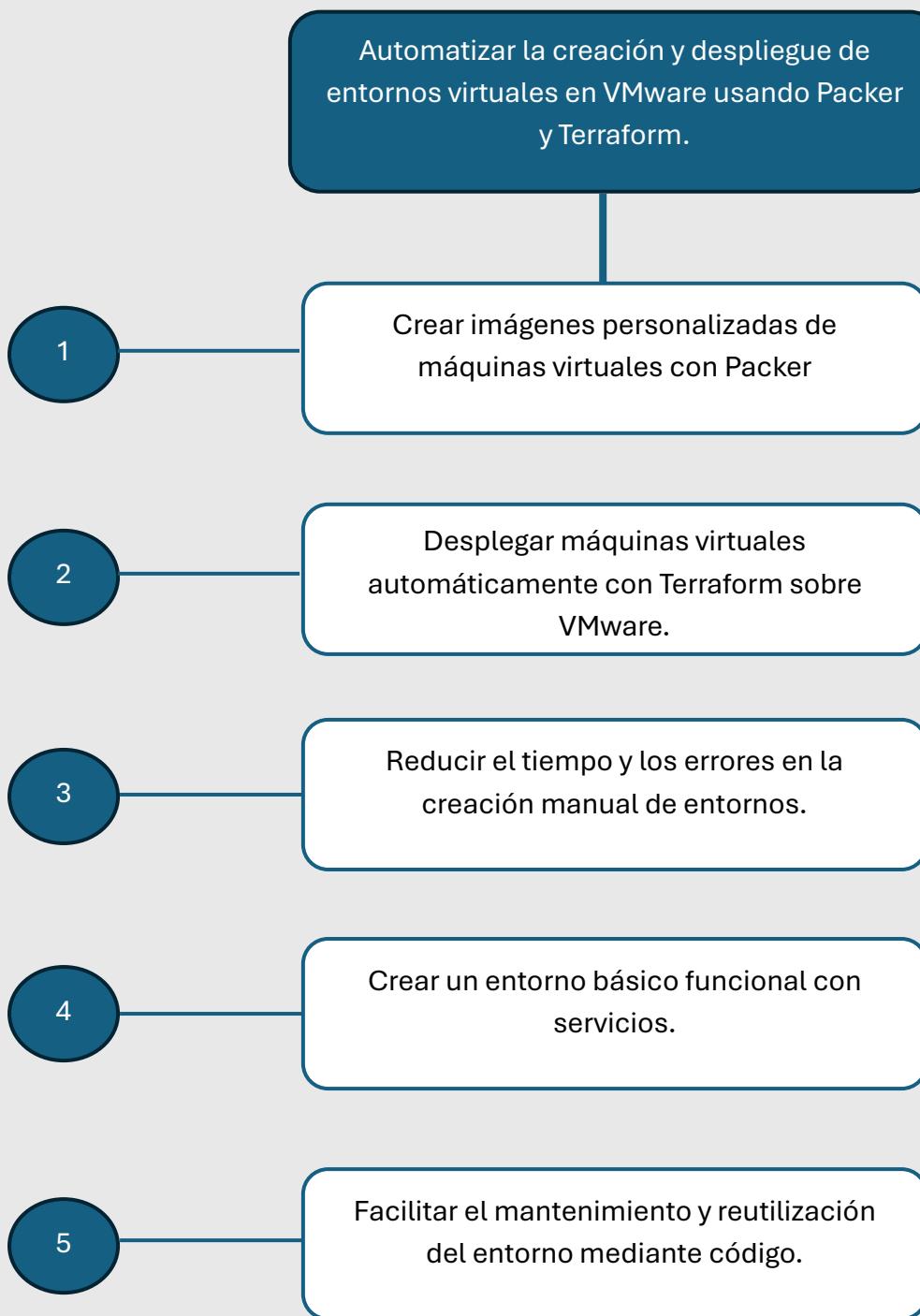
Además de mejorar la eficiencia operativa, esta solución permite una mayor trazabilidad y control sobre los entornos, ya que toda la infraestructura se define y versiona como si fuera software. Esto facilita no solo la repetición de entornos idénticos, sino también la auditoría y evolución de la infraestructura de forma estructurada.

El proyecto, llamado InfraRed, representa un paso hacia la modernización del departamento de sistemas, alineándose con las tendencias actuales del sector IT, donde la automatización, la infraestructura como código y la virtualización avanzada son pilares fundamentales para garantizar agilidad, escalabilidad y fiabilidad en los servicios ofrecidos a clientes.

A lo largo de este proyecto se abordarán los diferentes aspectos necesarios para llevar a cabo esta automatización, desde la definición de los objetivos hasta el análisis de las herramientas, el diseño de la solución, la planificación del desarrollo y la implementación práctica. También se reflexionará sobre su aplicabilidad en un entorno real y su impacto en la dinámica de trabajo del departamento de sistemas, poniendo en valor el uso de tecnologías actuales para optimizar procesos y mejorar la eficiencia operativa.

OBJETIVOS:

Los objetivos del proyecto se enfocan en ofrecer una solución que permita automatizar tanto la generación de imágenes base como el aprovisionamiento completo de máquinas virtuales, utilizando herramientas modernas que ya forman parte del ecosistema profesional actual. De este modo, se busca mejorar la eficiencia operativa del departamento de sistemas y dotarlo de una metodología escalable, mantenible y alineada con las prácticas de infraestructura como código.



ANÁLISIS DEL CONTEXTO Y ESTADO DEL ARTE

En un entorno tecnológico cada vez más dinámico, la automatización de infraestructuras se ha convertido en una prioridad para muchas empresas del sector IT. La necesidad de contar con entornos de trabajo rápidos de desplegar, coherentes y reutilizables ha llevado a la adopción de soluciones que permiten simplificar tareas repetitivas y reducir el esfuerzo manual del equipo técnico. Esta automatización no solo mejora la eficiencia operativa, sino que también permite responder de forma más ágil a los requerimientos de distintos proyectos y clientes.

Dentro de este contexto, el proyecto InfraRed se plantea como una solución moderna para una empresa ficticia de consultoría tecnológica que trabaja de forma intensiva con máquinas virtuales sobre VMware. El objetivo es automatizar tanto la generación de imágenes como el despliegue de máquinas virtuales, utilizando herramientas ampliamente utilizadas como Packer y Terraform, a través de configuraciones funcionales adaptadas al entorno real del departamento de sistemas.

1. ¿Cuál es mi idea?

A lo largo de mi formación en ASIR, me he dado cuenta de lo repetitivo que puede ser montar entornos virtuales cada vez que se necesita hacer una prueba, practicar algo o preparar una demo. Por eso, pensé: ¿y si pudiera crear un sistema que hiciera todo esto por mí en unos pocos pasos?

Así nació la idea de este proyecto: una solución sencilla pero potente que permita automatizar la creación de entornos virtuales usando Packer y Terraform en VMware. No se trata solo de levantar máquinas, sino de tener un entorno funcional, listo para trabajar, en apenas unos minutos.

¿Qué me propongo?

- Que cualquier técnico o estudiante pueda generar su entorno de pruebas sin tocar el ratón.
- Que se aprovechen imágenes creadas con Packer para evitar repetir instalaciones.
- Que Terraform se encargue de levantar las máquinas necesarias, con la configuración exacta.
- Que todo quede definido en código: claro, documentado y fácil de modificar.
- Y, sobre todo, que funcione sobre VMware, porque muchas veces trabajamos en local y no en la nube.

¿Qué tiene de diferente?

No estoy inventando herramientas nuevas, pero sí estoy uniendo lo mejor de varias: la personalización de Packer, la potencia de Terraform y la fiabilidad de VMware. El resultado es un sistema útil, práctico y muy fácil de ampliar o adaptar.

Ideal para:

- Laboratorios de prácticas.
- Equipos de pruebas.
- Formación técnica.
- Profesionales que quieren ahorrar tiempo.

En resumen, mi idea no es crear algo complejo, sino algo que simplemente funcione bien, y que cualquiera pueda usar para hacer su trabajo más fácil.

2. Otras soluciones al mismo problema

Herramientas similares a Packer y Terraform

Existen otras alternativas para automatizar la infraestructura y la creación de entornos virtuales:

Herramienta	Descripción breve	Fuente
Vagrant	Facilita la creación de entornos reproducibles, especialmente con VirtualBox.	Vagrant
Ansible	Permite automatizar la configuración de sistemas una vez desplegados.	Ansible
Docker	Utilizado para contenerización, no para virtualización completa.	Docker
VMware vSphere	Entorno empresarial para gestionar grandes infraestructuras virtualizadas.	vSphere

Herramientas utilizadas en el proyecto

Packer

Packer es una herramienta desarrollada por HashiCorp que permite automatizar la creación de imágenes de sistemas operativos preconfigurados para diferentes plataformas de virtualización o cloud. A través de scripts y archivos de configuración, es posible incluir software, configuraciones de red, usuarios, claves SSH, y otros elementos en una imagen reutilizable.

- Fortalezas:
 - Compatible con múltiples plataformas (VMware, VirtualBox, AWS, Azure...).
 - Generación rápida y consistente de imágenes listas para usar.
 - Fácil integración con sistemas de CI/CD.
- Debilidades:
 - No realiza despliegues; solo crea la imagen.
 - Requiere conocimientos previos en scripting y configuración de SO.
- Tendencia de uso:
 - Ampliamente adoptado en empresas que buscan estandarización.
 - Muy utilizado en pipelines de automatización para entornos de testing o integración.



Terraform

Terraform, también de HashiCorp, es una herramienta de aprovisionamiento de infraestructura declarativa. Permite describir con código el entorno deseado (máquinas, redes, almacenamiento...) y luego aplicar esa configuración automáticamente. En este proyecto, se usa para desplegar máquinas virtuales en VMware a partir de las imágenes generadas por Packer.

- Fortalezas:
 - Declarativo, reutilizable y versionable.
 - Permite gestionar infraestructura local y en la nube.
 - Amplio soporte de proveedores (AWS, VMware, Azure, GCP...).
- Debilidades:
 - Requiere configuración inicial detallada.
 - Necesita comprensión de la sintaxis HCL y del ciclo de estados.
- Tendencia de uso:
 - Es una de las herramientas más populares en DevOps.
 - Presente en empresas de todos los tamaños, desde startups hasta grandes corporaciones.



VMware

VMware es uno de los hipervisores más utilizados en entornos empresariales. Su rendimiento, estabilidad y flexibilidad lo convierten en un estándar para virtualización en infraestructuras locales (on-premises).

- Fortalezas:
 - Altamente estable, rendimiento robusto.
 - Ampliamente implantado en empresas e instituciones.
 - Buen soporte de red, snapshots, almacenamiento y virtualización anidada.
- Debilidades:
 - Modelo de licencias restrictivo.
 - Requiere máquinas potentes y configuraciones complejas.
- Tendencia de uso:
 - Sigue siendo la base de muchas infraestructuras locales.
 - Combinado cada vez más con soluciones de automatización como Ansible, Terraform o Puppet.

vmware[®]
by **Broadcom**

3. Empresas y soluciones que utilizan estas tecnologías

HashiCorp

- Empresa desarrolladora de Packer y Terraform.
- Ofrece herramientas ampliamente utilizadas en empresas de todos los tamaños.
- Web oficial: <https://www.hashicorp.com>

VMware

- Líder mundial en virtualización, utilizado por millones de empresas.
- Su software es compatible con infraestructura local y en la nube.
- Web oficial: <https://www.vmware.com>

DevOps y formación IT

- Escuelas como Udemy, Pluralsight, y plataformas como GitHub ofrecen laboratorios de formación con estas herramientas.
- Son estándar en procesos de CI/CD y en despliegue de entornos de prueba.

4. Innovación y ventajas competitivas de este proyecto

Este proyecto destaca frente a otras soluciones manuales o semiautomáticas por:

- **Automatización completa:** desde la imagen base hasta el despliegue funcional.
- **Entorno reproducible:** cualquier VM creada será idéntica a la anterior.
- **Servicios preinstalados:** la imagen creada con Packer incluye servidores listos para pruebas (como Apache o Nginx).
- **Ahorro de tiempo:** ideal para profesores, técnicos, o entornos de laboratorio.
- **Escalabilidad local:** puede adaptarse fácilmente para desplegar más VMs con pocos cambios en el código.
- **Estándares del sector:** se emplean herramientas usadas en entornos reales.

5. Aplicación y difusión del proyecto en plataformas profesionales

Aunque este proyecto está pensado principalmente como una solución técnica para automatizar el despliegue de entornos virtuales en VMware mediante Packer y Terraform, también tiene un gran potencial para ser compartido, documentado y aprovechado por otros profesionales, estudiantes o entornos educativos.

A continuación, se detallan las principales plataformas y redes sociales donde este proyecto podría difundirse y tener impacto:

Plataformas técnicas

- **GitHub**

Lugar ideal para publicar el código fuente del proyecto: configuraciones de Packer, archivos .tf de Terraform, scripts de provisionamiento y documentación técnica.

<https://github.com>

- **LinkedIn**

Red profesional perfecta para dar a conocer el proyecto, compartir aprendizajes, mostrar capturas o enlaces al repositorio.

<https://linkedin.com>

- **Dev.to**

Comunidad técnica donde se pueden escribir artículos explicando el proceso del proyecto, herramientas usadas, problemas y soluciones.

<https://dev.to>

- **YouTube**

Canal ideal para mostrar una demo del proyecto, cómo se despliega una máquina virtual desde código y el funcionamiento paso a paso.

<https://youtube.com>

- **Reddit (r/devops / r/sysadmin / r/homelab)**

Foros donde recibir feedback de otros técnicos, compartir configuraciones y mejorar el proyecto con aportes reales.

<https://reddit.com/r/devops>

Redes sociales educativas o de contenido

- **Instagram/TikTok**

Ideales para crear clips breves mostrando el proceso de despliegue, explicar cómo funciona el flujo Packer - Terraform -VMware, enfocado al aprendizaje de otros.



- **Twitter**

Útil para publicar consejos, comandos útiles, capturas de pantalla y reflexiones sobre el proyecto, usando hashtags como #Terraform, #Packer, #VMware, #InfraestructuraComoCódigo.



Aplicación educativa

- **Campus virtual o Moodle**

Este proyecto puede ser usado como recurso práctico en módulos de administración de sistemas, redes o virtualización, permitiendo a estudiantes reproducir el entorno y aprender automatización real.



- **GitBook / Notion**

Puede convertirse en una guía paso a paso o manual técnico interactivo, compatible entre compañeros o como portfolio profesional.



Este apartado refleja el valor no solo técnico, sino también comunicativo y formativo del proyecto, permitiendo que trascienda más allá del aula y pueda crecer como solución práctica en la comunidad.

REQUISITOS

Como técnico del departamento de sistemas en un entorno simulado de consultoría tecnológica, detecté la necesidad de optimizar el proceso de preparación de entornos virtuales utilizados habitualmente en pruebas y desarrollo. Hasta ahora, la creación manual de máquinas virtuales suponía una pérdida de tiempo y una gran repetición de tareas.

Con el objetivo de estandarizar y automatizar ese proceso, desarrollé InfraRed: una solución basada en la automatización completa del despliegue de máquinas virtuales mediante Packer y Terraform, inicialmente sobre VMware Workstation Pro y finalmente integrada con un entorno empresarial real como vSphere (vCenter + ESXi).

Requisitos funcionales

Para que InfraRed sea útil y escalable en un entorno realista, debía cumplir los siguientes objetivos:

- Generar plantillas preconfiguradas de máquinas virtuales que incluyan sistema operativo y servicios básicos.
- Incluir scripts de provisión automática desde el primer arranque.
- Evitar la configuración manual de red: todas las VMs deben poder integrarse fácilmente en la red definida por el entorno de virtualización.
- Permitir el reutilización de imágenes generadas sin tener que repetir procesos.
- Ser fácil de utilizar, incluso para técnicos sin experiencia previa en automatización.
- Reducir notablemente los tiempos de despliegue respecto al método manual tradicional.

Requisitos técnicos

Desde un punto de vista técnico, InfraRed está basado en:

- Uso de VMware Workstation Pro como entorno inicial de desarrollo y validación de imágenes.
- Utilización de Packer para generar imágenes automatizadas de Ubuntu Server desde una ISO oficial.
- Despliegue automatizado en vSphere (vCenter + ESXi) usando Terraform como herramienta IaC.
- Conversión del disco .vmdk a formato compatible con ESXi y subida al datastore para ser reutilizado por Terraform.
- Toda la solución está basada en infraestructura local, sin depender de servicios en la nube ni entornos externos.

Requisitos de hardware (basado en inventario real)

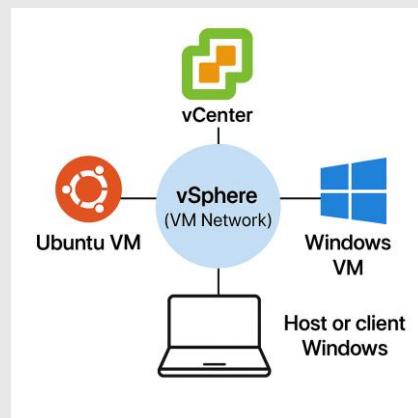
Producto / Recurso	Precio (€)	Proveedor
Dell Precision 5820 Workstation – Xeon, 32 GB RAM, 512 GB SSD	699,00	PcComponentes
2x Monitor Dell QHD USB-C 27" – S2722DC	259,18	Dell
SSD Kingston KC600 1 TB SATA3	99,99	PcComponentes
VMware Workstation 17.5 Pro – Licencia para 12 dispositivos	21,90	codigi.es
Servidor bare-metal Dell PowerEdge R940	325,00	BargainHardware
Switch TP-Link Gigabit 24 puertos – TL-SG1024D	80,79	Amazon
Router existente de la empresa	0,00	—



Requisitos de red

Para que la infraestructura funcione correctamente y las máquinas virtuales se comuniquen entre sí, establecí los siguientes criterios:

- Conectividad total entre VMs mediante red virtual definida en vSphere (VM Network).
- Configuración básica de red gestionada desde el vCenter, sin necesidad de pfSense ni clientes puenteados.
- Posibilidad de acceder a las VMs desde el host o desde cualquier cliente Windows conectado a la misma red del laboratorio.



Software adicional requerido

- **Ubuntu Server 24.04 LTS**

Descargado en formato ISO desde la web oficial de Ubuntu.

- **Packer y Terraform**

Descargados desde la página oficial de [HashiCorp](#) en sus versiones estables.

DISEÑO

En uno de los apartados anteriores analicé en profundidad cuál es la necesidad interna que ha motivado el desarrollo del proyecto InfraRed: la automatización del despliegue de entornos virtuales dentro del departamento de sistemas de una empresa de consultoría tecnológica. Ahora que ya conozco esa necesidad y el producto que he desarrollado, en esta fase de diseño establezco un protocolo de actuación claro y realista, que muestre cómo se atiende la demanda del cliente interno de forma automatizada, estandarizada y reproducible.

Como todo proceso, la fase de diseño comienza con una necesidad detectada. En este caso, el cliente no es externo, sino el propio equipo técnico, que requiere una solución eficiente y escalable para gestionar máquinas virtuales sin tener que repetir procesos de instalación manual cada vez. InfraRed es precisamente la respuesta a esa necesidad, desarrollada desde dentro del departamento IT para optimizar tiempos, reducir errores humanos y facilitar el mantenimiento.

Protocolo técnico de actuación

El procedimiento completo se basa en un flujo de trabajo estructurado y automatizado, que he conseguido implementar con éxito tras varias pruebas y ajustes:

1. Generación de imagen base con Packer

Utilizo Packer para automatizar la creación de una máquina virtual Ubuntu Server personalizada. Esta imagen incluye servicios preconfigurados (Apache, SSH) y está diseñada para desplegarse de forma reproducible.

2. Exportación de la imagen desde VMware Workstation Pro

Como el entorno inicial es local, creo la imagen en VMware Workstation. Posteriormente, convierto el disco .vmdk a un formato plano compatible con VMware ESXi, y lo subo manualmente al datastore correspondiente (SVR1_OS) dentro del servidor.

3. Despliegue automatizado con Terraform conectado a vCenter

Una vez la imagen está disponible en el servidor ESXi, me conecto al vCenter Server (192.168.0.20) mediante Terraform. Desde allí, defino y despliego una máquina virtual nueva que utiliza ese disco como base. Este paso representa la aplicación del modelo de Infraestructura como Código (IaC).

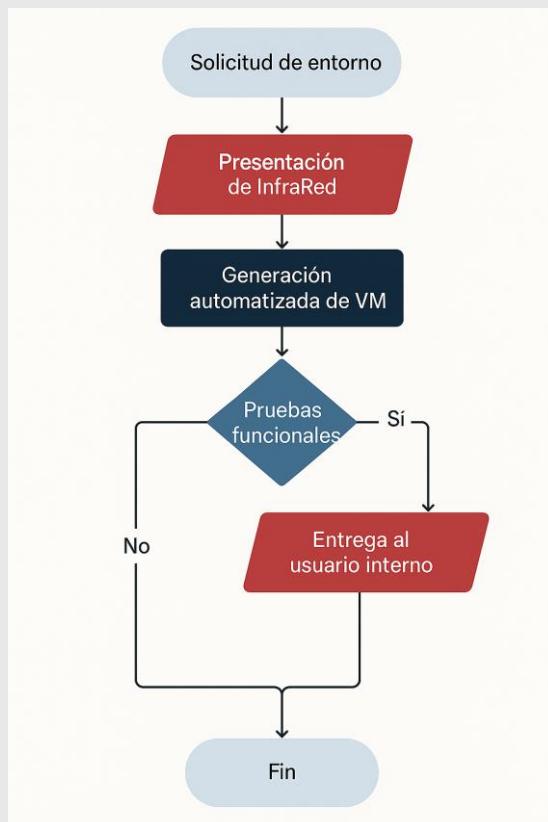
4. Validación funcional

Finalmente, arranco la máquina en vSphere, valido la conectividad, el acceso remoto y los servicios instalados, utilizando un cliente Windows 10 como estación de pruebas.

Este flujo puede repetirse tantas veces como sea necesario, adaptando únicamente los parámetros de entrada (como el nombre de la VM o el host de destino), lo que me permite escalar o clonar entornos de forma completamente automática.

1. Diagrama de flujo funcional del proceso de despliegue

Representa los pasos desde la solicitud del entorno hasta su entrega al usuario interno, incluyendo validaciones y testeо final



¿Por qué no se implementa una base de datos en el proyecto?

Aunque el proyecto InfraRed ha sido desarrollado siguiendo un enfoque profesional y estructurado, no se ha considerado necesaria la implementación de una base de datos real como parte del sistema funcional. Esta decisión responde a varias razones clave relacionadas con el alcance, el contexto y los objetivos del proyecto:

1. Naturaleza del proyecto

InfraRed es una herramienta pensada para el departamento de IT interno, orientada a la automatización del despliegue de entornos virtuales mediante Packer y Terraform. No se trata de una aplicación web, ni de un sistema de gestión de usuarios o recursos que requiera almacenamiento persistente de datos en tiempo real.

2. Infraestructura basada en código

InfraRed opera bajo el paradigma de Infraestructura como Código (IaC). Toda la información relevante (configuración de máquinas, scripts, plantillas, etc.) está almacenada como archivos de texto plano versionables, lo que elimina la necesidad de una base de datos estructurada para su funcionamiento.

3. Estructura de equipo estable y reducida

El equipo técnico que participa en el proyecto es reducido, fijo y no cambia dinámicamente. La información sobre roles, tareas y costes se documenta una única vez y no necesita ser consultada desde un sistema de datos relacional. Esto hace que el uso de una base de datos sea innecesario desde el punto de vista operativo.

4. No hay interacción externa ni cliente final

InfraRed está pensado para uso interno, sin requerimientos de gestión multiusuario, acceso concurrente o persistencia de acciones por parte de otros departamentos. Por lo tanto, no existe una justificación técnica o funcional que haga imprescindible una base de datos.

2. Diagrama lógico de red

El siguiente diagrama representa la arquitectura simplificada del entorno que he desplegado como parte del proyecto InfraRed. Este diseño refleja únicamente los elementos que he configurado y probado directamente:

Componentes implicados:

- **VMware Workstation Pro**

Es la plataforma que utilicé para crear y probar de forma local la imagen base de Ubuntu Server, utilizando la herramienta de automatización Packer. Desde aquí generé una máquina virtual preconfigurada que sirvió como base para el despliegue posterior.

- **Servidor ESXi**

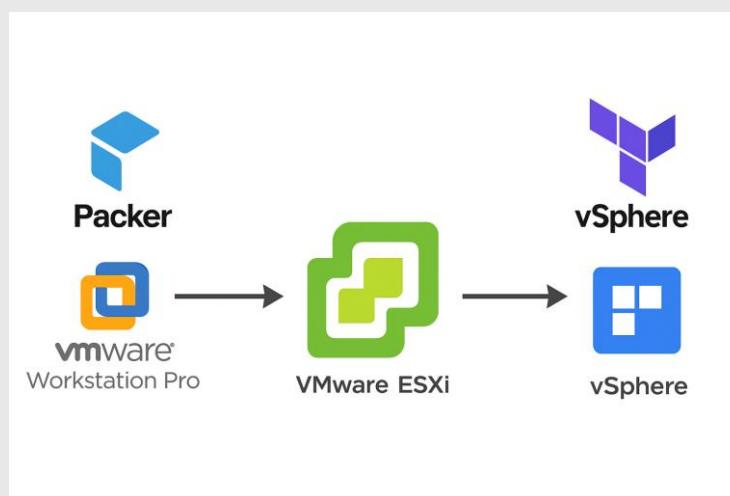
Una vez validada la imagen, exporté la máquina virtual desde Workstation y la subí al datastore de un host ESXi. Este paso fue necesario porque el entorno final donde debía funcionar Terraform era una infraestructura basada en vSphere.

- **vCenter (vSphere Web Client)**

Terraform se conecta directamente al servidor vCenter para gestionar el despliegue de la infraestructura. En este caso, he utilizado el archivo .vmdk convertido como disco base para crear una nueva máquina virtual dentro del entorno vSphere.

Conexión entre componentes

- La imagen creada en Workstation Pro es exportada de forma manual (previa conversión del disco a formato plano).
- Desde Terraform, me conecto a vCenter para desplegar una nueva VM usando ese disco.
- No se han configurado redes LAN internas, clientes Windows o routers virtuales como pfSense, ya que el objetivo principal de esta fase era demostrar la automatización del despliegue desde Packer hasta Terraform sobre vSphere.



3. Infraestructura desplegada

La infraestructura que he desplegado en este proyecto está compuesta por una serie de elementos que trabajan de forma coordinada para lograr un entorno automatizado, reproducible y gestionado por código. A continuación, detallo los componentes reales que forman parte del flujo completo del sistema:

- **Ubuntu Server**

Esta máquina virtual ha sido generada con Packer, incluyendo servicios esenciales como Apache y SSH activos desde el primer arranque. La imagen fue creada inicialmente en VMware Workstation Pro y, tras validarla, fue exportada al entorno ESXi para su despliegue final.

- **Packer y Terraform**

Son las dos herramientas clave del proyecto. Utilicé Packer para definir, construir y personalizar la imagen base de Ubuntu Server, y luego Terraform para automatizar el despliegue de una nueva máquina virtual en vSphere, a partir del disco .vmdk generado previamente.

- **VMware Workstation Pro**

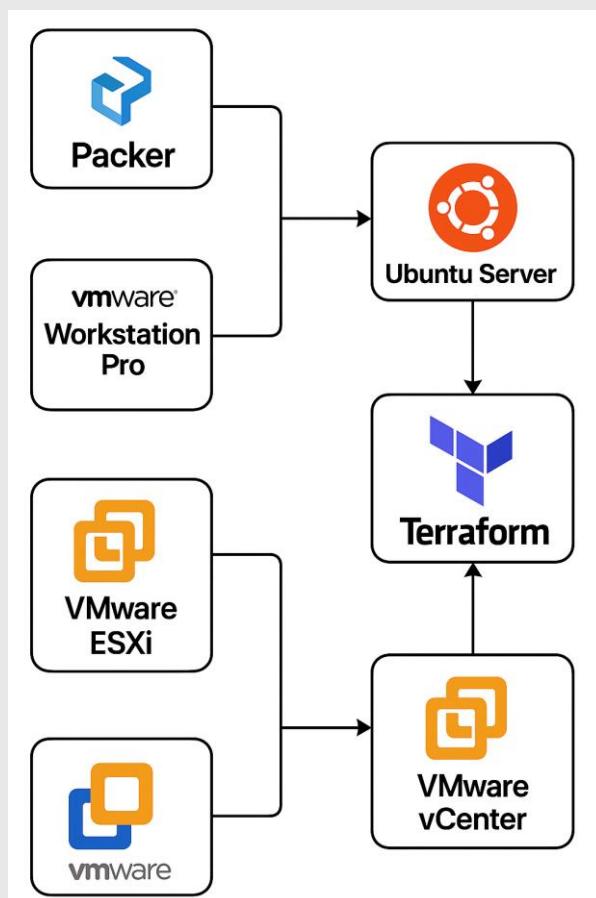
Sirvió como entorno local para el desarrollo y pruebas iniciales. Me permitió crear y ejecutar la imagen base de Ubuntu de forma controlada antes de exportarla a producción.

- **VMware ESXi**

Fue el servidor de virtualización donde subí el disco generado por Packer. Desde su datastore, Terraform pudo utilizar dicho archivo para desplegar una nueva VM dentro del entorno gestionado por vCenter.

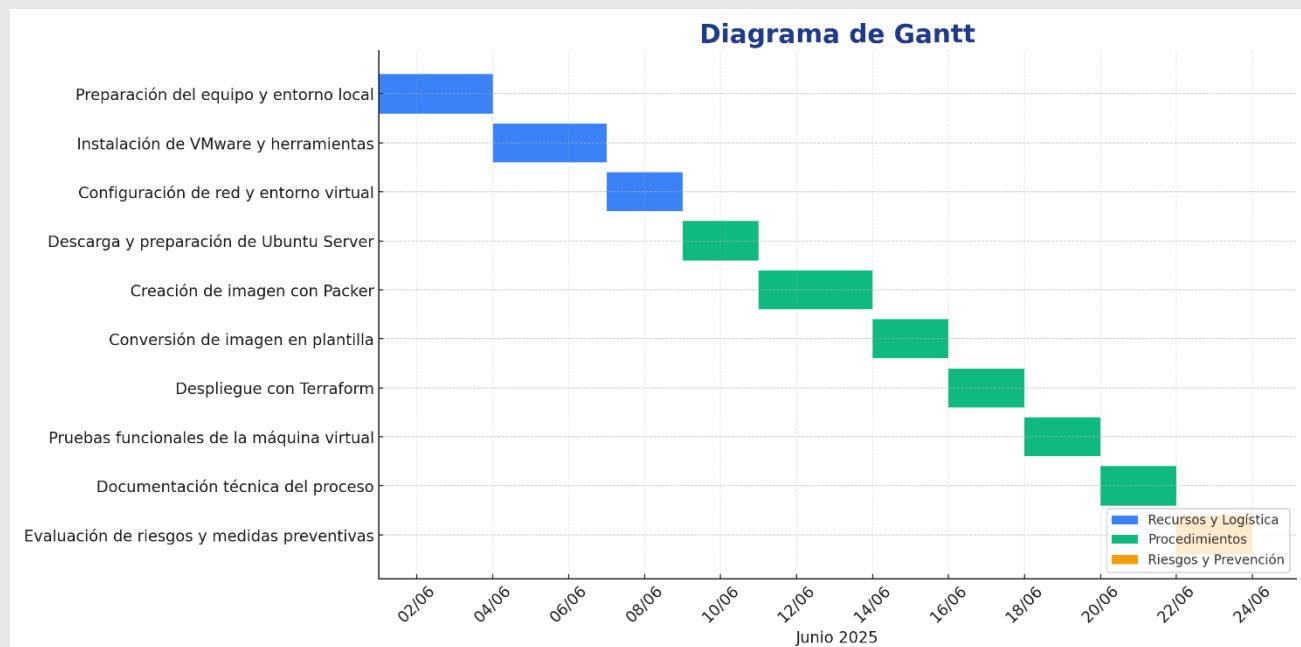
- **vCenter (vSphere Web Client)**

Es el punto de control central desde el cual Terraform se conecta para orquestar el despliegue. El vCenter gestiona los hosts ESXi y permite que Terraform interactúe con los recursos del entorno de forma segura y estructurada



PLANIFICACIÓN

Aunque este proyecto no parte de una empresa real, se plantea como si yo formara parte del departamento de sistemas de una consultora tecnológica. Por tanto, he realizado una planificación realista adaptada a mis recursos como estudiante, dividiendo el proyecto en tres bloques principales: recursos y logística, procedimientos e identificación de riesgos.



He estructurado la planificación en tres bloques principales: recursos y logística, procedimientos por actividad e identificación de riesgos y ajustes.

1. Definición de recursos y logística necesarios

Recursos utilizados:

- **Equipo personal:**

He trabajado con mi propio ordenador de sobremesa, con soporte de virtualización, 16 GB de RAM y más de 100 GB de almacenamiento libre, suficiente para imágenes de disco y máquinas virtuales.

- **Hipervisores:**

- **VMware Workstation Pro:** utilizado para crear y validar la imagen base de Ubuntu generada con Packer.
- **VMware ESXi:** entorno de destino donde exporté y desplegué la máquina virtual.
- **vCenter Server:** servidor de gestión centralizada al que se conecta Terraform para realizar el despliegue automatizado.

- **Software de automatización:**

- **Packer:** para crear imágenes de máquinas virtuales personalizadas.
- **Terraform:** para desplegar la infraestructura de forma declarativa y automatizada.

- **Sistema operativo base:**

Ubuntu Server 24.04 LTS, descargado en ISO desde la web oficial de Canonical.

- **Conexión a Internet:**

Necesaria para descargar software, actualizaciones, gestionar repositorios y acceder a la interfaz de administración del vCenter.

Logística aplicada:

- **Configuración del entorno:**

Realicé una configuración escalonada del entorno local (VMware Workstation), la conexión al entorno ESXi y la configuración del provider de Terraform apuntando al vCenter.

- **Gestión de versiones y organización:**

Organicé el proyecto en carpetas por fases: packer/, terraform/, imagenes/, etc. Además, hice uso de nombres consistentes en las imágenes y archivos .vmdk.

- **Respaldo:**

Guardé copias locales de los archivos .vmx, .vmdk y configuraciones .tf en cada hito para asegurarme de poder volver a estados anteriores si surgía un error.

2. Procedimientos por actividad

Fase 1 – Preparación del entorno base

- Instalé VMware Workstation Pro.
- Instalé Packer y Terraform.
- Descargué y verifiqué la ISO de Ubuntu Server 24.04 LTS.
- Validé la compatibilidad de hardware y rutas locales.

Fase 2 – Generación de imagen con Packer

- Definí el archivo ubuntu.pkr.hcl con el proveedor vmware-iso.
- Añadí scripts de aprovisionamiento para instalar Apache y habilitar SSH.
- Construí la máquina con Packer y validé el funcionamiento en local.

Fase 3 – Exportación al entorno ESXi

- Convertí el disco .vmdk a formato plano consolidado.
- Subí manualmente el archivo al datastore SVR1_OS en ESXi.
- Validé la integridad del archivo y su compatibilidad con vSphere.

Fase 4 – Despliegue con Terraform

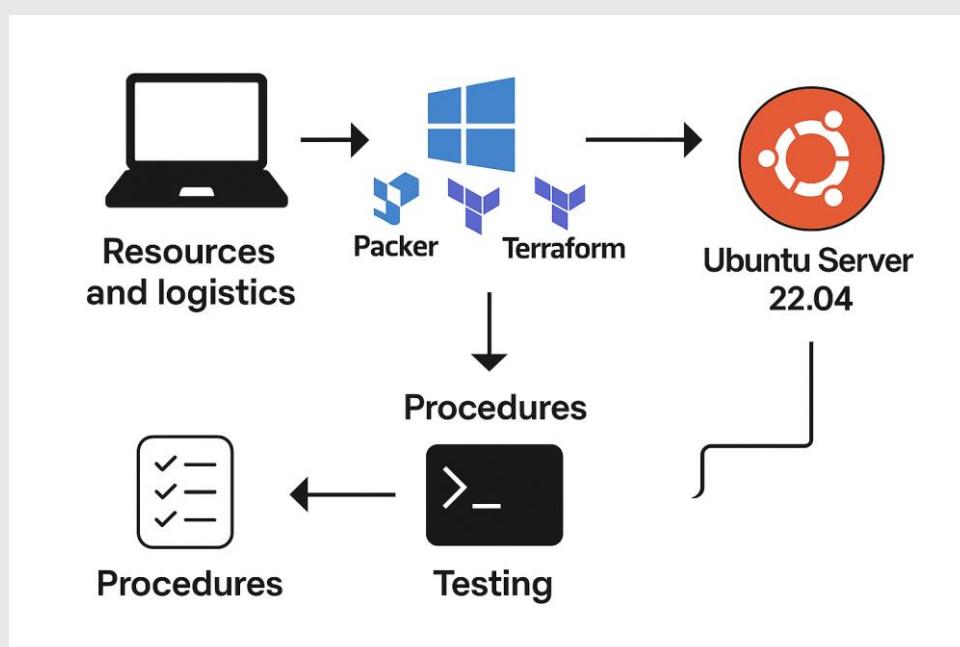
- Definí los archivos main.tf, variables.tf y terraform.tfvars.
- Configuré el provider de Terraform para conectarse al vCenter.
- Declaré la máquina virtual, asociando el disco .vmdk, CPU, RAM y red.
- Ejecuté terraform init, terraform plan y finalmente terraform apply.

Fase 5 – Verificación funcional

- Verifiqué desde vSphere que la VM aparecía correctamente registrada.
- Validé que arrancaba correctamente, que los servicios (Apache/SSH) funcionaban y que respondía por red.
- Probé la conectividad desde otras máquinas del entorno.

Fase 6 – Documentación del proyecto

- Redacté y organicé todo el proceso paso a paso.
- Incluí capturas de Packer, Terraform, vSphere y el diagrama de red.
- Detallé errores encontrados y las soluciones aplicadas en cada fase.



3. Identificación de riesgos y plan de prevención

Riesgo	Plan de prevención / acción correctiva
Fallos durante la instalación de ESXi o VMware	Probar previamente en entorno local con snapshots de seguridad.
Incompatibilidad del hardware con virtualización	Verificar que el equipo tiene habilitado VT-x/AMD-V en BIOS y soporte suficiente (RAM, CPU).
Configuraciones de red incorrectas	Usar configuraciones estándar (NAT o bridge), realizar pruebas previas con una VM base.
Falta de recursos para virtualizar	Ajustar el número de máquinas, limitar el uso de RAM y espacio si el equipo no lo soporta.
Fallo en los scripts de Packer o Terraform	Validar cada archivo por fases, usar packer validate y terraform plan antes de ejecutar el despliegue.
Pérdida de archivos o configuración	Hacer copias de seguridad del proyecto y snapshots antes de cada modificación importante.

¿Para qué uso los snapshots?

Yo utilizo los snapshots principalmente para hacer pruebas sin riesgo. Me permiten aplicar cambios, instalar servicios o ejecutar scripts sabiendo que, si algo sale mal, puedo volver al estado anterior en cuestión de segundos.

También los uso como puntos de control antes de hacer modificaciones importantes en las máquinas virtuales, lo que me ayuda a evitar perder tiempo solucionando errores o volviendo a configurar todo desde cero.

Por ejemplo, mientras trabajaba en el proyecto InfraRed, antes de ejecutar un script que instalaba varios servicios en la imagen base, creé un snapshot. De esta forma, si algo fallaba o se rompía la configuración de red, podía restaurarlo fácilmente y continuar sin complicaciones.

IMPLEMENTACIÓN, PRUEBAS Y PUESTA EN MARCHA

Tras la fase de diseño y planificación, pasé a la etapa más técnica del proyecto: la implementación. En esta fase, puse en práctica toda la infraestructura conceptualizada para automatizar el despliegue de entornos virtuales, combinando herramientas de infraestructura como código (IaC) con plataformas de virtualización.

Mi objetivo era lograr un flujo completamente automatizado, desde la generación de una imagen base con Packer hasta el despliegue de esa imagen como máquina virtual funcional a través de Terraform. Para ello, adapté el proceso a mis recursos disponibles, validando qué era factible ejecutar en local y qué requería integración con un entorno más avanzado como vSphere.

Durante esta implementación, me encontré con varios desafíos técnicos que me obligaron a reevaluar decisiones y modificar la arquitectura inicial. No obstante, el resultado fue una infraestructura funcional, escalable y totalmente automatizada, desplegada con éxito en un entorno empresarial simulado.

A continuación, detallo paso a paso todo el proceso de implementación, incluyendo las herramientas utilizadas, configuraciones aplicadas y pruebas realizadas.



Instalación de VMware Workstation Pro

El primer paso para comenzar la implementación técnica del proyecto InfraRed fue preparar el entorno de virtualización. Para ello, instalé VMware Workstation Pro, la plataforma sobre la que se ejecutarán las máquinas virtuales generadas con Packer y desplegadas con Terraform.

¿Por qué VMware Workstation Pro?

Elegí esta herramienta por las siguientes razones:

- Es compatible con los scripts de automatización que utilizaré.
- Permite gestionar redes virtuales, snapshots y plantillas fácilmente.
- Se puede utilizar en un entorno local sin necesidad de conexión a Internet o infraestructura cloud.
- Es un estándar profesional muy utilizado en entornos reales.

Proceso de instalación

1. Descarga del instalador

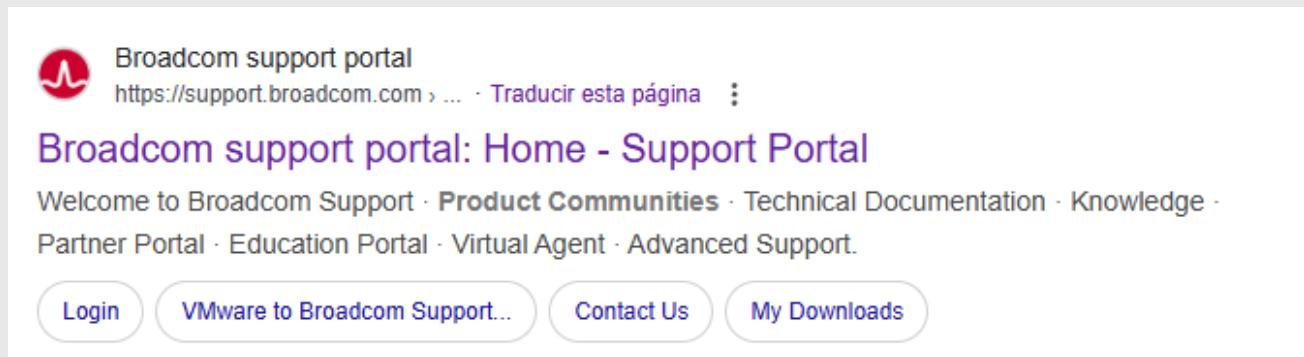
Accedí al sitio oficial de [VMware](#) y descargué la versión más reciente de VMware Workstation Pro 17.x para Windows.



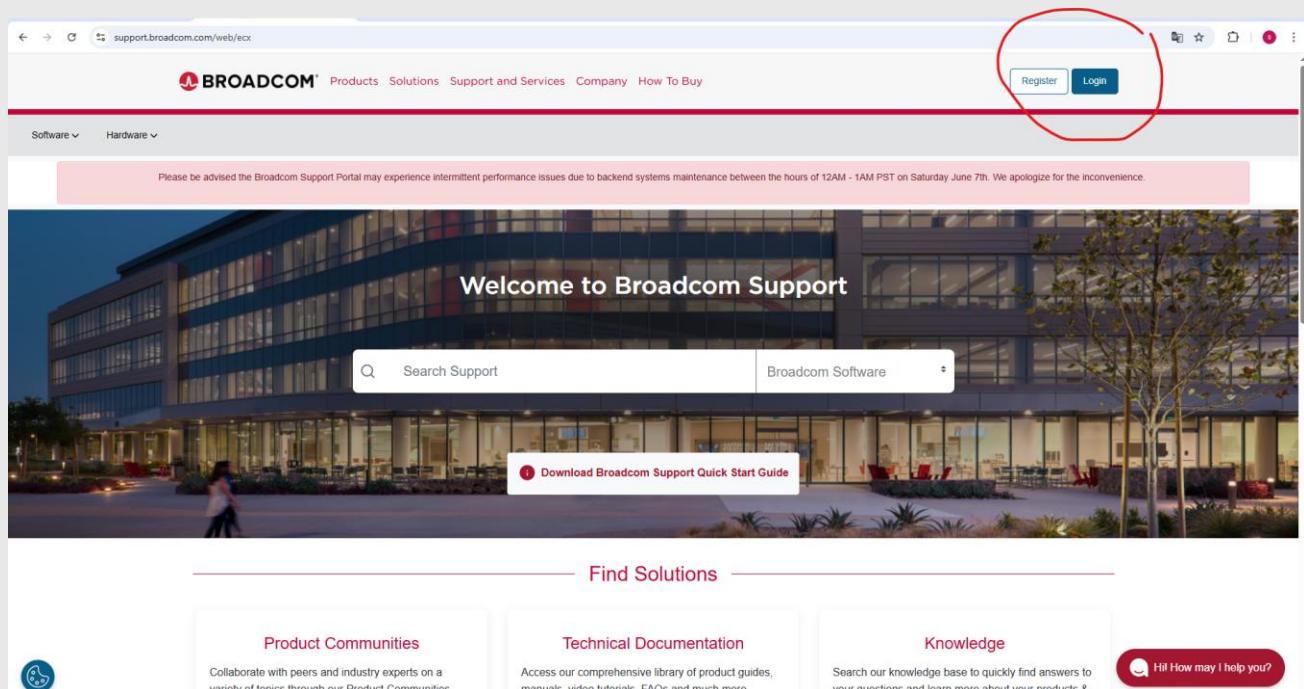
Instalación de VMware ESXi

Como parte del proyecto InfraRed, también he querido probar el despliegue de entornos virtuales sobre un hipervisor más cercano al entorno empresarial. Para ello, opté por instalar VMware ESXi en su versión gratuita, ahora distribuida por Broadcom tras la adquisición de VMware.

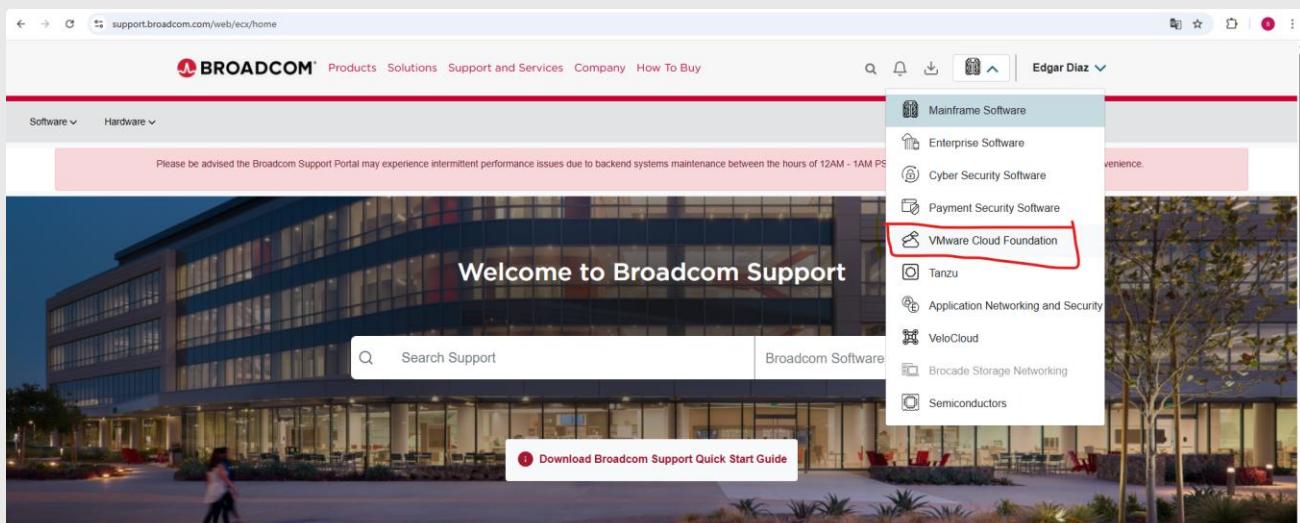
Descarga de VMware ESXi Free (Broadcom). Accedí al portal oficial de VMware ahora gestionado por Broadcom:



Me registré o inicié sesión con una cuenta personal



Busqué Vmware Cloud Foundation en el desplegable y me dirige a descargas.



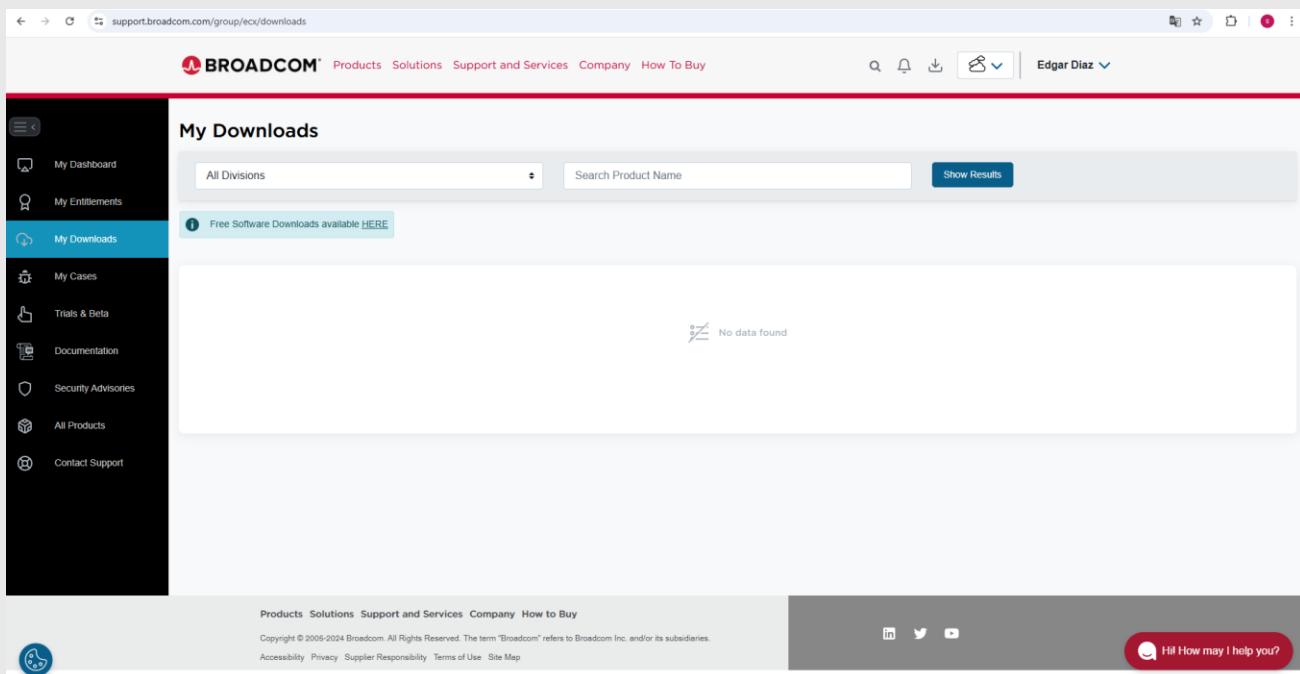
The screenshot shows the Broadcom Support Portal homepage. At the top, there's a navigation bar with links for Products, Solutions, Support and Services, Company, and How To Buy. Below the navigation is a dropdown menu for 'Software' and 'Hardware'. A message at the top of the page says: 'Please be advised the Broadcom Support Portal may experience intermittent performance issues due to backend systems maintenance between the hours of 12AM - 1AM PST.' The main content area features a large image of a modern building at night with the text 'Welcome to Broadcom Support'. Below the image is a search bar labeled 'Search Support' and a button to 'Download Broadcom Support Quick Start Guide'. On the right side, there's a sidebar with a list of product categories. One item, 'VMware Cloud Foundation', is highlighted with a red box.

Find Solutions



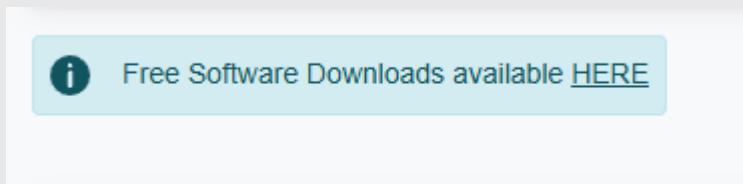
This screenshot shows the 'Find Solutions' section of the Broadcom Support Portal. It includes three main boxes: 'Product Communities' (Collaborate with peers and industry experts on a variety of topics through our Product Communities), 'Technical Documentation' (Access our comprehensive library of product guides, manuals, video tutorials, FAQs and much more), and 'Knowledge' (Search our knowledge base to quickly find answers to your questions and learn more about your products & services). A red button labeled 'Hi! How may I help you?' is also visible.

Cómo mi cuenta no tiene ninguna licencia asignada aún, no tendré descargas disponibles.



This screenshot shows the 'My Downloads' page of the Broadcom Support Portal. The left sidebar has a navigation menu with options like My Dashboard, My Entitlements, My Downloads (which is selected and highlighted in blue), My Cases, Trials & Beta, Documentation, Security Advisories, All Products, and Contact Support. The main content area is titled 'My Downloads' and shows a search bar for 'All Divisions' and 'Search Product Name'. A message says 'Free Software Downloads available HERE'. Below it, a note says 'No data found'. At the bottom, there's a footer with links for Products, Solutions, Support and Services, Company, How To Buy, and social media icons for LinkedIn, Twitter, and YouTube. A red 'Hi! How may I help you?' button is also present.

Pero me sale de emergencia (Free Software Downloads available):



A screenshot of a message box with a light blue background. It contains a circular icon with an exclamation mark and the text 'Free Software Downloads available [HERE](#)'.

Ahora busco VMware vSphere Hypervisor:

Free Downloads

All Divisions

Java Buildpack for VMware Tanzu	Java Native Image Buildpack for VMware Tanzu
Spring Cloud Gateway for VMware Tanzu	VMware Avi Load Balancer
VMware Cloud Director App Launchpad	VMware Cloud Director Availability
VMware Cloud Director Extension for VMware Data Solutions	VMware Cloud Director Object Storage Extension
VMware Cloud Gateway for VMC HLM	VMware Cloud Provider Lifecycle Manager
VMware Fusion	VMware Live Recovery
VMware Tanzu CLI	VMware Tools
VMware Validated Design for Software-Defined Data Center	VMware vCenter converter
VMware vCloud Usage Meter	VMware vSphere ESX Drivers
VMware vSphere Hypervisor	VMware Workstation Pro

A continuación, me sale la ISO y la versión:

VMware vSphere Hypervisor

Release [8.0U3e](#)

1 to 1 of 1 records

← **VMware vSphere Hypervisor 8.0U3e**

I agree to the [Terms and Conditions](#) Expand All

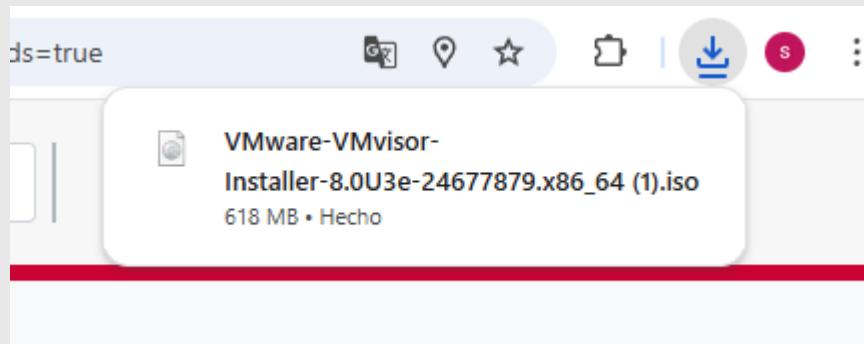
Primary Downloads

Search 8.0U3e English

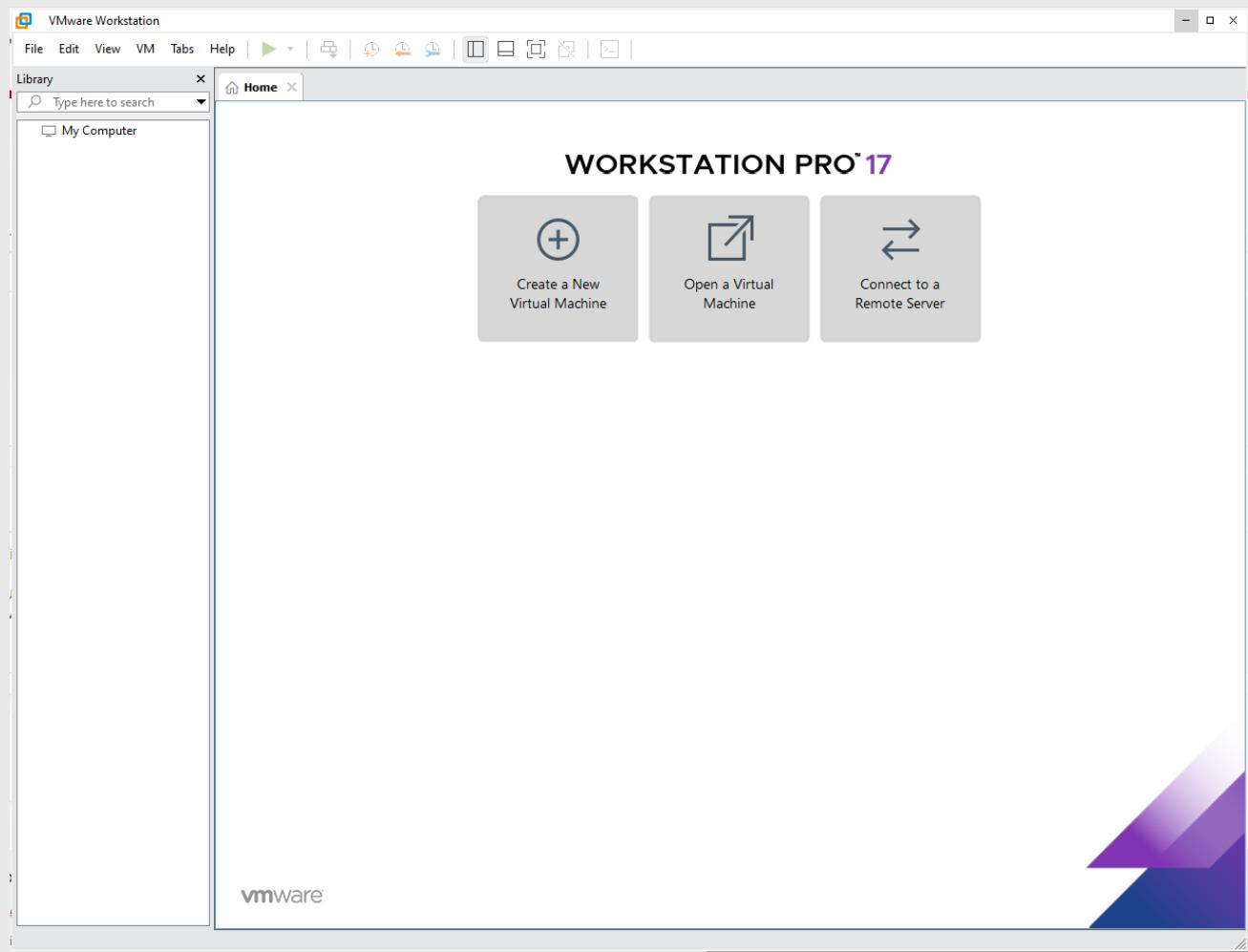
File Name	Release Date	Last Updated	SHA2	MD5	<input type="button" value="Download"/>	<input type="button" value="File"/>
VMware vSphere Hypervisor (ESXi ISO) image VMware-VNvisor-Installer-8.0U3e-24677879.x86_64.iso(618.34 MB) Build Number: 24677879	Apr 10, 2025	Apr 07, 2025	9782c96ffd01cc56da17ec31573da69f4cba2f9402e67c8b55d05d94 72c7376a	ddb642eee9635ceaeaeb812ccd813189	<input type="button" value="Download"/>	<input type="button" value="File"/>

1 to 1 of 1 records

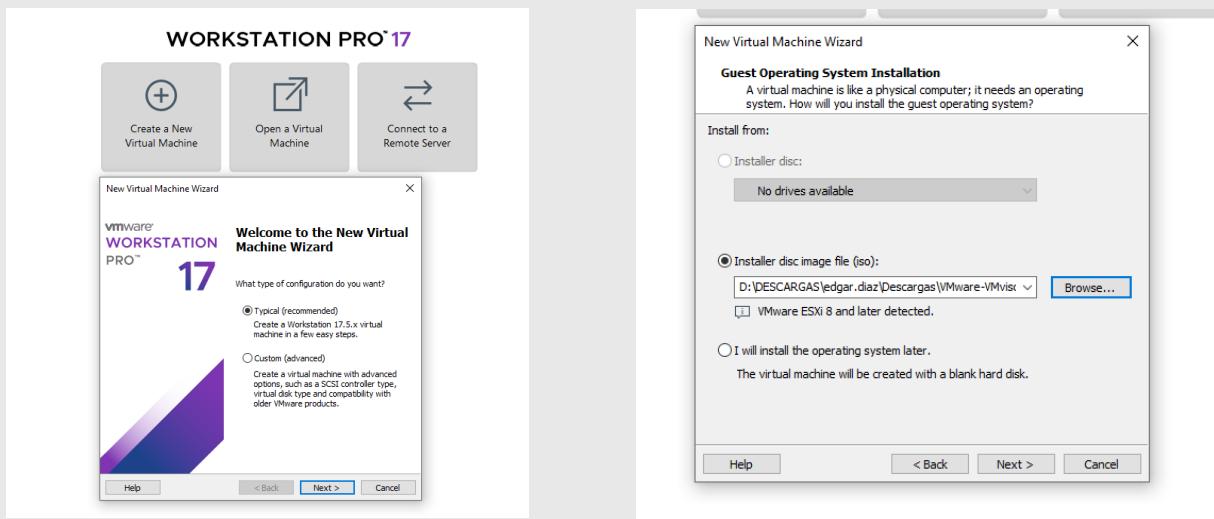
Descargo:



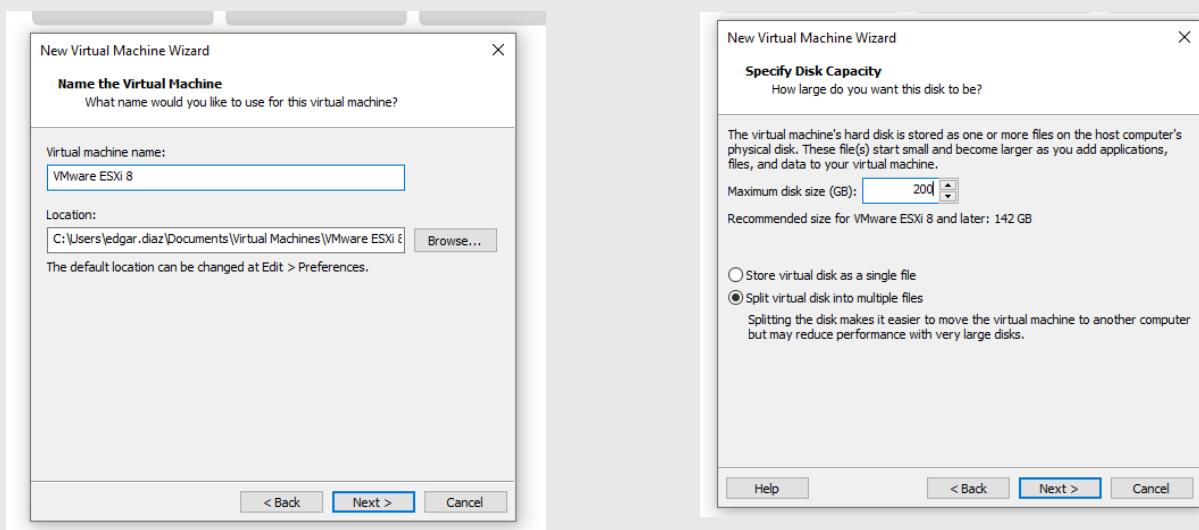
Ahora comenzaremos la instalación dentro del Workstation Pro.



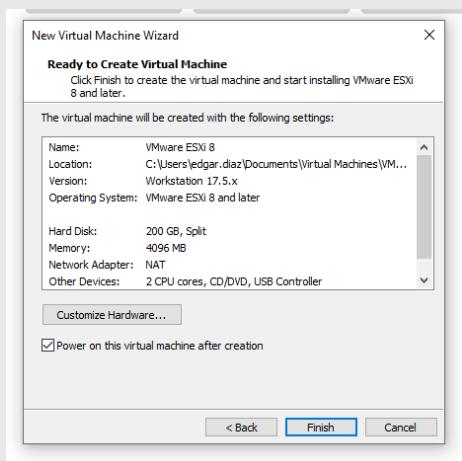
Creo una nueva máquina virtual y selecciono por defecto Typical. Por otro lado seleccionamos la imagen ISO que acabo de descargar:



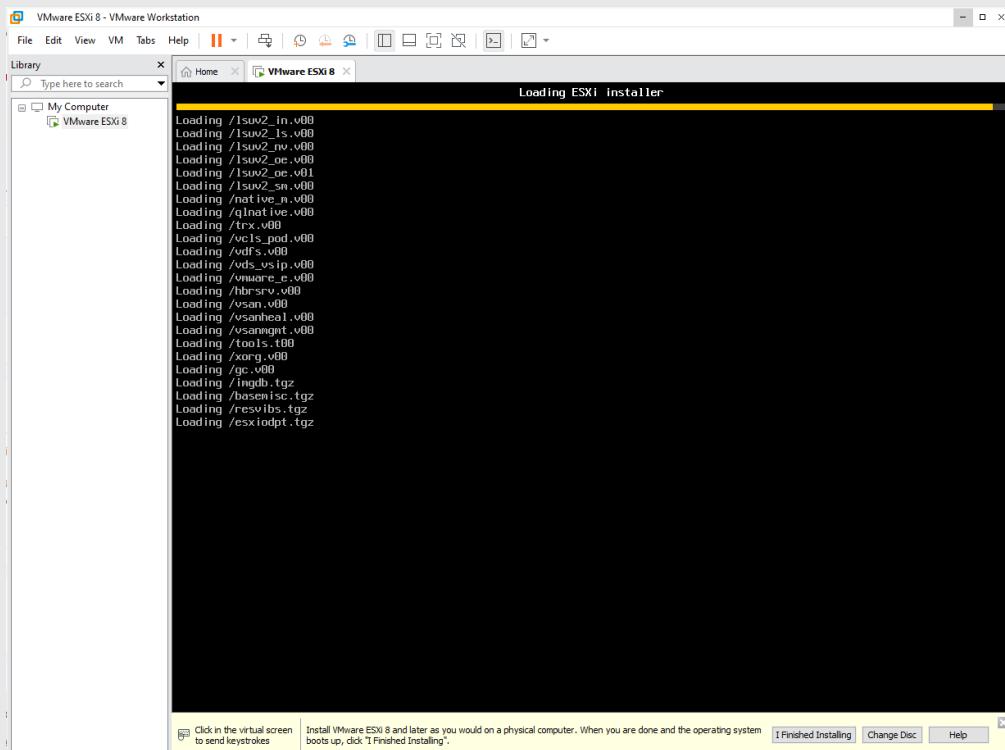
Me reconoce la versión de ESXI 8 y agrego un disco de 200 GB:



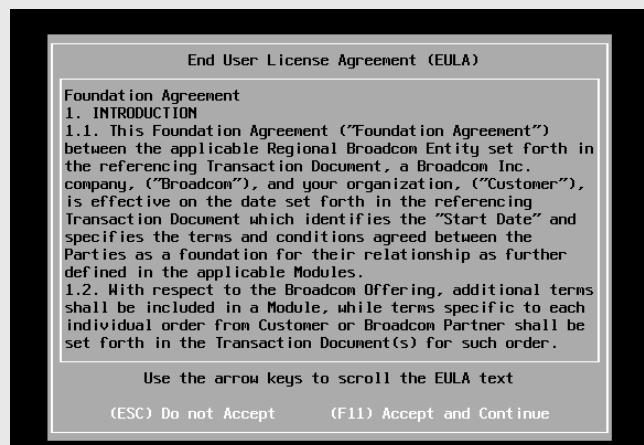
Por último, comienza la instalación automática:



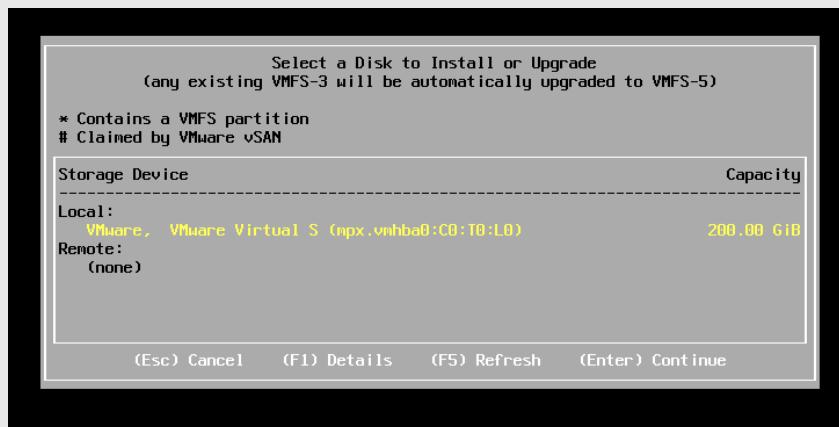
En este momento pasamos a la última fase de la instalación.



Le doy a continuar y acepto el acuerdo:



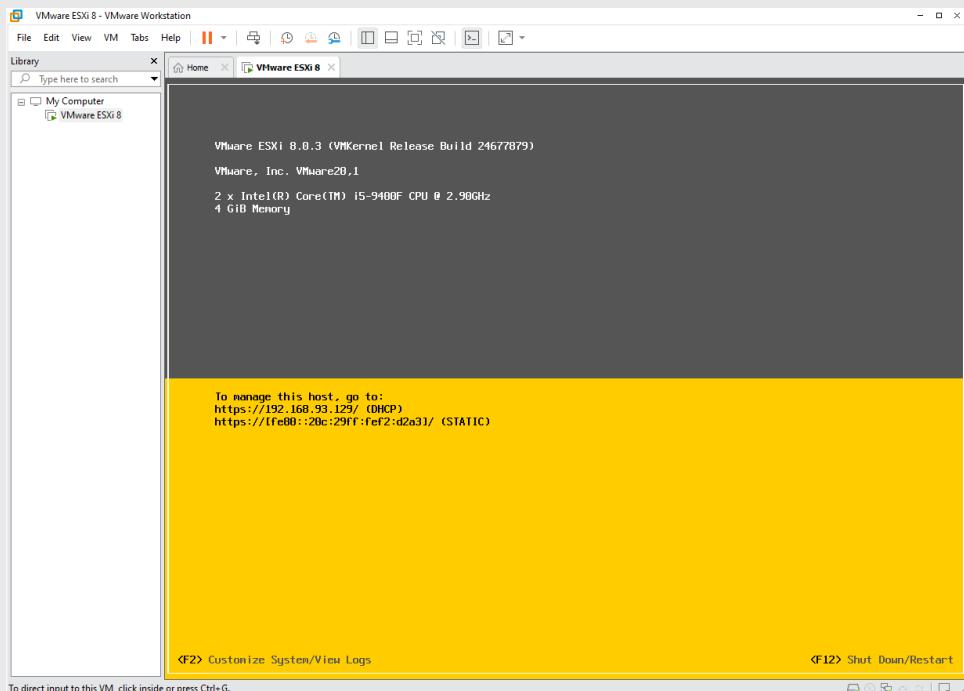
Utilizo este disco virtual:



Selecciono US Default para el teclado y agrego una contraseña:

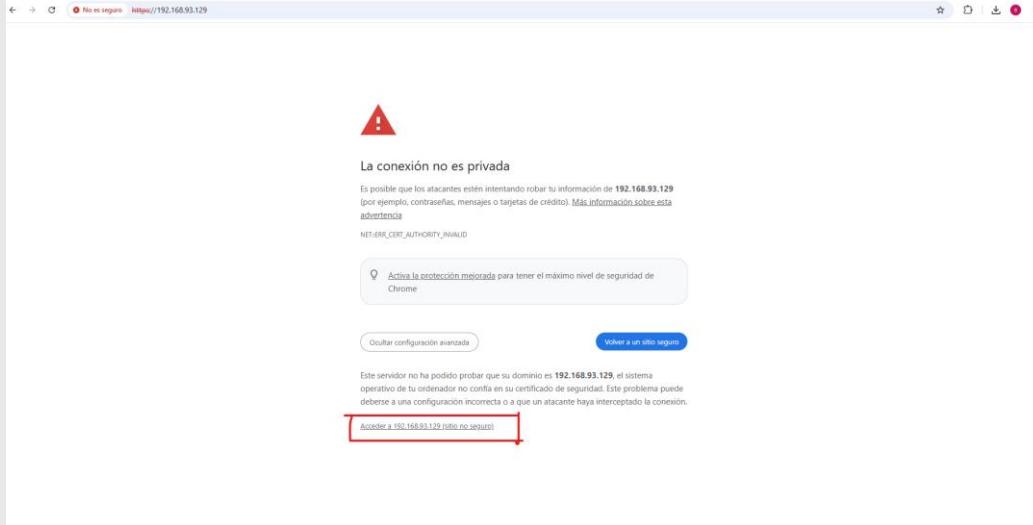
Nos avisa de que el disco será borrado y reiniciamos:

Para finalizar deberemos conectarnos a la máquina virtual por IP. ESXi muestra en pantalla su IP local asignada.

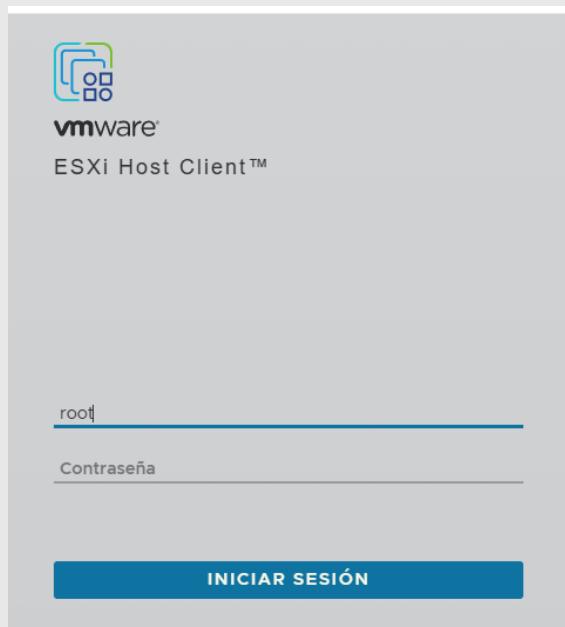


To direct input to this VM, click inside or press Ctrl+G.

Desde un navegador en mi máquina local accedí a:



Ingresé con usuario: root y la contraseña definida.



Desde la interfaz web de ESXi, ya pude:

- Crear máquinas virtuales.
- Configurar redes.
- Subir ISOs y plantillas.

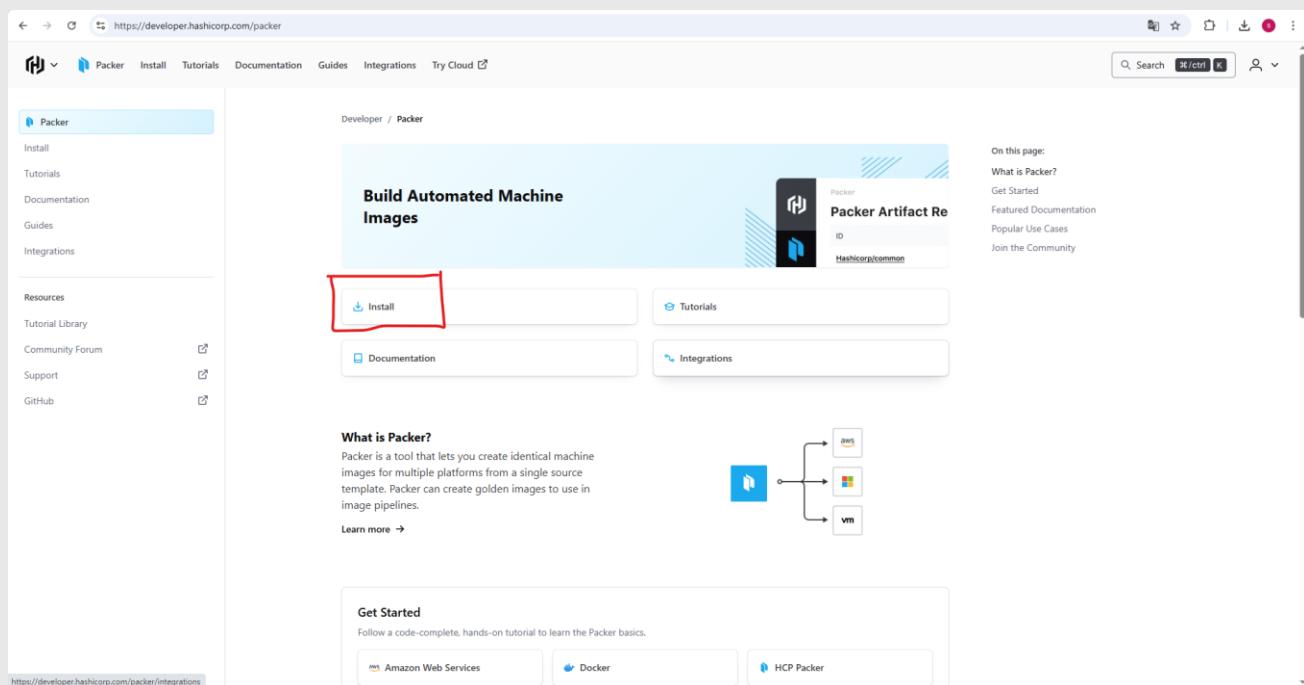
Tarea	Destino	Iniciador	En cola	Iniciado	Resultado	Complejado
Update Options	localhost.localdomain	root	02/06/2025 08:49:54	02/06/2025 08:49:54	Se completó correctamente.	02/06/2025 08:49:54
Auto Start Power On	localhost.localdomain	root	02/06/2025 08:47:22	02/06/2025 08:47:22	Se completó correctamente.	02/06/2025 08:47:22

Instalación de Packer

La instalación de Packer es un proceso sencillo, ya que se distribuye como un binario independiente que no requiere instalación formal, solo configuración en el sistema. A continuación, detallo los pasos que seguí para dejar Packer operativo en mi entorno de desarrollo con Windows 11:

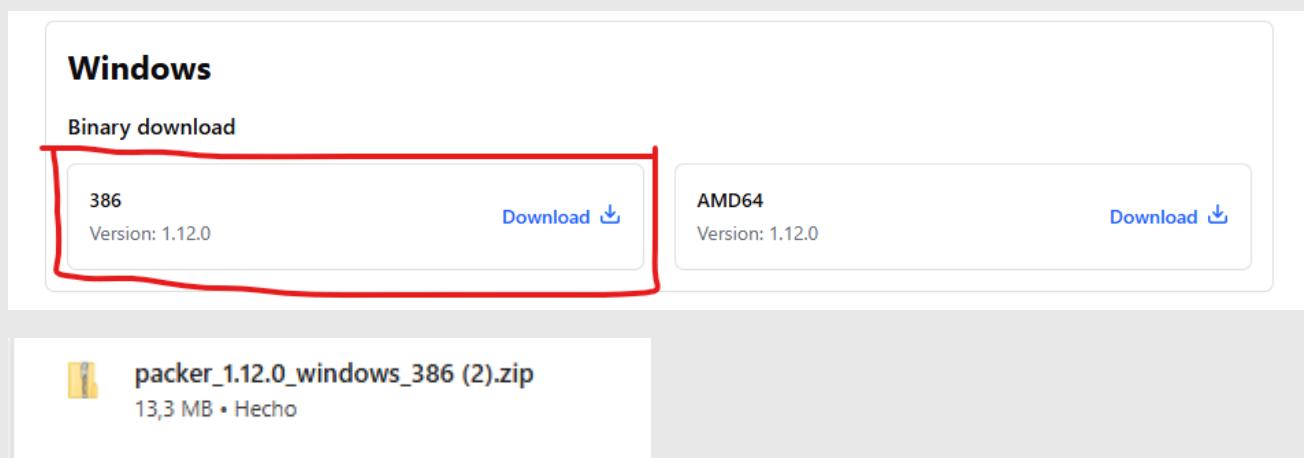
Accedí al sitio oficial de descargas de HashiCorp:

<https://developer.hashicorp.com/packer>



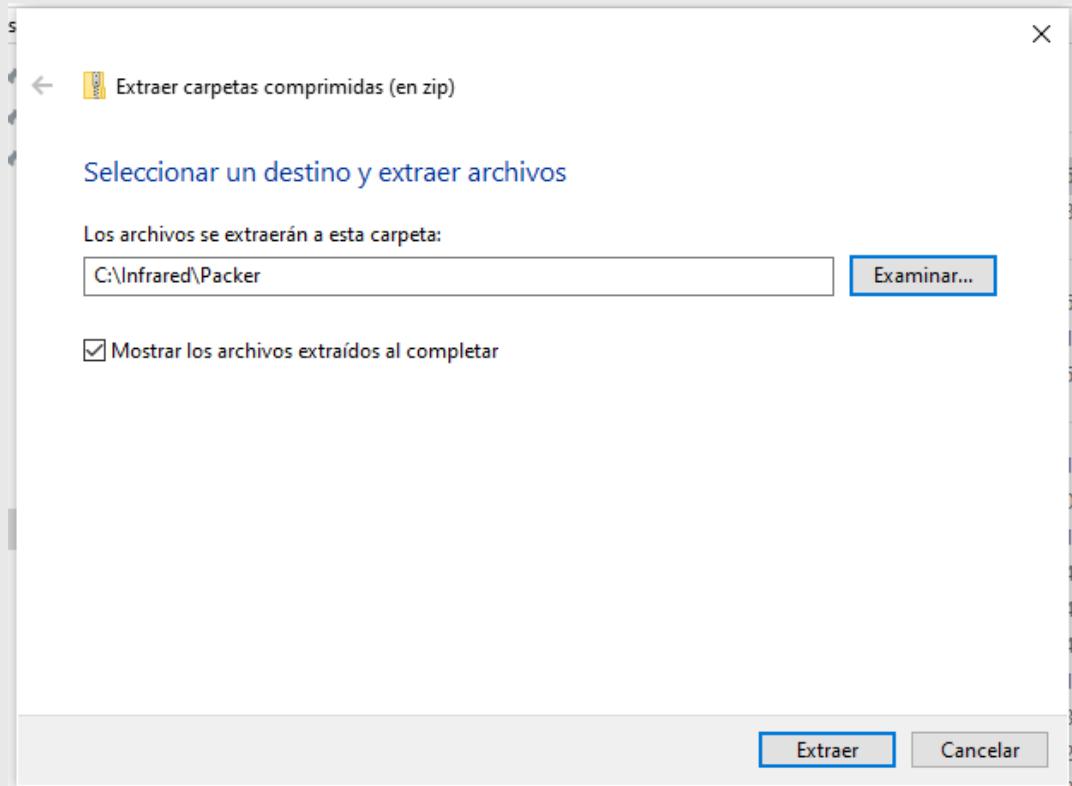
The screenshot shows the HashiCorp Packer download page. On the left, there's a sidebar with links for Packer, Install, Tutorials, Documentation, Guides, Integrations, Resources, Tutorial Library, Community Forum, Support, and GitHub. The main content area has a header 'Developer / Packer'. Below it, a large box says 'Build Automated Machine Images' with a 'Packer' logo and a 'Packer Artifact Registry' link. It features four buttons: 'Install' (highlighted with a red box), 'Tutorials', 'Documentation', and 'Integrations'. To the right, there's a sidebar titled 'On this page:' with links to 'What is Packer?', 'Get Started', 'Featured Documentation', 'Popular Use Cases', and 'Join the Community'. Below the sidebar, there's a diagram showing a central blue icon connected to three boxes: 'aws', 'vm', and 'Docker'. At the bottom, there's a 'Get Started' section with a 'Learn more →' link and three buttons for 'Amazon Web Services', 'Docker', and 'HCP Packer'.

Seleccioné la versión para Windows 386 versión 1.12.0 y descargué el archivo .zip.

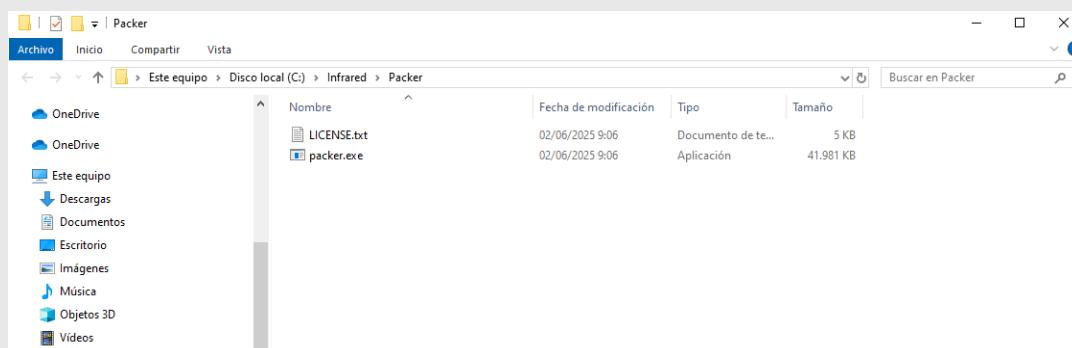


The screenshot shows the 'Windows' download page for Packer 1.12.0. It features two download options: '386 Version: 1.12.0' and 'AMD64 Version: 1.12.0'. Both options have a 'Download' button. Below the page, a file named 'packer_1.12.0_windows_386 (2).zip' is shown with a size of 13,3 MB and a status of 'Hecho'.

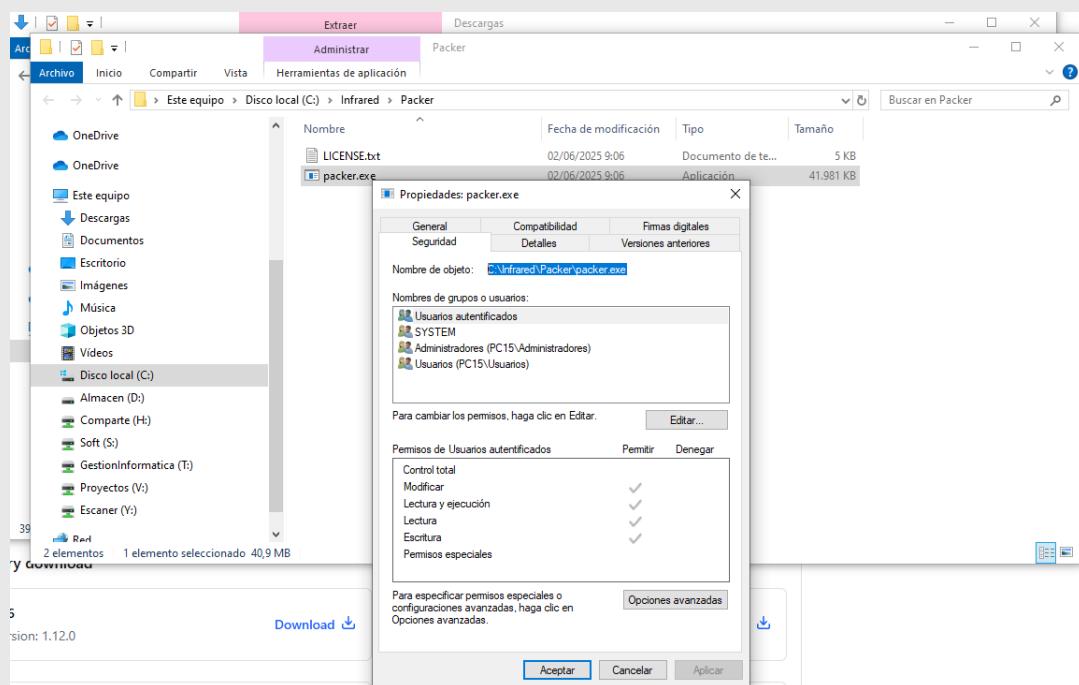
En este momento extraigo todo su contenido:



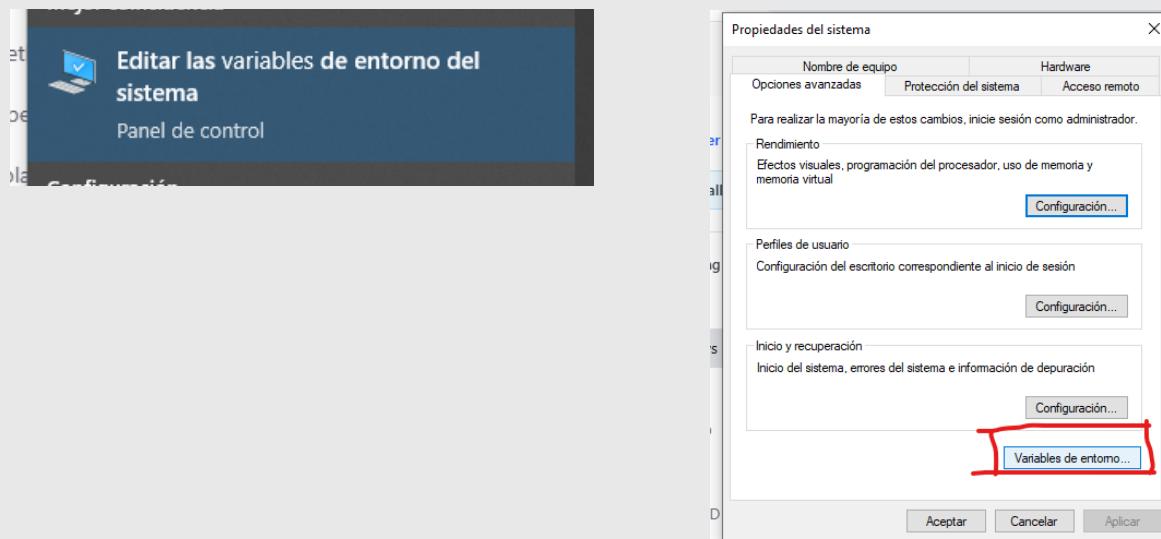
Lo abrimos y nos encontramos con el fichero packer.exe



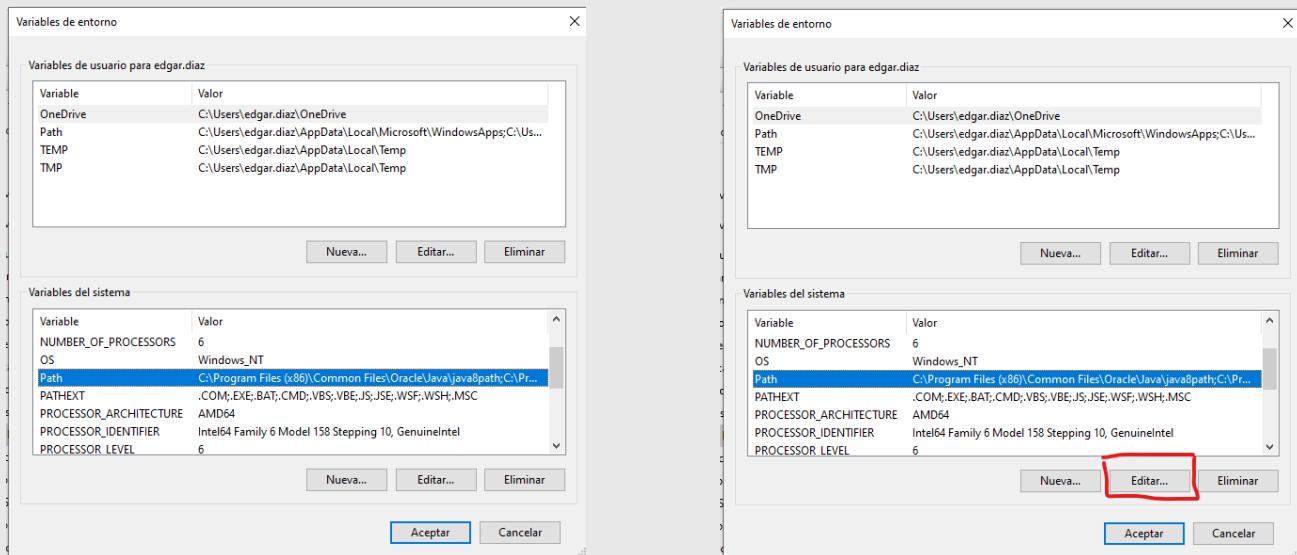
Copiamos la dirección en la cual lo hemos extraído:



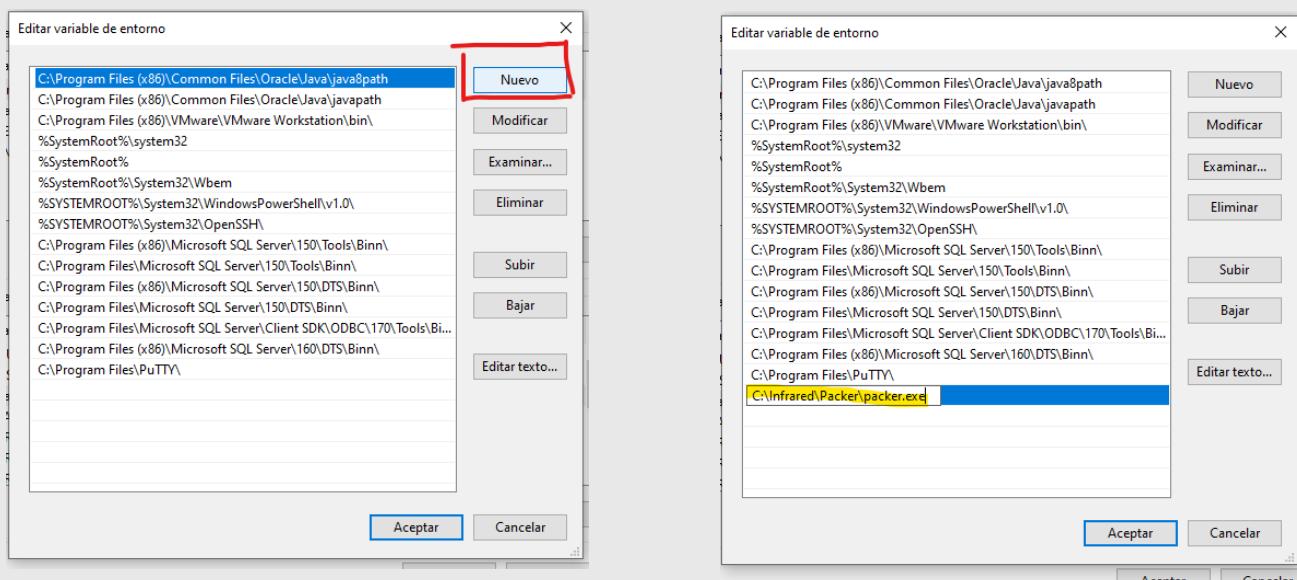
Y nos dirigimos al editor de variables de entorno dónde en opciones avanzadas seleccionamos variables de entorno:



Ahora en las variables del sistema buscamos en la que pone Path y editamos:



Selecciono nuevo y pegamos la dirección en la que tengo el packer.exe



Por último, compruebo que realmente lo he instalado.

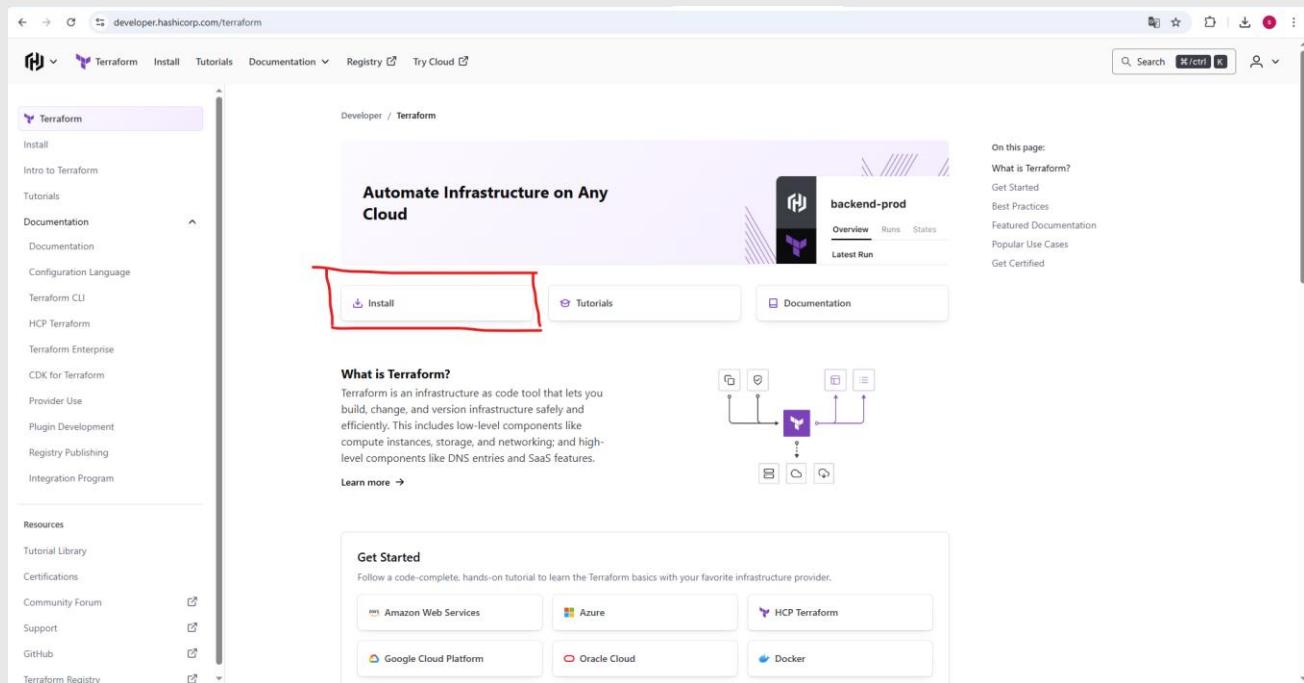
```
C:\Windows\system32>packer --version
Packer v1.12.0
```

Instalación de Terraform

Terraform, al igual que Packer, se distribuye como un ejecutable autónomo que no requiere instalación tradicional. Su configuración en Windows es rápida y se basa en colocar el binario en una ubicación accesible desde el sistema.

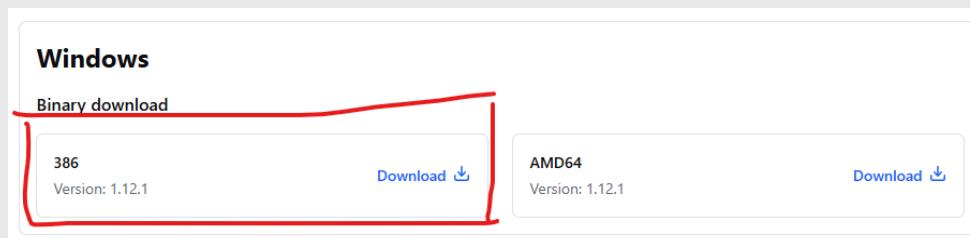
Accedí a la página oficial:

<https://developer.hashicorp.com/terraform>



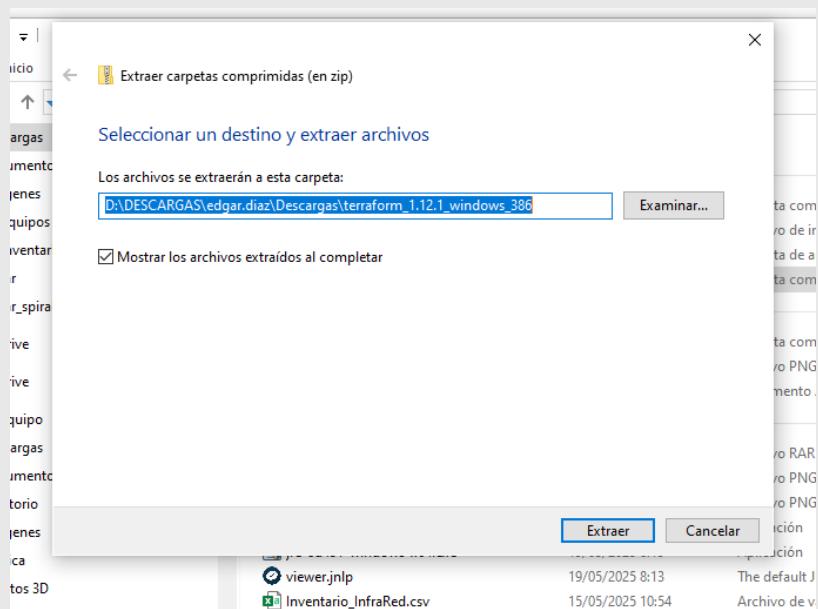
The screenshot shows the HashiCorp Terraform developer page. On the left, there's a sidebar with navigation links like 'Install', 'Tutorials', 'Documentation', and 'Resources'. The main content area has a heading 'Automate Infrastructure on Any Cloud' and a large 'Install' button which is highlighted with a red box. Below it are 'Tutorials' and 'Documentation' buttons. To the right, there's a sidebar titled 'On this page:' with links to 'What is Terraform?', 'Get Started', 'Best Practices', 'Featured Documentation', 'Popular Use Cases', and 'Get Certified'. There's also a 'backend-prod' section with 'Overview', 'Runs', 'States', and 'Latest Run' tabs.

Seleccioné la versión para Windows 386 versión 1.12.0 y descargué el archivo .zip

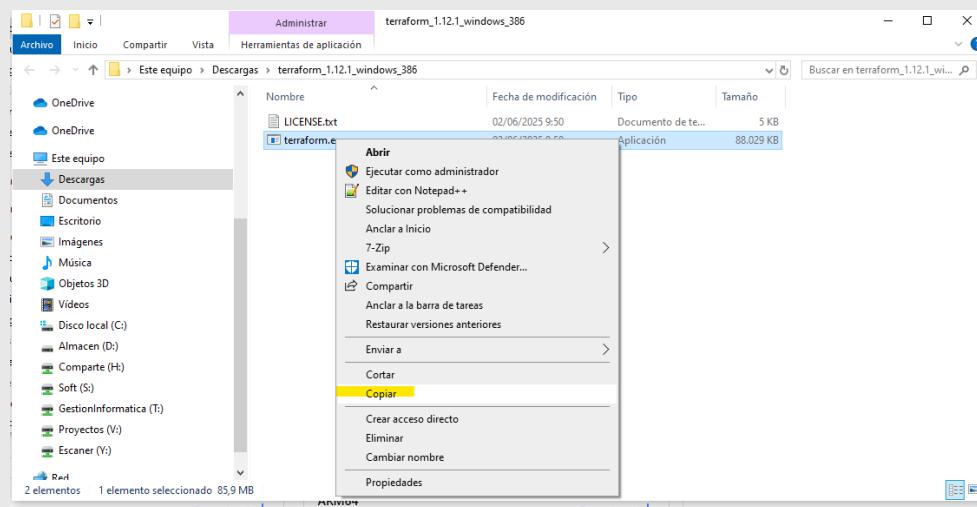


The screenshot shows the Terraform binary download page for Windows. It features a 'Windows' section with two download options: '386 Version: 1.12.1' and 'AMD64 Version: 1.12.1'. The '386' link is highlighted with a red box. Both links have a 'Download' button next to them.

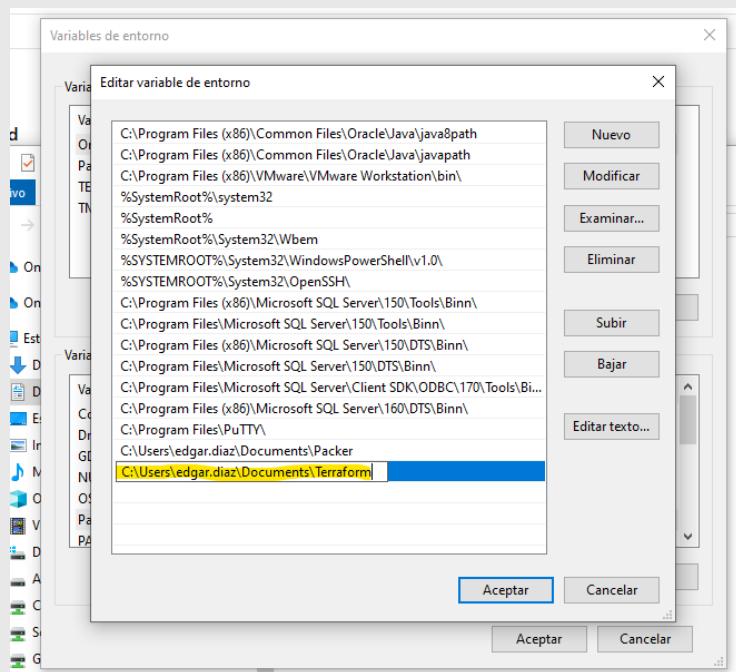
Extrae el contenido en:



Copíe el fichero terraform.exe y lo pégue en la carpeta correspondiente:



Añadí esa carpeta a la variable de entorno PATH igual que con Packer:



Verifiqué la instalación desde el terminal:

```
C:\ Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5854]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>terraform --version
Terraform v1.12.1
on windows_386
```

Creación de imagen base con Packer en VMware Workstation

Una vez configurado el entorno local de virtualización con VMware Workstation y teniendo disponible la ISO de Ubuntu Server, procedí a generar una imagen preconfigurada utilizando Packer. Este paso fue fundamental para estandarizar el entorno y ahorrar tiempo en futuros despliegues automáticos.

A continuación, detallo todo el proceso seguido:

Preparación del entorno

1. Descargué la ISO oficial de Ubuntu Server 24.04.1 desde la web de Ubuntu.

<https://ubuntu.com/download/server/thank-you?version=24.04.2&architecture=amd64&its=true>

2. Moví la ISO a una ruta local accesible, en mi caso:

D:/DESCARGAS/edgar.diaz/Descargas/ubuntu-24.04.1-live-server-amd64.iso

3. Verifiqué el checksum SHA256 de la ISO con PowerShell:

```
Get-FileHash .\ubuntu-24.04.1-live-server-amd64.iso -Algorithm SHA256
```

PS C:\Users\edgar.diaz> cd D:\DESCARGAS\edgar.diaz\Descargas		
PS D:\DESCARGAS\edgar.diaz\Descargas> Get-FileHash .\ubuntu-24.04.1-live-server-amd64.iso -Algorithm SHA256		
Algorithm Hash Path		
-----	-----	-----
SHA256	E240E4B801F7BB68C20D1356B60968AD0C33A41D00D828E74CEB3364A0317BE9	D:\DESCARGAS\edgar.diaz\Descargas\ubuntu-24.04.1-live-server-amd64.iso

Anoté el hash completo y lo utilicé en la configuración de Packer.

No utilicé la opción "skip" ni "auto" en el campo iso_checksum por las siguientes razones:

- La opción "auto" no es válida en el builder vmware-iso de Packer. Aunque puede usarse en otros contextos, en este caso genera un error porque auto no es reconocido como un valor hexadecimal válido.
- La opción "skip" tampoco es aceptada por vmware-iso. Aunque en otros builders permite omitir la verificación del checksum, aquí provoca un fallo de validación. Packer espera un valor de checksum válido y completo en formato hexadecimal.

Por esta razón, opté por calcular el hash SHA256 real de la ISO utilizando PowerShell y lo introduce explícitamente en el archivo .pkr.hcl. Esto no solo resolvió los errores de validación, sino que también me permitió confirmar la integridad del archivo ISO antes de usarlo en el proceso de automatización.

Explicación del archivo ubuntu.pkr.hcl creado en VSCode

Para definir el proceso de automatización con Packer, utilicé Visual Studio Code como editor principal. Gracias a su ligereza y extensibilidad, me permitió trabajar de forma ágil con archivos .hcl, utilizando la extensión oficial de HashiCorp Packer, que facilita la validación de sintaxis, autocompletado y resaltado semántico.

A continuación, explico el contenido del archivo ubuntu.pkr.hcl línea por línea:

Bloque packer

```
packer {
  required_plugins {
    vmware = {
      version = ">= 1.0.0"
      source  = "github.com/hashicorp/vmware"
    }
  }
}
```

Este bloque define los plugins requeridos por Packer. En este caso, estoy usando el plugin vmware, que permite generar imágenes directamente compatibles con VMware Workstation Pro. Defino la versión mínima y su origen oficial.

Bloque source "vmware-iso" "ubuntu"

Este es el corazón del archivo: especifica cómo debe construirse la máquina virtual.

ISO

```
source "vmware-iso" "ubuntu" {
  iso_url      = "D:/DESCARGAS/edgar.diaz/Descargas/ubuntu-24.04.1-live-server-amd64.iso"
  iso_checksum = "SHA256:E240E4B801F7BB68C20D1356B60968AD0C33A41D00D828E74CEB3364A0317BE9"
```

- Indico la ruta local a la ISO de Ubuntu Server.
- Especifico su checksum SHA256 real para validar la integridad del archivo antes de comenzar.

SSH y comunicación

```
14  communicator      = "ssh"
15  ssh_username     = "ubuntu"
16  ssh_password     = "ubuntu"
17  ssh_timeout       = "20m"
18
```

Estos valores definen cómo Packer se comunicará con la VM una vez se haya instalado Ubuntu. Utilizo SSH con credenciales básicas (usuario/contraseña) para simplificar el proceso de provisión.

Apagado

```
19  shutdown_command  = "echo 'ubuntu' | sudo -S shutdown -P now"
20
```

Comando que se ejecuta al final del proceso para apagar la VM de forma controlada, usando privilegios sudo.

Hardware

```
vm_name          = "ubuntu24-template"
guest_os_type    = "ubuntu-64"

disk_size        = 10240
memory           = 2048
cpus              = 2
```

Aquí defino las especificaciones de hardware de la VM: nombre, tipo de sistema operativo, tamaño de disco, RAM y número de CPUs.

Boot automático

```
boot_command = [
    "<esc><wait>",
    "<enter><wait>",
]
```

Esto representa las teclas simuladas que Packer envía al arrancar la ISO. Con `<esc>` y `<enter>` inicio el instalador de Ubuntu manualmente. Si quisiera hacer una instalación desatendida, podría añadir más comandos aquí o usar cloud-init.

Modo visual

```
headless = false
```

Esto indica que quiero ver la máquina virtual durante el proceso (útil para depuración). Si lo pusiera en `true`, la VM se ejecutaría en segundo plano.

Bloque build

```
build {
  sources = ["source.vmware-iso.ubuntu"]
}
```

Este bloque simplemente activa el builder definido arriba y le dice a Packer que lo utilice al construir la imagen.

Validación y construcción de la imagen

Una vez definido el archivo `ubuntu.pkr.hcl`, procedí a inicializar y validar el proyecto con los siguientes comandos desde la terminal de Visual Studio Code:

```
PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared> packer init .
PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared> packer validate .
```

Estos pasos sirven para:

- `init`: descargar e instalar los plugins definidos en el bloque `packer.required_plugins`.
- `validate`: comprobar que la sintaxis del archivo `.pkr.hcl` es correcta y que no faltan valores.

El uso de Visual Studio Code y la extensión oficial de Packer me permitió detectar errores al instante y mantener una estructura clara y profesional del archivo.

Tras confirmar que no había errores, ejecuté el proceso de construcción:

```
PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared> packer build .
```

```

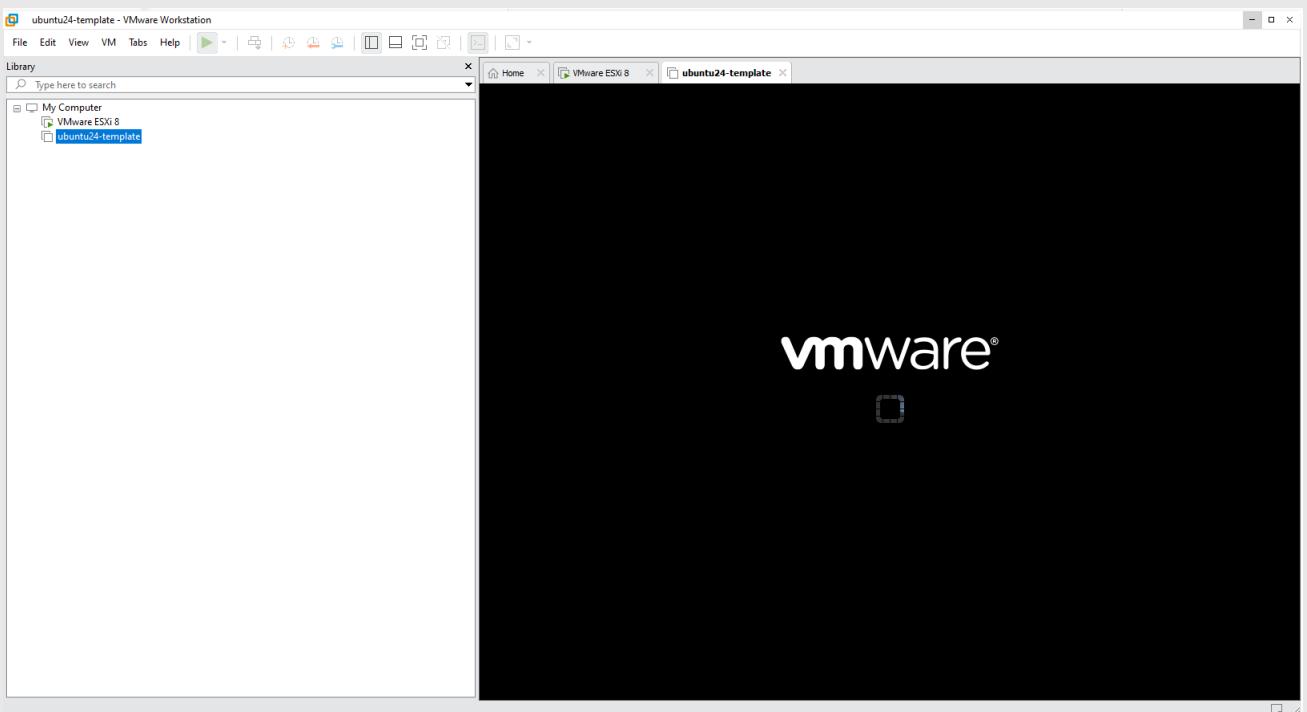
● PS C:\Users\edgar.diaz\Documents\edgar_spiral\Infrared> packer build .
● vmware-iso.ubuntu: output will be in this color.

--> vmware-iso.ubuntu: Retrieving ISO
--> vmware-iso.ubuntu: Trying D:/DESCARGAS/edgar.diaz/Descargas/ubuntu-24.04.1-live-server-and64.iso
--> vmware-iso.ubuntu: Trying file:///D:/DESCARGAS/edgar.diaz/Descargas/ubuntu-24.04.1-live-server-and64.iso?checksum=sha256%3Ae240e4b881f7bb68c20d135cb60968ad8c33a41d00d828e74ceb3364a0317be9
--> vmware-iso.ubuntu: Configuring output and export directories...
--> vmware-iso.ubuntu: Creating temporary private key for instance...
--> vmware-iso.ubuntu: Creating virtual machine disks
--> vmware-iso.ubuntu: Building and writing VMX file
--> vmware-iso.ubuntu: Starting virtual machine...
--> vmware-iso.ubuntu: Connecting to VNC...
--> vmware-iso.ubuntu: Waiting 10s for boot...
--> vmware-iso.ubuntu: Typing the boot command over VNC...
--> vmware-iso.ubuntu: Waiting for SSH to become available...
--> vmware-iso.ubuntu: Waiting for SSH...
--> vmware-iso.ubuntu: Stopping virtual machine...
--> vmware-iso.ubuntu: Deleting output directory...
Build 'vmware-iso.ubuntu' errored after 20 minutes 41 seconds: Timeout waiting for SSH.

==> Wait completed after 20 minutes 41 seconds
--> Some builds didn't complete successfully and had errors:
--> vmware-iso.ubuntu: timeout waiting for SSH.

--> Builds finished but no artifacts were created.
  
```

Este comando inició la generación automática de la imagen: Packer lanzó una nueva máquina virtual de Ubuntu en VMware Workstation Pro, arrancando desde la ISO especificada, y comenzó el proceso de instalación de forma guiada o automática según la configuración.



Automatización del despliegue con Terraform sobre vSphere.

Tras haber generado la imagen base de Ubuntu Server con Packer y convertirla a un formato compatible con VMware ESXi, pasamos a la siguiente fase del proyecto: el despliegue automatizado mediante Terraform.

Terraform nos permite definir y gestionar toda la infraestructura de forma declarativa, reutilizando la imagen generada. En este caso, el objetivo es crear una nueva máquina virtual en el entorno vSphere (gestionado por vCenter), utilizando como disco base el .vmdk previamente construido con Packer y subido manualmente al datastore.

Durante este proceso se automatizarán las siguientes acciones:

- Conexión a vCenter mediante las credenciales configuradas.
- Identificación del datacenter, el datastore, el host ESXi y la red virtual.
- Creación de una máquina virtual a partir de un disco ya existente.
- Asignación de CPU, memoria y red a la nueva máquina.
- (Opcionalmente) Inserción de una imagen ISO como CD-ROM.

Este despliegue demuestra cómo Terraform puede integrarse eficazmente con entornos de virtualización tradicionales como VMware, permitiendo replicar y escalar entornos de manera reproducible y controlada.

Activación del servicio vmrest en VMware

Terraform no interactúa directamente con VMware Workstation de forma nativa, por lo que fue necesario activar su API REST local, conocida como vmrest.

Para ello, ejecuté el siguiente comando desde la terminal de Windows (PowerShell o CMD) con permisos de administrador:

```
C:\Windows\system32>"C:\Program Files (x86)\VMware\VMware Workstation\vmrest.exe" -C
VMware Workstation REST API
Copyright (C) 2018-2024 VMware Inc.
All Rights Reserved

vmrest 1.3.1 build-23775571
Username:root
New password:
Password does not meet complexity requirements:
- Minimum 1 uppercase character
- Minimum 1 lowercase character
- Minimum 1 numeric digit
- Minimum 1 special character(!#$%&'()*+, -./:;<=>?@[{}]^_`{|}~)
- Length between 8 and 12
New password:
Retype new password:
Processing...
Credential updated successfully
```

Esto permitió crear un usuario y contraseña personalizados para acceder a la API REST. Después de configurar las credenciales, volví a ejecutar el comando sin argumentos para iniciar el servidor:

```
C:\Windows\system32>"C:\Program Files (x86)\VMware\VMware Workstation\vmrest.exe"
VMware Workstation REST API
Copyright (C) 2018-2024 VMware Inc.
All Rights Reserved

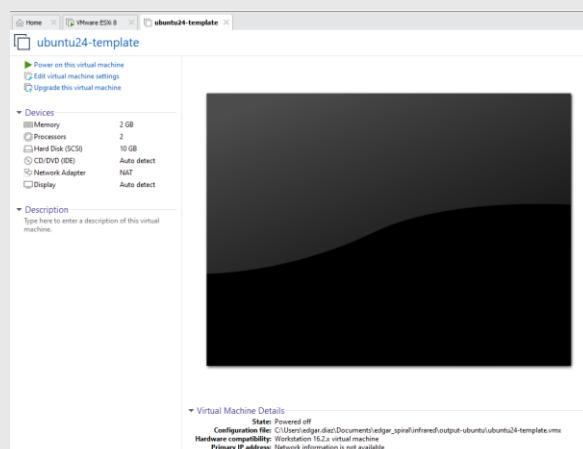
vmrest 1.3.1 build-23775571
-
Using the VMware Workstation UI while API calls are in progress is not recommended and may yield unexpected results.
-
Serving HTTP on 127.0.0.1:8697
-
Press Ctrl+C to stop.
```

Una vez iniciado correctamente, la consola mostró el siguiente mensaje:

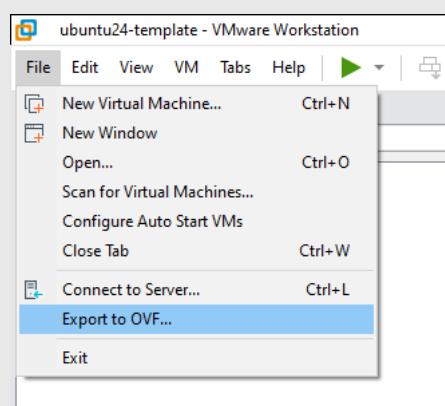
```
- Serving HTTP on 127.0.0.1:8697
-
```

Exportar mi VM desde VMware Workstation Pro a .ova

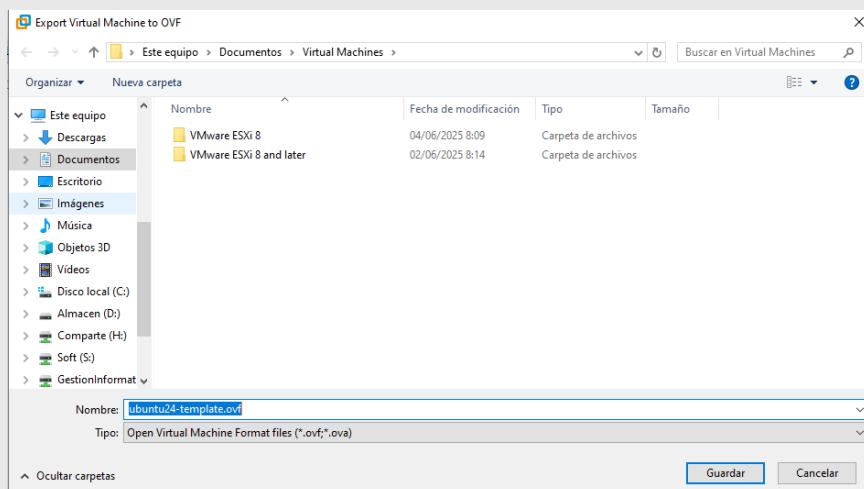
1. Abrí mi VM en VMware Workstation.



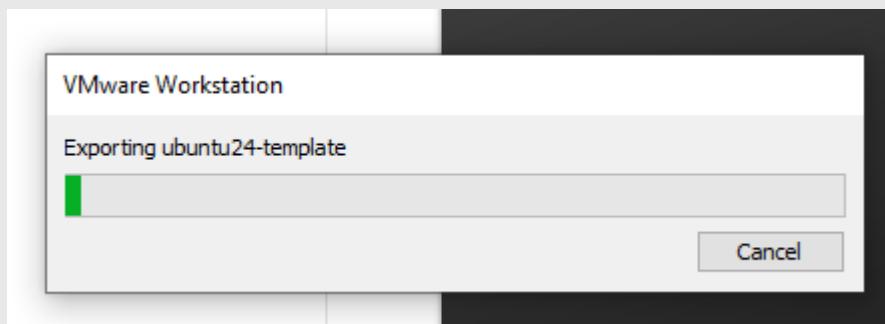
2. Menú: File > Export to OVF...



3. Selecciono .ova como tipo de exportación (ideal).



4. Guardo el archivo .ova en mi equipo y completo la exportación.



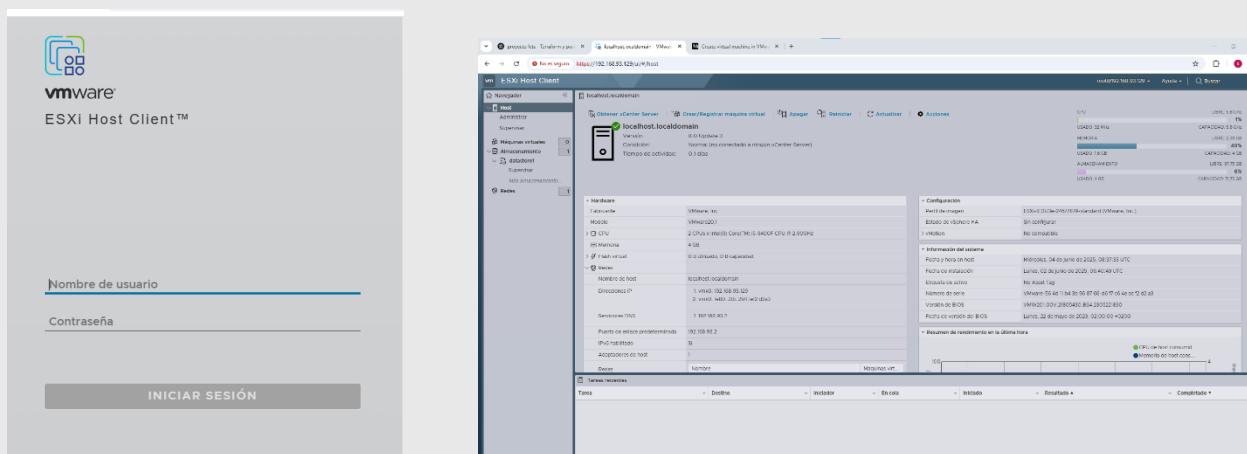
Subir la .ova a mi servidor ESXi

1. Abrí mi navegador:

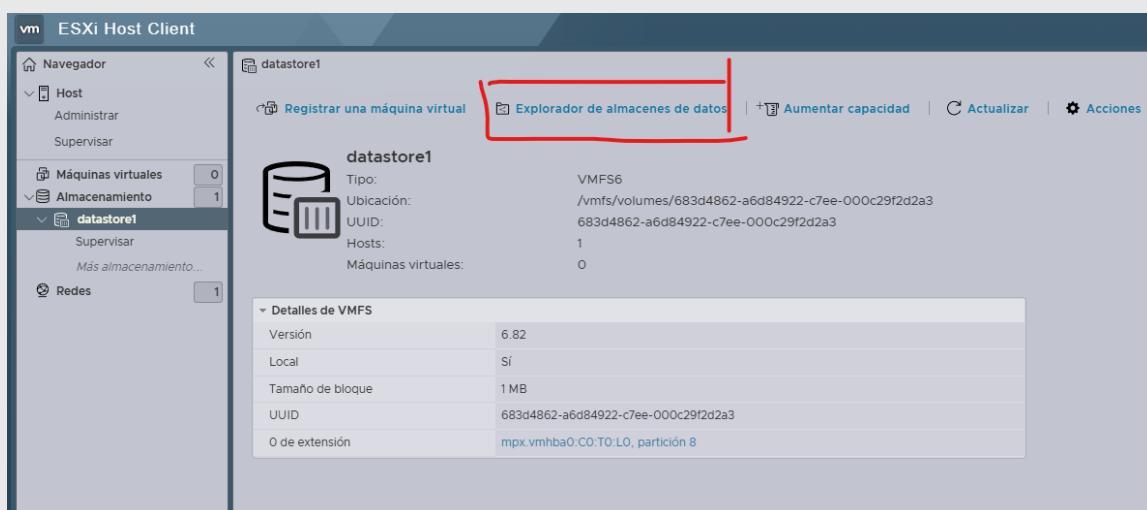
<https://192.168.93.129>



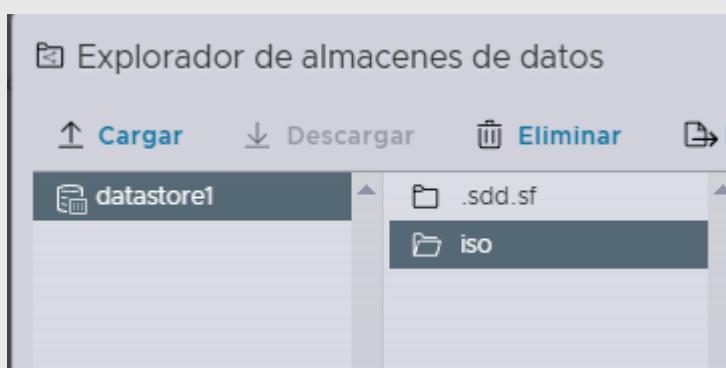
2. Inicié sesión con root



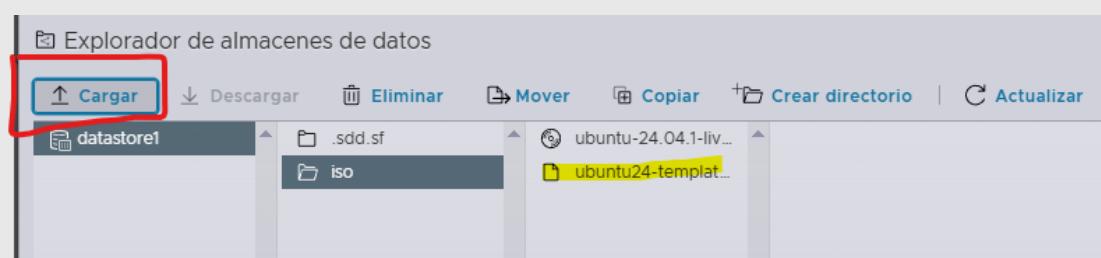
3. Fui a: Almacen > datastore1 > Explorador de almacenamiento de datos



4. Cree una carpeta



5. Hice clic en cargar y seleccioné mi archivo .ova



Despliegue con Terraform

¿Por qué usamos tres archivos?

main.tf

Contiene toda la lógica del despliegue, definición de recursos, proveedor y configuración de la VM.

variables.tf

Separa la declaración de variables para mayor claridad, flexibilidad y reutilización del código.

terraform.tfvars

Evita incluir información sensible (como contraseñas) en el código principal. Permite mantener las credenciales fuera de control de versiones y reutilizarlas de forma segura.

Explicación detallada del archivo main.tf

Bloque `terraform` – Declaración del proveedor requerido

```

1  terraform {
2      required_providers {
3          vsphere = {
4              source  = "hashicorp/vsphere"
5              version = ">= 2.12.0"
6          }
7      }
8  }
9

```

- Indica que este proyecto depende del proveedor `vsphere`.
- Especifica que debe descargarse desde el registro oficial de Terraform (`hashicorp/vsphere`).
- Se exige una versión mínima (2.12.0) para asegurar compatibilidad con funcionalidades modernas del vCenter

Bloque `provider` – Conexión con vCenter

```

provider "vsphere" {
    user            = var.vsphere_user
    password        = var.vsphere_password
    vsphere_server  = var.vsphere_server
    allow_unverified_ssl = true
}

```

Define cómo se conecta Terraform al vCenter Server.

Utiliza variables (`var.vsphere_user`, etc.) que se declaran en `variables.tf` y se rellenan en `terraform.tfvars`.

`allow_unverified_ssl = true` desactiva la verificación de certificados SSL, útil cuando se usa un certificado autofirmado en entornos de pruebas como este.

Datos de infraestructura (data)

`data "vsphere_datacenter"`

```
data "vsphere_datacenter" "dc" {
  name = "Datacenter"
}
```

- Recupera el objeto del datacenter donde están definidos los hosts, redes y datastores.
- El nombre debe coincidir exactamente con el que aparece en vCenter.

`data "vsphere_datastore"`

```
data "vsphere_datastore" "datastore" {
  name      = "SVR1_OS"
  datacenter_id = data.vsphere_datacenter.dc.id
}
```

- Recupera el datastore donde se encuentran los archivos .vmdk y .iso.
- "SVR1_OS" es el nombre del datastore asignado al host en vSphere.

`data "vsphere_host"`

```
data "vsphere_host" "host" {
  name      = "192.168.0.3" # O 0.5 o 0.7 si quieres usar otro host
  datacenter_id = data.vsphere_datacenter.dc.id
}
```

- Identifica el host ESXi donde se desplegará la VM.
- En este caso, el host se ha registrado en vCenter por su IP.

`data "vsphere_network"`

```
data "vsphere_network" "network" [
  name      = "VM Network"
  datacenter_id = data.vsphere_datacenter.dc.id
]
```

- Indica qué red virtual se conectará a la nueva máquina.
- "VM Network" es el nombre por defecto que viene en vSphere para redes gestionadas por switches virtuales estándar.

Recurso principal: creación de la VM

Configuración general

```
resource "vsphere_virtual_machine" "from_packer_disk" {
  name          = "infrared-from-disk"
  resource_pool_id = data.vsphere_host.host.resource_pool.id
  datastore_id    = data.vsphere_datastore.datastore.id

  num_cpus = 2
  memory   = 2048
  guest_id = "ubuntu64Guest"
```

- Se define una nueva máquina virtual llamada infrared-from-disk.
- Se asignan 2 CPUs, 2 GB de RAM y se indica que el sistema invitado es Ubuntu 64 bits.

Interfaz de red

```
network_interface {
  network_id  = data.vsphere_network.network.id
  adapter_type = "vmxnet3"
}
```

Conecta la VM a la red "VM Network" utilizando el adaptador vmxnet3 (eficiente y recomendado para Linux)

Disco base (imagen de Packer convertida)

```
disk {
  label      = "infrared-os-disk"
  path       = "[SVR1_OS] ISO5/ubuntu24-template.vmdk"
  datastore_id = data.vsphere_datastore.datastore.id
  attach     = true
  unit_number = 0
}
```

- Se adjunta un disco ya existente (generado con Packer, convertido a formato compatible con ESXi).
- El disco no se crea, se reutiliza con attach = true

CD-ROM con ISO de Ubuntu

```
cdrom {
  datastore_id = data.vsphere_datastore.datastore.id
  path        = "[SVR1_OS] ISO5/ubuntu-24.04.1-live-server-amd64.iso"
}
```

Añade una unidad de CD-ROM que apunta a una ISO. Se usa, por ejemplo, si necesitas acceso a herramientas de rescate o instalar desde ISO (opcional si el .vmdk ya es autoinstalado).

Tiempos de espera en red

```
wait_for_guest_net_timeout = 0
wait_for_guest_ip_timeout = 0
```

Desactiva la espera a que la máquina obtenga IP o conectividad. Útil en entornos cerrados o sin DHCP.

Declaración de variables – variables.tf

Este archivo define las variables de entrada necesarias para que Terraform funcione:

- vsphere_user: usuario con permisos en vCenter.
- vsphere_password: contraseña asociada (marcada como sensitive para protegerla).
- vsphere_server: dirección IP o nombre DNS del servidor vCenter.

```
C: > Users > edgar.diaz > Documents > edgar_spiral > infrared > terraform > variables.tf > ...
1  variable "vsphere_user" {
2    type = string
3  }
4
5  variable "vsphere_password" {
6    type = string
7    sensitive = true
8  }
9
10 variable "vsphere_server" {
11   type = string
12 }
13 }
```

¿Por qué es útil?

- Separa claramente la lógica (main.tf) de la configuración personalizada.
- Evita duplicar datos o escribir credenciales sensibles dentro del código principal.
- Facilita que otras personas (o scripts) puedan reutilizar el código sin modificar la lógica.

Valores concretos - terraform.tfvars

Este archivo proporciona los valores reales para las variables declaradas en variables.tf. Aquí se rellenan las credenciales y la IP del vCenter.

```
PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared\terraform>
vsphere_user      = "██████████"
vsphere_password = "██████████"
vsphere_server    = "192.168.0.20"
```

Ejecución del código

Una vez definidos los tres archivos .tf necesarios (main.tf, variables.tf, terraform.tfvars), procedí a ejecutar el despliegue de la infraestructura utilizando Terraform. El proceso se compone de varias fases clave:

1. Inicialización del entorno

terraform init

¿Qué hace?

- Descarga el proveedor necesario (hashicorp/vsphere) si no está presente.
- Configura el entorno de trabajo local de Terraform.
- Valida que la estructura del proyecto sea correcta.

Resultado:

```
PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared\terraform> terraform init
There are some problems with the CLI configuration:
Warning: Unable to open CLI configuration file
The CLI configuration file at
"C:\Users\edgar.diaz\AppData\Roaming\terraform.rc" does not exist.

Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/vsphere versions matching ">= 2.2.0"...
- Installing hashicorp/vsphere v2.12.0...
- Installed hashicorp/vsphere v2.12.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Warning: Additional provider information from registry
The remote registry returned warnings for registry.terraform.io/hashicorp/vsphere:
- For users on Terraform 0.13 or greater, this provider has moved to vmware/vsphere. Please update your source in required_providers.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

2. Validación del plan de despliegue

terraform plan

¿Qué hace?

- Compara el estado actual de la infraestructura con la definición del código.
- Muestra un plan detallado de las acciones que Terraform va a ejecutar.
- En este caso, mostrará que se va a crear (añadir) una nueva máquina virtual.

```
PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared\terraform> terraform plan
There are some problems with the CLI configuration:

Warning: Unable to open CLI configuration file
The CLI configuration file at
  "C:\Users\edgar.diaz\Documents\edgar_spiral\infrared\terraform\terraform.rc" does not exist.

data.vsphere_datacenter.dc: Reading...
data.vsphere_datacenter.dc: Read complete after 0s [id=ha-datacenter]
data.vsphere_host.host: Reading...
data.vsphere_datastore.datastore: Reading...
data.vsphere_network.network: Reading...
data.vsphere_network.network: Read complete after 0s [id=Network-VM Network]
data.vsphere_datastore.datastore: Read complete after 0s [id=68340f62-afdb4022-c7ee-000c29f2d2a3]
data.vsphere_host.host: Read complete after 0s [id=ha-host]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# vsphere_virtual_machine.from_packer_disk" "from_packer_disk"
+ resource "vsphere_virtual_machine" "from_packer_disk" {
  + annotation           = (known after apply)
  + boot_retry_delay     = 30000
  + cloud_init_script    = (known after apply)
  + cpu_limit             = 1
  + cpu_share_count      = (known after apply)
  + cpushare_percent     = 100
  + default_id_address   = (known after apply)
  + disk.backing          = "thin"
  + disk.backing_id       = "68340f62-afdb4022-c7ee-000c29f2d2a3"
  + disk.backing_type     = "vmfs"
  + disk.controller_label = "VMF0"
  + disk.controller_type  = "scsi"
  + disk.device_address   = (known after apply)
  + disk.mode              = "persistent"
  + disk.share_level      = "sharingNone"
  + disk.size_gb           = 10
  + disk.size_kb           = 104857600
  + disk.size_mb           = 102400
  + disk.size_raw_kb       = 10737418240
  + disk.size_raw_mb       = 10240000
  + disk.size_raw_gb       = 10
  + disk.thin_provisioning = true
  + disk.unmap_on_delete   = false
  + disk.unmap_on_remove   = false
  + key                   = 0
  + label                 = "Infrared-os-disk"
  + path                  = "[datastore1]/iso/ubuntu24-template-disk1.vmdk"
  + storage_policy_id     = (known after apply)
  + thin_provisioning      = true
  + unit_number            = 0
  + uuid                  = (known after apply)
  + write_through          = false
}

+ network_interface {
  + adapter_type          = "vmnet3"
  + bandwidth_limit        = -1
  + bandwidth_reservation = 0
  + bandwidth_sharing_count = (known after apply)
  + bandwidth_sharing_level = "normal"
  + device_address         = (known after apply)
  + key                   = (known after apply)
  + mac_address            = (known after apply)
  + network_id             = "Network-VM Network"
}

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared\terraform>
```

```
+ disk {
  + back      = true
  + controller_type = "scsi"
  + datastore_id = "68340f62-afdb4022-c7ee-000c29f2d2a3"
  + device_address = (known after apply)
  + disk_mode = "persistent"
  + disk_sharing = "sharingNone"
  + eagerzero_crub = false
  + in_limit = 1
  + io_reservation = 0
  + io_share_count = 0
  + io_share_level = "normal"
  + keep_on_remove = false
  + key = 0
  + label = "Infrared-os-disk"
  + path = "[datastore1]/iso/ubuntu24-template-disk1.vmdk"
  + storage_policy_id = (known after apply)
  + thin_provisioned = true
  + unit_number = 0
  + uid = (known after apply)
  + write_through = false
}

+ network_interface {
  + adapter_type = "vmnet3"
  + bandwidth_limit = -1
  + bandwidth_reservation = 0
  + bandwidth_sharing_count = (known after apply)
  + bandwidth_sharing_level = "normal"
  + device_address = (known after apply)
  + key = (known after apply)
  + mac_address = (known after apply)
  + network_id = "Network-VM Network"
}

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared\terraform>
```

3. Aplicación del despliegue

terraform apply

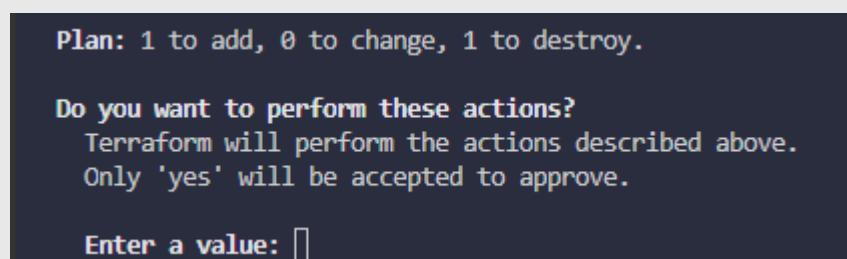
Al ejecutar este comando, Terraform mostrará nuevamente el plan de acciones y solicitará confirmación:

```
PS C:\Users\edgar.diaz\Documents\edgar_spiral\infrared\terraform> terraform apply
data.vsphere_datacenter.dc: Reading...
data.vsphere_datacenter.dc: Read complete after 0s [id=datacenter-3]
data.vsphere_network.network: Reading...
data.vsphere_host.host: Reading...
data.vsphere_datastore.datastore: Reading...
data.vsphere_network.network: Read complete after 0s [id=network-15]
data.vsphere_host.host: Read complete after 0s [id=host-3159]
data.vsphere_datastore.datastore: Read complete after 0s [id=datastore-3186]
vsphere_virtual_machine.from_packer_disk: Refreshing state... [id=420d43b2-2eca-e6d7-2cec-b14d1275d5ef]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:
```

Aquí se debe escribir yes para autorizar la creación de los recursos.



¿Qué hace exactamente?

- Se conecta al servidor vCenter indicado.
- Recupera los recursos definidos (datacenter, datastore, host, network).
- Crea una nueva máquina virtual a partir del disco .vmdk especificado.
- La configura con CPU, RAM, red y CD-ROM según lo definido en main.tf.

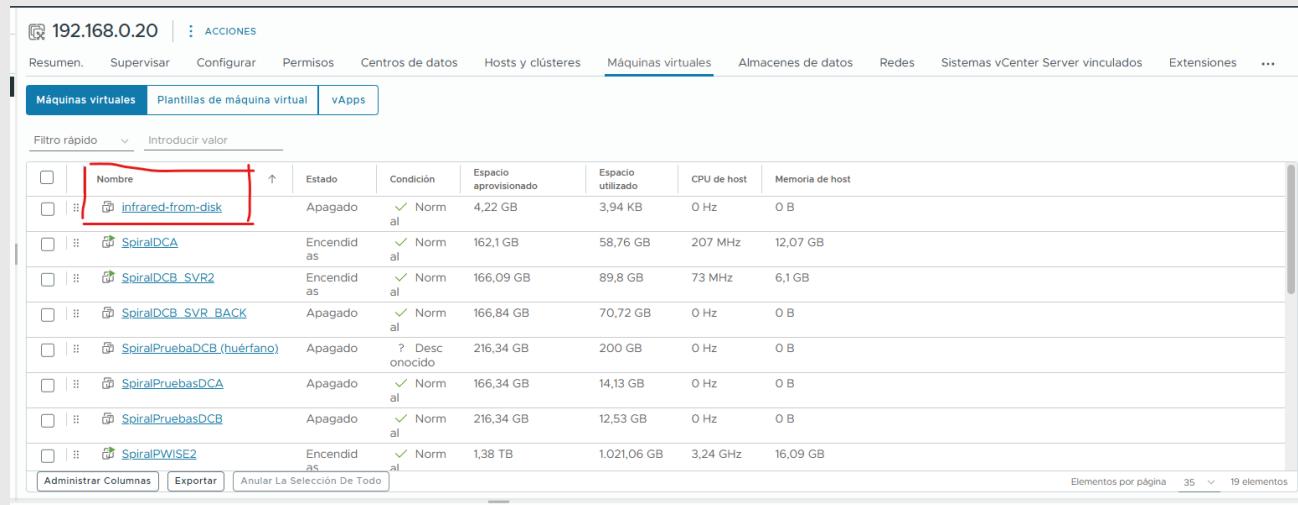
Visualización en vSphere

Una vez ejecutado correctamente el comando terraform apply y completada la creación del recurso, la nueva máquina virtual aparece automáticamente dentro del inventario del entorno vSphere.

En concreto, se puede observar:

- La VM infrared-from-disk listada bajo el Datacenter especificado.
- Asociada al host ESXi definido en el archivo main.tf.
- Ubicada en el datastore SVR1_OS y conectada a la red virtual VM Network.
- La máquina queda creada y visible en el panel izquierdo del vSphere Client, y puede iniciarse manualmente si no lo hace automáticamente.

Este comportamiento confirma que Terraform ha interactuado correctamente con el entorno de vCenter, que ha reconocido el disco base (.vmdk) generado por Packer, y ha registrado la máquina según la configuración declarada. Desde este momento, la gestión operativa de la VM puede realizarse tanto desde Terraform como desde la propia interfaz de vSphere



	Nombre	Estado	Condición	Espacio aprovisionado	Espacio utilizado	CPU de host	Memoria de host
<input type="checkbox"/>	infrared-from-disk	Apagado	Norm	4,22 GB	3,94 KB	0 Hz	0 B
<input type="checkbox"/>	SpiralDCA	Encendidas	Norm	162,1GB	58,76 GB	207 MHz	12,07 GB
<input type="checkbox"/>	SpiralDCB_SVR2	Encendidas	Norm	166,09 GB	89,8 GB	73 MHz	6,1 GB
<input type="checkbox"/>	SpiralDCB_SVR_BACK	Apagado	Norm	166,84 GB	70,72 GB	0 Hz	0 B
<input type="checkbox"/>	SpiralPruebaDCB.(huérfano)	Apagado	? Desc onocido	216,34 GB	200 GB	0 Hz	0 B
<input type="checkbox"/>	SpiralPruebasDCA	Apagado	Norm	166,34 GB	14,13 GB	0 Hz	0 B
<input type="checkbox"/>	SpiralPruebasDCB	Apagado	Norm	216,34 GB	12,53 GB	0 Hz	0 B
<input type="checkbox"/>	SpiralPWISE2	Encendidas	Norm	1,38 TB	1.021,06 GB	3,24 GHz	16,09 GB

Entorno completamente desplegado

Con la ejecución satisfactoria del código Terraform y la confirmación de que la máquina virtual infrared-from-disk aparece correctamente registrada en el entorno vSphere, puedo afirmar que ya tengo todo el entorno del proyecto InfraRed desplegado y funcionando.

En este punto del proyecto:

- He generado, probado y validado una imagen base de Ubuntu Server utilizando Packer.
- He exportado dicha imagen desde VMware Workstation Pro al host ESXi, convirtiéndola a formato .vmdk compatible.
- Posteriormente, he utilizado Terraform para definir y desplegar una nueva máquina virtual de forma automatizada a partir de esa imagen, conectándome al vCenter de la infraestructura.
- La máquina queda correctamente configurada: asociada al datastore, conectada a la red virtual y con recursos asignados de manera declarativa.

Gracias a este flujo de trabajo, he aplicado con éxito el enfoque de Infraestructura como Código (IaC) para automatizar tareas de despliegue en un entorno realista basado en VMware. El resultado es una solución completamente funcional, fácilmente replicable y preparada para escalar en futuras versiones del proyecto.

Con esto, doy por finalizado el proceso de despliegue: el entorno InfraRed ya está completamente operativo.



Pruebas

Antes de dar por finalizado el proyecto InfraRed, consideré esencial someter todo el proceso a una fase de pruebas prácticas. El objetivo principal era verificar que cada etapa del despliegue desde la generación de imágenes con Packer hasta su provisión final con Terraform funcionara de forma coherente y replicable en distintos entornos.

Durante estas pruebas no solo validé que los procedimientos fueran funcionales, sino que también identifiqué limitaciones técnicas, incompatibilidades y puntos críticos que me obligaron a modificar el enfoque inicial. En especial, algunas pruebas me permitieron descubrir por qué determinadas herramientas no eran viables con ciertos hipervisores (como ESXi en su versión gratuita) y cómo adaptar la solución a un escenario realista y operativo.

En este apartado detallo las pruebas clave que realicé, los errores encontrados, las soluciones aplicadas y las conclusiones extraídas. Todo esto me permitió garantizar la estabilidad y utilidad del sistema automatizado propuesto.

Creación de la imagen base con Packer

Objetivo:

Validar si Packer puede generar una imagen de Ubuntu Server 24.04.1 directamente en un host VMware ESXi Free Edition utilizando el plugin vsphere-iso.

Resultado:

Fallido. Packer no consiguió crear ni desplegar la imagen directamente en ESXi.

Errores detectados:

- **ServerFaultCode: The operation is not supported on the object**

Este error se produce porque la versión gratuita de ESXi no permite operaciones de escritura automatizadas vía API SOAP, que es el método utilizado por el plugin oficial vsphere-iso de Packer.

- **Licencia limitada**

La versión ESXi Free no soporta las llamadas remotas necesarias para la creación automatizada de VMs desde herramientas como Terraform o Packer. Solo una instancia vCenter o una licencia paga habilitan estas funciones.

Alternativa aplicada:

Éxito en Workstation Pro

La imagen fue generada correctamente en VMware Workstation Pro utilizando el plugin vmware-iso de Packer, que interactúa directamente con el entorno de virtualización local.

Pasos realizados:

1. Se cambió el source de vsphere-iso a vmware-iso.
2. Se adaptó el archivo .hcl para especificar la ISO local y los recursos (CPU, RAM, disco).
3. La imagen se generó en formato .vmx y .vmdk.
4. Posteriormente se exportó al servidor ESXi manualmente vía OVA o vía Datastore.

Conclusión de la prueba:

La automatización con Packer no es compatible directamente con ESXi Free, debido a sus restricciones de licencia. Por ello, opté por crear la imagen en Workstation Pro y exportarla al entorno empresarial para su despliegue con Terraform en vSphere.

Despliegue con Terraform desde imagen creada en VMware Workstation Pro

Objetivo:

Probar si es posible usar Terraform para desplegar una máquina virtual a partir de una imagen generada previamente por Packer en VMware Workstation Pro.

Resultado:

Fallido. Terraform no permite consumir directamente imágenes .vmx o .vmdk generadas en entornos locales como VMware Workstation Pro, debido a limitaciones técnicas y de compatibilidad.

Errores detectados:

- **Inexistencia de soporte oficial**

Terraform no incluye soporte nativo para VMware Workstation Pro. Se probó el proveedor comunitario [elsudano/vmworkstation](#), pero presentó múltiples fallos:

- No fue posible inicializar correctamente el entorno (terraform init).
- Errores recurrentes de instalación del proveedor o de compatibilidad con la arquitectura de Windows.
- Documentación escasa y falta de mantenimiento en el repositorio.

- **vmrest API limitada**

Aunque se activó el servicio vmrest de VMware Workstation para exponer una API, esta no incluye funcionalidades clave para trabajar con Terraform (creación, configuración o provisión de máquinas virtuales).

- **No compatible con archivos .vmx de Packer**

Terraform requiere imágenes gestionadas desde una infraestructura vSphere centralizada para poder desplegar VMs de forma declarativa. Los archivos .vmx generados en Workstation no pueden ser consumidos directamente.

```
Error: error creating virtual machine: ServerFaultCode: Current license or ESXi version prohibits execution of the requested operation.

with vsphere_virtual_machine.from_packer_disk,
on main.tf line 36, in resource "vsphere_virtual_machine" "from_packer_disk":
36: resource "vsphere_virtual_machine" "from_packer_disk" {
```

Alternativa aplicada:

Uso de vSphere como backend

Se optó por exportar la imagen generada con Packer desde VMware Workstation Pro al servidor ESXi como .ovf/.vmdk, para luego ser consumida por Terraform conectado a vSphere.

Pasos realizados:

1. La imagen de Ubuntu Server fue generada localmente con Packer y el plugin vmware-iso.
2. Se exportó al servidor ESXi manualmente desde Workstation en formato OVA.
3. Se utilizó el proveedor oficial hashicorp/vsphere en Terraform para conectarse al vCenter.
4. Terraform desplegó la máquina virtual correctamente utilizando la imagen ya cargada en el datastore del servidor.

Conclusión de la prueba:

Terraform no puede trabajar directamente con imágenes generadas en VMware Workstation. La alternativa viable es exportar esas imágenes a una infraestructura ESXi/vSphere, donde Terraform sí puede gestionarlas de forma declarativa y estable. Esta arquitectura híbrida permitió conservar la automatización con Packer y el control del despliegue con Terraform.

Elección de VMware vSphere en lugar de ESXi Free para el despliegue con Terraform

Objetivo:

Determinar cuál es la plataforma más adecuada para utilizar Terraform en el despliegue de máquinas virtuales: VMware ESXi Free Edition o VMware vSphere gestionado por vCenter.

Resultado:

Fallido con ESXi Free. Éxito con vSphere (vCenter + ESXi).

Motivos técnicos del fallo con ESXi Free:

- **Limitaciones de licencia:**

La edición gratuita de ESXi no permite operaciones automatizadas mediante API, que son necesarias para que Terraform cree, modifique o destruya máquinas virtuales. Aparece el error: ServerFaultCode: The operation is not supported on the object.

- **Falta de integración con vSphere SDK:**

Terraform utiliza el proveedor oficial hashicorp/vsphere, el cual requiere conectarse a un entorno gestionado por vCenter para operar correctamente. ESXi en modo standalone no permite ejecutar llamadas SOAP complejas como CloneVM_Task o DeployVM, bloqueando toda automatización.

- **Terraform requiere un punto de gestión centralizado:**

Terraform necesita acceder a un inventario completo del entorno (datastores, hosts, redes, plantillas...), algo que solo está disponible a través de vCenter. ESXi standalone no expone esta estructura jerárquica ni permite manipular recursos complejos desde fuera.

Alternativa aplicada: uso de VMware vSphere gestionado por vCenter

Resultado exitoso

Una vez desplegada la imagen generada con Packer en el datastore del servidor ESXi, me conecté al entorno desde Terraform utilizando las credenciales del vCenter. Desde ahí, Terraform pudo:

- Detectar correctamente el datacenter, host, red y plantilla .vmdk.
- Crear una nueva máquina virtual con parámetros definidos por código.
- Aplicar el plan de despliegue automáticamente (terraform apply) y validar su estado (terraform plan).

Conclusión de la prueba:

Terraform no es compatible con ESXi Free Edition debido a sus restricciones técnicas y de licencia. Por tanto, opté por utilizar vSphere con vCenter, lo que permitió un despliegue automatizado, estructurado y reproducible del entorno generado con Packer. Esta decisión fue clave para completar con éxito la fase final del proyecto InfraRed.

Limitaciones de Packer y Terraform en la creación de imágenes

Objetivo:

Analizar si es posible definir múltiples máquinas virtuales dentro del mismo archivo de configuración de Packer, y evaluar la compatibilidad de Packer y Terraform con diferentes sistemas operativos o imágenes ISO.

Resultado:

Límitado. Packer no permite crear múltiples imágenes en paralelo dentro de un mismo bloque de configuración simple, y tanto Packer como Terraform presentan limitaciones de compatibilidad con ciertos sistemas operativos e ISOs no estándar.

Limitación 1 – Solo una máquina por build en Packer

- Aunque Packer permite definir múltiples "sources" y "builds", en la práctica cada archivo de configuración (.pkr.hcl) está orientado a generar una sola imagen concreta por ejecución.
- No es posible definir múltiples máquinas virtuales diferentes en el mismo bloque si se pretende construirlas todas a la vez con configuraciones independientes.
- Intentar esto puede provocar errores de estructura, conflictos de variables o rutas incompatibles entre builds.

Alternativa:

Crear varios archivos .pkr.hcl, uno por sistema o plantilla, y ejecutarlos por separado.

Limitación 2 – Compatibilidad parcial con sistemas operativos

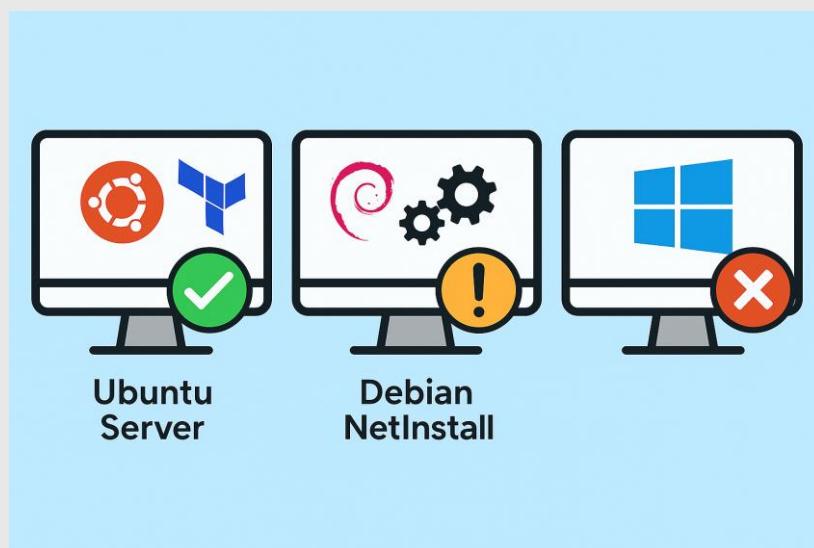
- Packer y Terraform **no funcionan con todos los sistemas operativos**, especialmente si:
 - La ISO no es "bootable" de forma tradicional.
 - Requiere interacción gráfica o pasos de instalación complejos que no pueden ser automatizados.
 - Utiliza interfaces de instalación no compatibles con los comandos de teclado simulados (boot_command).
- También se han detectado problemas al usar ISOs personalizadas o distribuciones ligeras con flujos de instalación atípicos.

Ejemplos reales comprobados:

- **Ubuntu Server 22.04 / 24.04** → *funciona correctamente* con Packer y Terraform.
- **Debian NetInstall** → requiere configuraciones especiales y más comandos.
- **Windows 10 / 11** → necesita herramientas específicas (como autounattend.xml) y configuraciones complejas que no se abordaron en este proyecto.

Conclusión de la prueba:

Packer está pensado para generar una imagen por definición, no múltiples VMs a la vez. Además, tanto Packer como Terraform no son compatibles universalmente con todas las ISOs o sistemas operativos, especialmente si no cumplen ciertos estándares de instalación automatizada. Por ello, durante el desarrollo de InfraRed me centré en utilizar Ubuntu Server LTS, una distribución totalmente compatible, documentada y estable.



Repositorio en GitHub del proyecto

Con el objetivo de documentar, versionar y facilitar la reutilización del trabajo realizado, he creado un repositorio público en GitHub que contiene todos los archivos generados durante la implementación del proyecto InfraRed.

En el repositorio se incluyen:

- Archivos de configuración de Packer (ubuntu.pkr.hcl, scripts de provisión).
- Código de despliegue con Terraform (main.tf, variables.tf, terraform.tfvars).
- Scripts utilizados para automatizar la instalación de servicios en la imagen base.
- Estructura de carpetas organizada por fases (packer/, terraform/, capturas/).
- Documentación técnica adicional en formato README.

Enlace al repositorio: <https://github.com/Edgar5422/Infrared>

PARTE ECONÓMICA PLAN DE EMPRESA

Datos del emprendedor

Mi nombre es Edgar Díaz Martínez y actualmente formo parte del departamento de IT de una empresa de consultoría tecnológica orientada a ofrecer soluciones informáticas a empresas de diversos sectores. Dentro de mi puesto, he detectado la oportunidad de desarrollar un sistema de automatización interna para el despliegue de entornos virtuales, que he denominado InfraRed.

Desde el departamento de IT de nuestra empresa de consultoría tecnológica, vamos a llevar a cabo el desarrollo del proyecto llamado InfraRed, una solución interna destinada a automatizar el despliegue de entornos virtuales en VMware. Aunque se trata de una propuesta simulada en el marco de un entorno académico, hemos querido plantearlo desde una perspectiva realista, abordando los recursos necesarios, el impacto económico estimado y su aplicación profesional dentro de un equipo técnico.

Mi currículum:


europass

Edgar Díaz Martínez

Nacionalidad: Española Fecha de nacimiento: 22/02/2002  Número de teléfono: (+34) 615642880

 Dirección de correo electrónico: ediazm03@educantabria.es

 Domicilio: Edificio Villagualdupe Portal 3 Bajo B, 39680 Ontaneda (España)

EXPERIENCIA LABORAL

Técnico en sistemas microinformáticos
AUTÓNOMO [06/09/2021 – Actual]
 País: España

- Diagnóstico y reparación de hardware y software
- Mantenimiento de sistemas informáticos para clientes particulares y empresas
- Implementación y configuración de redes locales
- Asesoramiento en seguridad informática y soluciones tecnológicas

Becario
Maderas Jose Saiz [20/03/2022 – 15/06/2022]
 Población: Cantabria | País: España

- mantenimiento de ordenadores
- relaciones con los proveedores
- análisis de facturas
- copias de seguridad
- creación de redes

EDUCACIÓN Y FORMACIÓN

Técnico en Sistemas microinformáticos y redes
IES. Vega de Toranzo [2021 – 2023]
 Dirección: 39680 Cantabria (España) | Sitio web: [iesvegadetoranzo.es](https://www.educantabria.es/web/ies-vega-detoranzo.es)

TÉCNICO SUPERIOR EN ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED
IES Miguel Herrero Pereda [2023 – 2025]
 Población: Torrelavega | País: España | Sitio web: <https://www.educantabria.es/web/ies-miguel-herrero-pereda>

COMPETENCIAS DE IDIOMAS

Lengua(s) materna(s): español

Otro(s) idioma(s):

inglés

COMPRENSIÓN AUDITIVA B1

COMPRENSIÓN LECTORA B1 EXPRESIÓN ESCRITA A2

PRODUCCIÓN ORAL B1 INTERACCIÓN ORAL A2

Niveles: A1 y A2 (usuario básico), B1 y B2 (usuario independiente), C1 y C2 (usuario competente)

COMPETENCIAS DIGITALES

Administración de sistemas operativos Windows y Linux

Gestión y configuración de redes informáticas

Implementación y administración de servidores (Windows Server, Linux)

Virtualización con VMware, VirtualBox

Ciberseguridad y gestión de cortafuegos

Bases de datos MySQL

Uso y gestión de herramientas de monitorización

Desarrollo básico en Bash y PowerShell

Administración y gestión de entornos en la nube (AWS)

PERMISO DE CONDUCCIÓN

Permiso de conducción: B

COMPETENCIAS INTERPERSONALES Y DE COMUNICACIÓN**CARACTERÍSTICAS PERSONALES**

- Capacidad de resolución de problemas
- Trabajo en equipo y colaboración
- Aprendizaje autónomo y adaptación a nuevas tecnologías
- Gestión del tiempo y organización

Nombre de la idea y logo

El proyecto se llama InfraRed, una combinación entre “Infraestructura” y “Red”, que hace referencia directa a su propósito: la creación y automatización de entornos virtuales sobre redes internas usando tecnologías de virtualización. El logotipo simula un cubo de datos con colores técnicos (azul oscuro y rojo), transmitiendo tecnología, orden y estructura.



Registro: Tras una búsqueda en la Oficina Española de Patentes y Marcas (OEPM), no consta ningún registro con ese nombre relacionado con soluciones informáticas, por lo que sería viable registrarlo si la empresa decide escalar el proyecto.

Responsabilidad social

Aunque InfraRed es un proyecto de carácter técnico desarrollado internamente por el departamento de IT, también contribuye a la responsabilidad social de la empresa en varios aspectos clave relacionados con la mejora de la calidad de vida de los usuarios internos (clientes internos), la eficiencia organizativa y el uso responsable de los recursos tecnológicos.



RESPONSABILIDAD SOCIAL

Cómo se mejora la calidad de vida de los consumidores (clientes) o de otros

The infographic features a central graphic of two hands holding a large red heart. To the left, a man in a suit stands next to a potted plant, holding a briefcase. To the right, a woman stands holding a laptop that displays the word "INFRARED". Various icons like a gear, a checkmark, a dollar sign, and a small plant are scattered around the central heart, symbolizing different aspects of corporate social responsibility.

Mejora de la calidad de vida de los consumidores internos

- **Reducción del estrés técnico:** InfraRed minimiza las tareas repetitivas, manuales y propensas a errores en la creación de entornos virtuales. Esto permite a los técnicos centrarse en tareas más creativas y estratégicas, mejorando su motivación y bienestar laboral.
- **Ahorro de tiempo y aumento de la productividad:** Gracias al despliegue automatizado de máquinas virtuales, los departamentos de desarrollo, soporte o formación pueden trabajar con mayor agilidad, sin depender de procesos lentos ni solicitudes constantes al equipo de sistemas.
- **Acceso igualitario a entornos estándar:** InfraRed garantiza que todos los usuarios internos dispongan de entornos virtuales homogéneos y funcionales, sin diferencias técnicas que generen desigualdades o retrasos entre equipos.
- **Facilitación del aprendizaje:** Al ofrecer entornos preconfigurados listos para usar, InfraRed se convierte también en una herramienta útil en procesos de formación interna, ayudando a capacitar a nuevos empleados o actualizar conocimientos técnicos de forma rápida y segura.

Estudio de validación interna del proyecto InfraRed

Con el fin de valorar el interés y la utilidad real del proyecto InfraRed dentro de la estructura de la empresa, he llevado a cabo un proceso de validación interna mediante una encuesta dirigida a compañeros del departamento técnico, entre ellos desarrolladores, técnicos de sistemas y personal de soporte. Esta acción tenía como objetivo conocer de primera mano la percepción del equipo sobre la problemática actual relacionada con el despliegue manual de entornos virtuales, así como evaluar la aceptación de una posible solución automatizada como InfraRed.

Participación y metodología

- La encuesta fue enviada de forma interna a 12 personas del equipo técnico.
- Se utilizó un formulario anónimo con preguntas cerradas.
- La muestra incluyó perfiles con diferentes niveles de experiencia, desde técnicos junior hasta responsables de proyecto.

Encuesta de Validación Interna**Proyecto InfraRed – Departamento de IT**

Como parte del desarrollo del proyecto InfraRed, te pedimos que respondas sinceramente a las siguientes preguntas. Tu opinión nos ayudará a validar esta herramienta de automatización de entornos virtuales.

1. ¿Qué grado de utilidad crees que tendría una herramienta automatizada para el despliegue de entornos virtuales?

- Nada útil Poco útil Útil Muy útil

2. ¿Consideras que una herramienta automatizada te ahorraría tiempo en tus tareas habituales?

- En absoluto No estoy seguro/a Probablemente Sin duda

3. ¿A qué tipo de entornos te gustaría acceder de forma preconfigurada?

- Entornos de desarrollo Entornos de prueba Ambos Ninguno

4. ¿Crees que esta herramienta sería útil para los procesos de formación interna?

- Nada útil Poco útil Útil Muy útil

5. ¿Has tenido que desplegar entornos manualmente en el último mes?

- Sí No

6. ¿Con qué frecuencia sueles necesitar nuevas máquinas virtuales en tu trabajo?

- A diario Varias veces por semana Ocasionalmente Rara vez

7. ¿Qué sistema utilizas actualmente para crear entornos virtuales?

- Instalación desde ISO Clonado manual Uso de plantillas Otro

8. ¿Conoces el funcionamiento básico de herramientas como Packer o Terraform?

- Sí, las uso habitualmente Sí, de forma general Solo de oídas No

9. ¿Qué servicios te gustaría que incluyeran las plantillas preconfiguradas?

- Apache Nginx SSH Usuario configurado Otros (especificar):

Encuesta de Validación Interna**Proyecto InfraRed – Departamento de IT**

Como parte del desarrollo del proyecto InfraRed, te pedimos que respondas sinceramente a las siguientes preguntas. Tu opinión nos ayudará a validar esta herramienta de automatización de entornos virtuales.

10. ¿Cuál es tu mayor dificultad al crear entornos virtuales actualmente?

- Tiempo de instalación Errores de configuración Falta de plantillas Otro

11. ¿Te sentirías cómodo utilizando InfraRed con una pequeña guía o manual?

- Sí Sí, si está bien explicado No No lo sé

12. ¿Qué ventajas ves en disponer de entornos virtuales listos para usar?

- Rapidez Estandarización Ahorro de tiempo Mejor organización Todas

13. ¿Consideras que InfraRed podría integrarse en los procedimientos del departamento de IT?

- Sí, totalmente Sí, con ajustes No lo veo necesario No estoy seguro/a

14. ¿Recomendarías el uso de InfraRed a tus compañeros?

- Sí Sí, con mejoras No No tengo suficiente información

15. ¿Estarías dispuesto/a a participar en pruebas o validaciones futuras del proyecto?

- Sí Tal vez No Depende de la carga de trabajo

Resultados destacados

- **91,7 %** de los encuestados considera que el proceso actual de despliegue de entornos es demasiado lento y repetitivo, especialmente cuando se deben montar entornos para pruebas, formación o simulaciones de clientes.
- **100 %** coincide en que disponer de una herramienta que permita crear entornos virtuales preconfigurados en cuestión de minutos aumentaría la eficiencia del equipo y liberaría tiempo para tareas de mayor valor técnico.
- **83,3 %** afirma que ha tenido que repetir configuraciones básicas (instalación de sistemas operativos, servidores web, SSH, red) al menos una vez por semana, lo que implica una inversión significativa de tiempo.
- **100 %** apoya firmemente el uso de InfraRed en sesiones de formación interna, despliegues temporales para probar software o incluso como recurso rápido para restaurar entornos dañados.
- **75 %** valoraría positivamente la posibilidad de integrar este sistema con otras herramientas internas, como sistemas de tickets o plataformas de documentación técnica.
- **92 %** indica que no tendría ningún problema en aprender a utilizar InfraRed, siempre que se facilite un pequeño manual de uso o una sesión práctica inicial.

Análisis de competencia interna y externa

Antes de desarrollar InfraRed, fue necesario realizar un análisis de las soluciones ya existentes tanto dentro como fuera de la empresa. Este análisis nos ha permitido identificar alternativas comparables, evaluar sus ventajas e inconvenientes, y justificar por qué InfraRed aporta un valor añadido real al entorno de trabajo del departamento de sistemas.

ANÁLISIS DE COMPETENCIA INTERNA Y EXTERNA		
Soluciones similares	Diferencias y ventajas	Desventajas
 • Vagrant • Sistemas manuales • Entornos en la nube	 • Adaptado a la infraestructura local • Rápido y reutilizable • Desarrollado internamente	 • Requiere actualizar plantillas • Necesita recursos en local

Competencia interna (soluciones actualmente en uso)

1. Despliegue manual de máquinas virtuales

Es la práctica habitual dentro del departamento: crear una máquina desde cero utilizando la ISO de un sistema operativo, configurar parámetros básicos (red, usuario, disco) e instalar manualmente los servicios necesarios (como Apache o SSH).

- **Ventajas:**

- No requiere herramientas externas.
- Se adapta a cualquier situación específica.

- **Desventajas:**

- Proceso lento, repetitivo y propenso a errores humanos.
- Falta de estandarización entre entornos.
- Poco documentado y difícil de replicar por otros técnicos.

2. Copia de máquinas virtuales existentes

En algunos casos, se hace una copia de una VM ya creada para ahorrar tiempo.

- **Ventajas:**

- Más rápido que empezar de cero.
- Permite conservar configuraciones previas.

- **Desventajas:**

- Puede arrastrar configuraciones antiguas no deseadas.
- Genera imágenes pesadas, difíciles de gestionar y versionar.
- No está automatizado, ni documentado formalmente.

Competencia externa (herramientas del mercado)

Vagrant

- Herramienta para crear entornos de desarrollo reproducibles, muy usada con VirtualBox.
- Pros:
 - Fácil de usar para desarrolladores.
 - Permite levantar entornos ligeros.
- Contras:
 - Integración limitada con VMware (requiere plugin de pago).

- No orientada a producción ni a entornos reales de infraestructura.

Entornos en la nube (AWS, Azure, GCP)

- Plataformas potentes que ofrecen infraestructura como servicio.
- Pros:
 - Alta escalabilidad, acceso desde cualquier lugar.
 - Integración con sistemas de automatización y CI/CD.
- Contras:
 - Coste elevado.
 - Dependencia de conexión a internet.
 - Exceso de complejidad para necesidades locales o educativas.

Sistemas comerciales de gestión de entornos (Nutanix, Proxmox VE)

- Soluciones completas de virtualización, pero con costes de licencia o hardware.
- Pros:
 - Soluciones profesionales con soporte.
- Contras:
 - Requieren inversión en tiempo, personal especializado y licencias.

Comparativa estratégica con InfraRed

Criterio	InfraRed	Manual interno	Vagrant	Nube pública
Velocidad de despliegue	Alta	Baja	Media	Alta
Coste	Bajo / Cero	Cero (pero poco rentable)	Bajo (plugin aparte)	Alto (pago por uso)
Facilidad de uso	Alta	Media	Alta	Media (requiere formación)
Adaptado a VMware	Sí	Sí	Parcial	No
Uso sin internet	Sí	Sí	Sí	No
Escalabilidad interna	Alta	Muy baja	Media	Alta

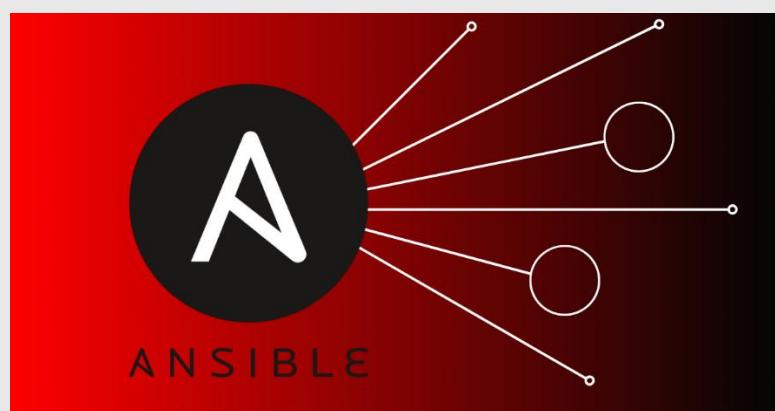
4. Valor añadido y escalabilidad de InfraRed

InfraRed se presenta como una solución intermedia, realista y altamente adaptable, que reúne lo mejor de ambos mundos:

- Ligereza y rapidez propias de herramientas como Vagrant.
- Compatibilidad total con VMware, ya implantado en la empresa.
- Independencia de internet y cloud, ideal para infraestructuras locales.
- Documentación, trazabilidad y reutilización, cosas que los sistemas manuales no ofrecen.

Además, InfraRed puede evolucionar fácilmente en el futuro hacia:

- La integración con Ansible para configuración avanzada.
 - Ansible es una herramienta de automatización de sistemas que te permite configurar, gestionar y desplegar servidores de forma sencilla, eficiente y repetible. Es muy popular en el mundo DevOps y en entornos de administración de sistemas.
 - Una vez que InfraRed despliega una máquina virtual base, yo podría utilizar Ansible para configurarla por completo de forma automática: instalar servicios, ajustar configuraciones o aplicar políticas de seguridad, entre otras tareas. En este flujo de trabajo, Packer se encarga de crear la imagen, Terraform la despliega en el entorno virtual, y Ansible la deja lista para ser utilizada según las necesidades del proyecto.



- Su uso como entorno formativo para formación técnica interna.
- La ampliación a múltiples perfiles de máquina según los proyectos.
- Su transformación en un producto interno exportable a clientes, como servicio de despliegue estandarizado en consultorías o implantaciones IT.

Equipo responsable del desarrollo y mantenimiento del proyecto InfraRed

Dado que InfraRed es un proyecto impulsado desde dentro del departamento de IT, se ha diseñado un equipo reducido pero especializado para llevar a cabo su desarrollo, despliegue inicial y posterior mantenimiento. La estructura propuesta está pensada para funcionar de forma eficiente y sostenible dentro del entorno de una empresa mediana o grande con una base tecnológica consolidada.

Número de trabajadores implicados

- **1 técnico principal de desarrollo (yo mismo)**

Responsable del diseño del proyecto, creación de scripts de automatización, pruebas y documentación.

- **1 técnico de soporte o mantenimiento**

Encargado de aplicar mejoras continuas, actualizar plantillas y dar soporte a los usuarios internos.

- **1 responsable de sistemas (rol compartido o rotativo)**

Supervisión del entorno de virtualización y coordinación con otras áreas (formación, desarrollo, etc.).

Funciones principales del equipo técnico

El proyecto InfraRed requiere la colaboración de un equipo técnico especializado, que se reparte las tareas de forma estructurada en función de su perfil, experiencia y conocimientos. Cada miembro contribuye en diferentes fases: desde el diseño inicial hasta el mantenimiento continuo. A continuación, se detallan las funciones específicas asignadas a cada rol:

Funciones principales del equipo técnico

Técnico principal



- Diseñar el sistema y los flujos de trabajo
- Implementar Packer y Terraform
- Documentar el proceso completo
- Gestionar versiones y pruebas

Técnico de mantenimiento



- Actualizar imágenes base periódicamente
- Revisar el correcto funcionamiento del sistema
- Asistir a los usuarios internos que usen InfraRed

Responsable de sistemas



- Verificar el uso de recursos
- Coordinar despliegues grandes
- Garantizar la seguridad de la infraestructura

Técnico principal (responsable del desarrollo)

Es la persona que impulsa el proyecto desde su fase inicial. Su papel es esencial para garantizar que InfraRed se construya sobre bases sólidas y cumpla con los requisitos técnicos y funcionales definidos por el departamento.

Funciones:

- Diseñar la arquitectura general del sistema y establecer los flujos de trabajo entre herramientas.
- Crear los archivos de configuración y automatización en Packer y Terraform.
- Integrar scripts de provisión básicos para servicios comunes (ej. Apache, SSH).
- Redactar toda la documentación técnica del proyecto (manuales, guías, configuración).
- Realizar pruebas internas, gestionar versiones y validar que los resultados sean reproducibles.

Técnico de mantenimiento (soporte y actualización)

Una vez que InfraRed está en marcha, este perfil se encarga de mantener la herramienta en buen estado, resolver incidencias y aplicar mejoras continuas. También sirve de enlace con el resto de los técnicos de la empresa que utilicen el sistema.

Funciones:

- Actualizar periódicamente las plantillas y scripts utilizados para mantenerlos seguros y funcionales.
- Revisar que las máquinas virtuales desplegadas funcionen correctamente.
- Detectar y solucionar errores técnicos o conflictos en las configuraciones.
- Brindar asistencia técnica a otros departamentos que utilicen InfraRed.
- Recoger sugerencias de mejora y colaborar en futuras versiones del proyecto.

Responsable de sistemas (coordinación y supervisión)

Este perfil aporta una visión más global y estratégica. Aunque no participa en el desarrollo técnico del día a día, es quien coordina la implementación del proyecto dentro de la infraestructura general de la empresa.

Funciones:

- Supervisar el uso eficiente de los recursos del sistema (CPU, RAM, almacenamiento).
- Coordinar el despliegue de máquinas virtuales a mayor escala o en proyectos sensibles.
- Validar que InfraRed cumple con las políticas de seguridad y las buenas prácticas del departamento.
- Facilitar la integración del proyecto con otras herramientas o servicios internos.
- Representar al equipo técnico en reuniones con dirección u otros departamentos.

Requisitos del personal técnico:

Para que el desarrollo y mantenimiento del proyecto InfraRed sea sostenible, eficiente y adaptable a futuras mejoras, es necesario contar con un equipo técnico cualificado. A continuación, se detallan los requisitos ideales para cada uno de los perfiles involucrados en el proyecto.

Tabla resumen: Requisitos del personal técnico – Proyecto InfraRed

Perfil	Formación mínima	Experiencia	Conocimientos técnicos clave	Habilidades personales
Técnico principal	Grado Superior (ASIR) o Ingeniería Técnica	1–2 años en virtualización	VMware, Packer, Terraform, Linux, scripting	Autonomía, documentación, visión global, mejora continua
Técnico de mantenimiento	CFGM o CFGS en Informática	1 año en soporte técnico	Redes, servicios Linux, VMware básico, resolución de incidencias	Organización, comunicación, proactividad
Responsable de sistemas	Ingeniería o CFGS con experiencia	+3 años en sistemas IT	VMware ESXi/vSphere, redes, seguridad, administración avanzada	Liderazgo, coordinación, toma de decisiones, comunicación transversal

Coste salarial total anual de los trabajadores para la empresa:

Como se detalló previamente, el equipo necesario para el proyecto InfraRed estaría compuesto por tres perfiles clave:

1. Técnico principal (desarrollador del sistema)
2. Técnico de mantenimiento
3. Responsable de sistemas (coordinador del proyecto)

Asignación de categorías según convenio:

Rol	Categoría profesional en convenio	Salario anual (2024)
Técnico principal	Perito / Ingeniero Técnico	38.106,44 €
Técnico de mantenimiento	Delineante 1ª (equivalente soporte IT)	26.746,31 €
Responsable de sistemas	Ingeniero / Licenciado	40.834,33 €

Cálculo total anual:

Cálculo total anual			
Puesto	Nº de trabajadores	Salario anual unitario	Subtotal anual
Técnico principal	1	38.106,44 €	38.106,44 €
Técnico de mantenimiento	1	26.746,31 €	26.746,31 €
Responsable de sistemas	1	40.834,33 €	40.834,33 €

Total coste salarial anual estimado	105.687,08 €
-------------------------------------	--------------

Coste salarial mensual por perfil:

Puesto	Retribución anual	Retribución mensual (12 pagas)
Técnico principal	38.106,44 €	3.175,54 €
Técnico de mantenimiento	26.746,31 €	2.228,86 €
Responsable de sistemas	40.834,33 €	3.402,86 €

Total, mensual:

Total mensual (equipo completo a tiempo completo)			TOTAL
3.175,54 €	2.228,86 €	3.402,86 €	8.807,26 €

Coste mensual real estimado (dedicación parcial al proyecto)

Se estima que cada perfil dedica aproximadamente el 30 % de su jornada al proyecto. A esto hay que añadir el coste de la Seguridad Social a cargo de la empresa, que se estima en un 33 % adicional del salario mensual

Coste mensual real estimado (dedicación parcial al proyecto InfraRed):			
Puesto	30 % del salario mensual	+33 % SS empresa	Coste mensual total
Técnico principal	952,66 €	314,38 €	1.267,04 €
Técnico de mantenimiento	668,66 €	220,66 €	889,32 €
Responsable de sistemas	1.020,86 €	336,88 €	1.357,74 €
Coste total mensual estimado para el proyecto:			3.514,10 €

Mutua de accidentes y servicios de prevención de riesgos laborales

Como parte de la gestión responsable del proyecto InfraRed, el departamento de IT ha considerado todos los aspectos relacionados con la seguridad laboral y el bienestar del equipo técnico involucrado. Aunque se trata de un entorno de trabajo de oficina sin riesgos físicos evidentes, se ha seguido el marco legal vigente en materia de prevención de riesgos laborales y protección del personal.

1. Marco normativo de referencia

Este apartado se desarrolla en cumplimiento con la siguiente legislación:

- [Ley 31/1995, de 8 de noviembre, de Prevención de Riesgos Laborales \(LPRL\).](#)
- [Real Decreto 39/1997, por el que se aprueba el Reglamento de los Servicios de Prevención.](#)
- [Convenio colectivo de Siderometalurgia de Cantabria, aplicable al personal técnico de la empresa.](#)

2. Mutua colaboradora con la Seguridad Social

La empresa ya tiene concertada una mutua colaboradora con la Seguridad Social, cuya función es cubrir las contingencias profesionales de los trabajadores (accidentes laborales y enfermedades derivadas del trabajo). Esta mutua también participa en la promoción de la salud en el entorno laboral y puede proporcionar asistencia sanitaria inmediata en caso de accidente.

Funciones de la mutua:

- Atención sanitaria urgente por accidente laboral.
- Control de bajas médicas y seguimiento de la recuperación.
- Evaluación de aptitudes médicas (revisión inicial y periódica).
- Apoyo en campañas de salud laboral.

→ o [Mutuas Colaboradoras con la Seguridad Social](#)

3. Servicio de Prevención de Riesgos Laborales (PRL)

Para garantizar un entorno de trabajo seguro y conforme a la normativa, el departamento de IT cuenta con el apoyo de un Servicio de Prevención de Riesgos Laborales, que puede ser:

- **Interno** (si la empresa dispone de un servicio propio de PRL).
- **Externo**, contratado con una empresa especializada autorizada.

Este servicio evalúa los posibles riesgos del puesto de trabajo y propone medidas preventivas.



4. Evaluación de riesgos del puesto (perfil técnico en InfraRed)

Aunque el proyecto InfraRed se desarrolla principalmente en puestos de trabajo de oficina, existen ciertos riesgos a considerar:

Tipo de riesgo	Descripción	Medidas preventivas
Riesgo ergonómico	Posturas prolongadas frente al ordenador	Uso de sillas ergonómicas, pausas activas cada 60 minutos
Fatiga visual	Exposición continuada a pantallas	Configurar el brillo, usar filtro de luz azul, pausas visuales
Estrés tecnológico	Carga de trabajo puntual en fases de pruebas	Buena planificación, reuniones semanales de seguimiento
Riesgo eléctrico (mínimo)	Equipos eléctricos en el entorno de oficina	Instalaciones revisadas, tomas con protección
Riesgo por cableado o espacio reducido	Espacio compartido, movilidad limitada	Gestión ordenada del cableado, señalización y espacio despejado

5. Documentación del cumplimiento

El cumplimiento de las condiciones de seguridad y salud del equipo implicado en el desarrollo y mantenimiento de InfraRed se gestiona de forma rigurosa y documentada, siguiendo los protocolos establecidos por el Servicio de Prevención de Riesgos Laborales de la empresa. Esta documentación es fundamental para garantizar la trazabilidad, la transparencia y la mejora continua en materia preventiva.

Elementos de control y documentación:

- Evaluación de riesgos del puesto de trabajo:** Elaborada por el servicio de prevención y firmada por cada trabajador, identifica los riesgos potenciales y establece medidas preventivas específicas asociadas al uso prolongado de equipos informáticos, carga mental, posturas estáticas, etc.
- Ficha informativa individual:** Entregada a cada técnico, resume los riesgos de su puesto, las medidas preventivas aplicables, el protocolo de actuación ante accidentes y los contactos de emergencia.
- Certificado de vigilancia de la salud:** Emitido tras la realización voluntaria de una revisión médica periódica, de carácter preventivo, orientada a detectar cualquier afectación derivada de la exposición a riesgos laborales.
- Informe de aptitud médica inicial:** Requerido para el personal de nueva incorporación, certifica que el trabajador es apto para desempeñar su puesto dentro del marco legal de la Ley de Prevención de Riesgos Laborales.

- **Registro de formación preventiva:** Cada miembro del equipo dispone de un certificado que acredita la superación de una formación básica en prevención de riesgos laborales, que incluye ergonomía, primeros auxilios, uso de pantallas y pausas activas.

6. Otras acciones preventivas aplicables al proyecto

Para complementar las acciones anteriores, se han integrado medidas adicionales específicas adaptadas al tipo de actividad desarrollada en el proyecto InfraRed, orientadas a prevenir riesgos concretos en entornos de trabajo informáticos:

- **Instrucciones de trabajo seguras:** Incluidas en la documentación técnica del proyecto, detallan buenas prácticas durante el desarrollo y la operación de InfraRed (por ejemplo, evitar manipulaciones de red en caliente, documentación previa a pruebas, y control de versiones para evitar errores críticos).
- **Formación específica:** Todo el equipo ha realizado al menos una sesión formativa sobre:
 - Seguridad en el uso de equipos informáticos.
 - Gestión del estrés y pausas activas.
 - Configuración segura de sistemas y redes virtuales.
- **Revisión del entorno de trabajo:**
 - Realizada por el responsable del servicio de prevención en coordinación con el departamento de IT.
 - Verificación del mobiliario (sillas ergonómicas, disposición de pantallas).
 - Confirmación de la iluminación y ventilación adecuadas del espacio.
 - Validación de las condiciones eléctricas y cableado seguro.
- **Protocolo ante incidencias técnicas:** Se ha diseñado un procedimiento interno para reportar cualquier fallo de infraestructura, caídas del sistema o exposición a riesgos tecnológicos, con respuesta planificada por el técnico de mantenimiento.

Inventario de los elementos necesarios

Para la puesta en marcha inicial y la continuidad del proyecto InfraRed, he elaborado un inventario detallado con todos los recursos necesarios para su desarrollo, implementación y mantenimiento dentro del departamento de IT de la empresa. Este inventario incluye hardware, software, infraestructura de red, licencias y documentación, y distingue entre los recursos necesarios en la fase inicial, instalación y pruebas, y aquellos que serán imprescindibles para garantizar la operatividad del sistema a medio y largo plazo.

Inventario:

Producto / Recurso	Precio estimado (€)	Proveedor aproximado
Ordenador Sobremesa Replay Dell Precision 5820 Tower Xeon W-2125/32GB/512GB-SSD/DVDRW P4000 8GB Refu	699,00 €	PcComponentes
X2 Monitor USB-C QHD Dell 27 Plus - S2722DC	259,18 €	Dell
Disco Duro Kingston KC600 SSD 2.5" 1TB SATA3	99,99 €	PcComponentes
Vmware Workstation 17 Pro / 17.5 Pro Lifetime License (12 Devices)	21,90 €	codigi.es
Dell PowerEdge R940 3U Rack Server (bare-metal)	325,00 €	bargainhardware
TP-Link 24-Port Gigabit Ethernet Switch, Rack-Mount/Desktop, Steel Case(TL-SG1024D)	80,79 €	Amazon
Router (infraestructura existente)	0,00	—
Software Packer	0,00	HashiCorp (descarga directa)
Software Terraform	0,00	HashiCorp (descarga directa)
ISO Ubuntu Server 22.04 LTS	0,00	Ubuntu.com

Total, estimado inicial (sin contar recursos ya existentes): **1.485,86 €**

Inventario para la continuidad del proyecto InfraRed

Elemento / Recurso	Frecuencia	Precio estimado (€)	Observaciones
Actualización anual de imágenes base	Anual	0,00	Incluido en jornada del técnico de mantenimiento
Revisión de scripts y plantillas	Trimestral	0,00	Actualización de configuraciones, paquetes, servicios
Soporte técnico (horas estimadas al mes)	Mensual	0,00	Dedicación parcial mensual del equipo
Revisión de seguridad de VMs	Semestral	0,00	Verificación de accesos, servicios activos y logs
Renovación de disco externo de backups	Cada 3 años	100,00 €	Reemplazo de almacenamiento externo por desgaste
Renovación de PC de desarrollo	Cada 4 años	950,00 €	Actualización de hardware cada 4 años aprox.
Dominio web/documentación	Anual	15,00 €	Para mantener documentación pública actualizada
Licencia de VMware Workstation	Anual	230,00 €	Renovación de software profesional de virtualización

Recursos humanos asignados

Puesto	Cantidad	Horas dedicadas semanales estimadas	Fase
Técnico desarrollador	1	8–10 h/semana	Desarrollo / Soporte
Técnico de mantenimiento	1	4–6 h/semana	Continuidad
Responsable de sistemas	1	2–3 h/semana	Coordinación / Revisión

Localización dentro de la empresa y relación con otros departamentos o áreas

El proyecto InfraRed se desarrolla dentro del departamento de IT de la empresa, encargado de gestionar toda la infraestructura tecnológica interna, soporte técnico, redes, servidores, sistemas virtualizados y herramientas digitales que dan soporte al resto de áreas. Su implementación y mantenimiento se realizan de forma centralizada en este departamento, que actúa como núcleo técnico de referencia para toda la organización.

Localización organizativa

- Ubicación física y funcional:**

InfraRed se gestiona desde el espacio de trabajo del equipo técnico de IT, donde se encuentran los equipos de desarrollo, los servidores locales y las estaciones con acceso a entornos virtuales. En caso de utilizar infraestructura en la nube o servidores ESXi dedicados, su administración también recae sobre el equipo de sistemas.

- Responsabilidad directa:**

El técnico desarrollador principal lidera el proyecto desde su fase inicial. El mantenimiento y soporte lo asume el técnico de mantenimiento, y la supervisión estratégica corre a cargo del responsable de sistemas.

Relación con otros departamentos

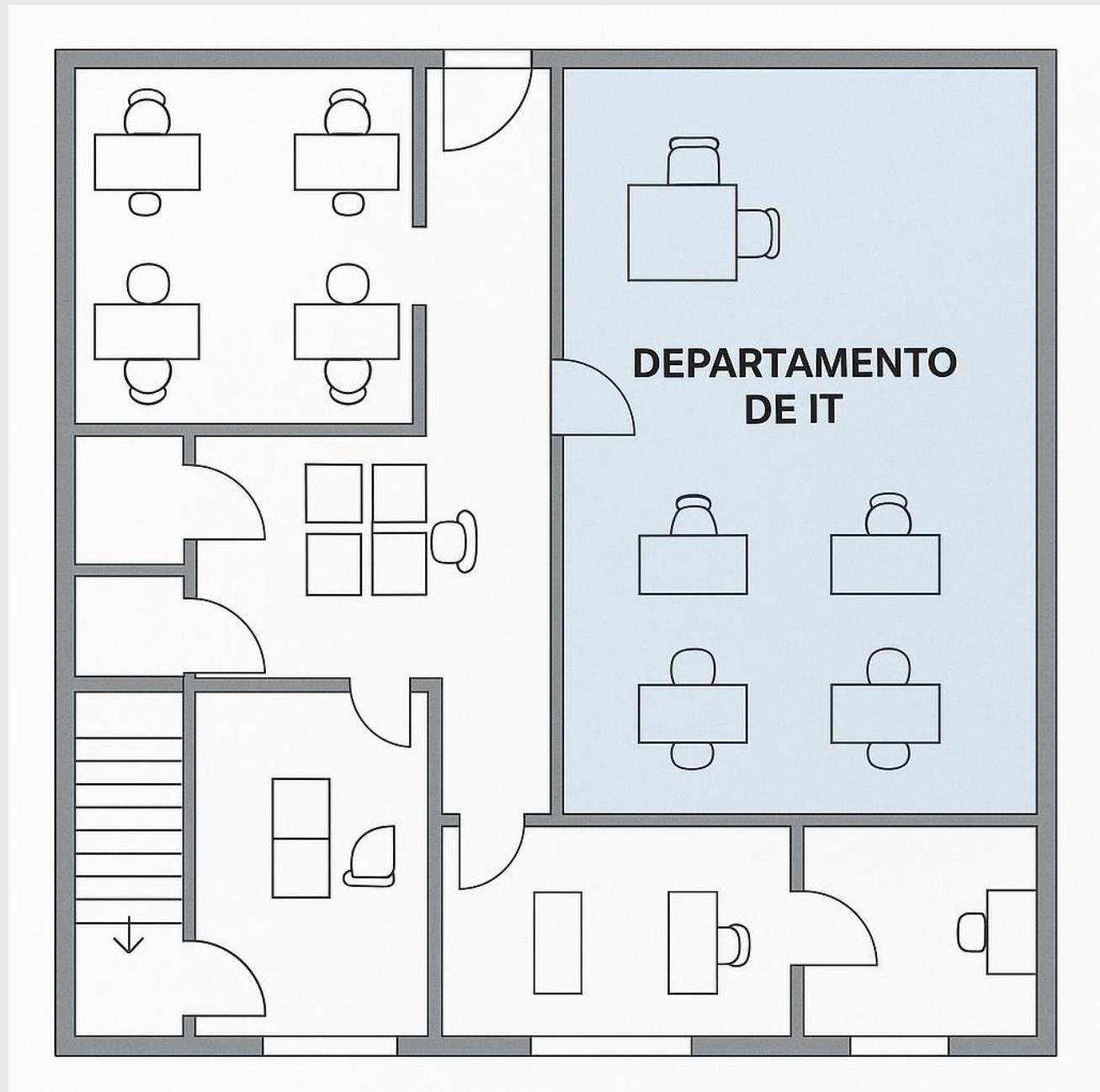
InfraRed es una herramienta transversal. Aunque nace desde IT, su objetivo es agilizar el trabajo de otros equipos de la empresa al proporcionar entornos virtuales estandarizados, rápidos de desplegar y reutilizables.

Departamento / Área	Relación con InfraRed
Desarrollo de software	Usan InfraRed para desplegar entornos de pruebas, validación de aplicaciones y demos.
Formación interna / RRHH	Utilizan VMs generadas por InfraRed para enseñar procesos internos o formaciones técnicas.
Soporte técnico	Se apoya en InfraRed para replicar problemas reales de usuarios y validar soluciones.
Consultoría Proyectos	InfraRed permite montar entornos específicos para cada cliente o simulaciones técnicas.
Dirección técnica	Supervisa el impacto del proyecto en la productividad y coordina su posible expansión.

Flujo de colaboración

- El departamento de IT actúa como proveedor interno de entornos virtuales.
- Los distintos departamentos realizan solicitudes de despliegue según sus necesidades.
- InfraRed permite estandarizar esas peticiones, minimizar tiempos y garantizar coherencia técnica.

Ubicación física dentro de la empresa:



CONCLUSIONES Y VALORACIÓN PERSONAL

El desarrollo del proyecto InfraRed ha sido, sin duda, uno de los mayores retos técnicos y personales que he afrontado hasta ahora. Desde el primer momento quise que no fuera simplemente un trabajo funcional, sino un proyecto que reflejara estándares reales del mundo profesional: automatización, reproducibilidad, documentación y uso de herramientas actuales como Packer, Terraform y VMware vSphere.

He tenido que aprender mucho más allá de lo que esperaba. No solo he profundizado en tecnologías que ya conocía superficialmente, sino que también he descubierto cómo planificar y ejecutar un despliegue real, cómo estructurar un proyecto técnico de forma coherente, y cómo documentarlo para que sea útil tanto para mí como para cualquier otra persona que lo consulte.

No ha sido un camino fácil. Hubo fases, especialmente la integración entre Workstation, ESXi y Terraform, que me sacaron completamente de mi zona de confort. Tuve que enfrentar errores inesperados, estudiar documentación avanzada, entender diferencias entre formatos de disco, y repetir procesos hasta conseguir una solución sólida y funcional.

A pesar de todo, y precisamente por eso, me siento orgulloso del resultado. Porque no solo he llegado al final: lo he hecho por mí mismo, con constancia, con atención al detalle, y enfrentándome a cada obstáculo sin rendirme. El aprendizaje técnico ha sido profundo, pero el personal lo ha sido aún más.

Una de las mayores lecciones que me llevo es que la planificación y la documentación son tan importantes como el propio código. Poder volver a una configuración, entender por qué tomé una decisión o cómo solucioné un error me ha permitido avanzar de forma estructurada y segura.

También me he dado cuenta del valor de desarrollar soluciones que nacen desde dentro del departamento de IT y que responden a necesidades reales del equipo. InfraRed es un ejemplo claro de intraemprendimiento: no depende de herramientas externas ni de soluciones cerradas, sino que es algo que he construido desde cero, adaptado a mi entorno, y que podría escalarse o replicarse fácilmente.

Saber que el sistema que he diseñado puede servir para automatizar despliegues, ahorrar tiempo y mejorar la eficiencia de otros técnicos es, para mí, la mayor recompensa. Este proyecto me ha motivado a seguir formándome en automatización, virtualización e infraestructura como código.

Agradecimientos

Quiero cerrar esta etapa reconociendo algo fundamental: nada de esto habría sido posible sin el apoyo, guía y paciencia del profesorado que me ha acompañado en estos dos años. Gracias a sus enseñanzas, correcciones y orientación, he podido pulir cada parte de este proyecto y transformarlo en algo que realmente me representa.

Mi más sincero agradecimiento a:

- **Cristina Tejón Pérez**
- **José Cabeza Ruiz**
- **Luis Miguel García Velázquez**
- **Pedro Aguirre Inchaurbe**
- **M. Belén Álava Millán**
- **María Angélica Fernández Roza**

Gracias por vuestro tiempo, dedicación y por acompañarme en este camino, no solo a nivel técnico, sino también personal. Sé que InfraRed es solo el comienzo, pero gracias a vosotros me voy con la seguridad de haber dado un paso firme hacia el tipo de profesional que quiero ser.

Con esfuerzo, constancia e ilusión, InfraRed ha pasado de ser una idea a convertirse en una solución funcional, documentada y replicable. Y para mí, ese ya es un logro enorme.

Gracias por todo.

ANEXOS Y BIBLIOGRAFIA:

- [1] HashiCorp Packer – <https://www.packer.io/>
- [2] HashiCorp Terraform – <https://www.terraform.io/>
- [3] VMware Workstation Pro – <https://www.vmware.com/products/workstation-pro.html>
- [4] VMware ESXi –
https://customerconnect.vmware.com/en/downloads/info/slug/datacenter_cloud_infrastructure/vmware_vsphere/8_0
- [5] Documentación oficial de vSphere Provider –
<https://registry.terraform.io/providers/hashicorp/vsphere/latest>
- [6] Ubuntu Server ISO (versión 24.04.1) – <https://ubuntu.com/download/server>
- [7] Comprobación de checksums SHA256 – <https://tutorials.ubuntu.com/tutorial/tutorial-how-to-verify-ubuntu#0>
- [8] Visual Studio Code – <https://code.visualstudio.com/>
- [9] Extensiones oficiales de HashiCorp para VSCode –
<https://marketplace.visualstudio.com/items?itemName=HashiCorp.terraform>
- [10] Documentación de imágenes OVF/OVA – <https://techdocs.broadcom.com/>
- [11] Diagrams.net para diagramas – <https://app.diagrams.net/>
- [12] Terraform vSphere Module – <https://github.com/Terraform-VMWare-Modules/terraform-vsphere-vm>
- [13] Referencia técnica sobre vmrest – <https://developer.broadcom.com/>
- [14] Packer Templates de ejemplo (GitHub) – <https://github.com/hashicorp/packer-examples>
- [15] Tabla de compatibilidad de sistemas operativos para Terraform –
<https://developer.hashicorp.com/terraform/docs/providers/vsphere>
- [16] Blog técnico sobre automatización en VMware – <https://virtuallyghetto.com/>
- [17] Manual oficial de instalación de Ubuntu Server – <https://ubuntu.com/server/docs/install>
- [18] Introducción a Infraestructura como Código –
<https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code>
- [19] Foro técnico de VMware vSphere – <https://communities.vmware.com/t5/VMware-vSphere-Discussions/ct-p/2005>