Universitat Oberta de Catalunya

uoc.edu

Infraestructura de Seguridad en la nube de Azure

Autor: Sergio López López
Grado en Ingeniería Informática
Seguridad Informática

Consultor: Jorge Miguel Moneo
Profesora responsable: Helena Rifà Pous

Fecha Entrega: enero de 2023

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Infraestructura de Seguridad en Azure*
Nombre del autor:	*Sergio López López*
Nombre del consultor/a:	*Jorge Miguel Moneo*
Nombre del PRA:	*Helena Rifà Pous*
Fecha de entrega (mm/aaaa):	*01/23*
Titulación o programa:	Grado en ingeniería informática
Área del Trabajo Final:	*Seguridad informática*
Idioma del trabajo:	*Castellano*
Palabras clave	*Nube, Azure, Herramientas*

Resumen del Trabajo

Actualmente la transformación y migración digital hacía un paradigma *cloud* es más evidente, las ventajas motivan a los diferentes negocios y organizaciones a mover y mantener su infraestructura en la nube. Sin embargo, existen puntos clave que hay que analizar para securizarla al máximo.

En muchas ocasiones, la Seguridad del objeto la delegamos al proveedor seleccionado, y en gran medida es correcto según la tipología de servicio que se ha seleccionado, pero aun así, el cliente debe conocer en qué aspectos es responsable y capaz de proteger.

Para ello, en este Trabajo de fin de Grado se pretende enumerar, describir y mitigar o solventar la exposición a estos puntos débiles que el cliente es responsable y como automatizarlo, basándonos en la nube de Microsoft, Azure. Pese a ser en muchos casos extrapolable a otras nubes públicas.

Abstract

Currently the digital transformation and migration to a cloud paradigm is more evident, the advantages motivate different businesses and organizations to move and maintain their infrastructure in the cloud. However, there are key points to analyze in order to secure it to the maximum.

On many occasions, the security of the object is delegated to the selected provider, and to a large extent it is correct depending on the type of service selected, but even

so, the client must know which points he is responsible for and able to protect.

For this, in this Final Degree Project we intend to list, describe and mitigate or solve the exposure to these weaknesses that the customer is responsible and how to automate, based on the Microsoft cloud, Azure. Despite being in many cases can be extrapolated to other public clouds.

Índice

Lista de Ilustraciones

Lista de tablas

1. Introducción

1.1. Contexto y justificación del Trabajo

Durante estos últimos años, según un reporte de Flexera, el 57% de las organizaciones han ido adoptando una transformación y migración digital de su carga de trabajo hacia la nube (*cloud*) [1], este proceso, en ocasiones viene guiado por el proveedor seleccionado, pero suele ser generalizado dada las diferentes dimensiones y características de cada organización. Por ello, durante este proceso de cambio, características y metodologías que se venían haciendo en entornos locales *on-premise* disrumpen con los servicios *cloud*, por lo que deben ser adaptadas a la arquitectura elegida, configurándose debidamente.

Este cambio se origina dada la necesidad de migración de su infraestructura y aplicaciones ante la creciente exigencia en el mercado. Las ventajas que alcanzan dichas organizaciones al dar el gran paso son las siguientes, un control de costes determinante, un mantenimiento de infraestructura más cómodo, obtención de alta disponibilidad, elasticidad y mayor tolerancia de fallos, entre otras.

Además, en muchas ocasiones, los primeros pasos hacia esta transformación, aun siendo claves, son los menos revisados dada la inexperiencia al cambio. Es decir, analizar el estado actual, reconocer el tipo de nube requerida (pública, privada, híbrida o *multicloud*), seleccionar proveedor *cloud* (Microsoft Azure, Amazon Web Service, Google Cloud, Alibaba Cloud, IBM, Salesforce o Oracle, entre otros [2]) y reconocer qué tipo de servicio es el más adecuado para el sistema –que será *legacy*– de la organización (SaaS, PaaS o IaaS) con las grandes diferencias, no tan solo funcionales, si no de responsabilidad en la seguridad a nivel del cliente que deba conocer y proteger.

Este proyecto se enfoca únicamente en un proveedor cloud –aunque puedan existir comparaciones y extrapolaciones entre otros similares– para centrar el campo de estudio al máximo. En este caso, se ha seleccionado a Microsoft® Azure, el segundo proveedor más utilizado del mundo según un estudio de mercado ofrecido por *Statista* [3].

Por ello, esta investigación analizará los puntos básicos de seguridad que detalla el proveedor Azure, como y porqué implementarlos y posteriormente se creará una herramienta automatizada que la organización pueda desplegar para encontrar, mitigar e intentar solventar las carencias dentro de la nube de Azure.

1.2. Motivación

Este trabajo se realiza con una motivación de aplicar el conocimiento sobre la creación y mantenimiento de infraestructura alojada en Azure aportando un valor extra centrándose en el ámbito de la seguridad informática, con la finalidad de crear una pequeña guía frente a la creciente acogida de las tecnologías *cloud*.

De tal manera, aun siendo una investigación enfocada en Azure, este aporte puede trasladarse a múltiples proveedores escogidos por distintas organizaciones, tal y como se ha mencionado en la introducción, con la finalidad de hacer frente los posibles agujeros de seguridad encontrados paso a paso.

Además, se añadirá como elemento principal una herramienta automatizada, nombrada como **Az Infra Sec** para permitir ser usada, por un simple *click & run* para las organizaciones que necesiten poder enfrentarse de manera sistemática, dado su negocio, a las posibles incidencias de seguridad en su día a día.

1.3. Objetivos del Trabajo

1.3.1 Objetivo General

El objetivo general del trabajo está enfocado en implementar una serie de mecanismos básicos para proteger y securizar la infraestructura y datos ubicados en Azure de una organización genérica.

1.3.2 Objetivos específicos

Seguidamente, se detallan los objetivos específicos que se deberán trabajar:

- Analizar los servicios que propiamente provee el proveedor (Microsoft), estudiarlos y ver cómo se pueden implementar y hacer uso de los mismos siguiendo los manuales y la documentación oficial, paso a paso.
- Aplicar una implementación basada en buenas prácticas y configuraciones acorde a una carga de trabajo de una organización ficticia.
- Automatizar el paso anterior, es decir, de forma que se realice de manera más óptima y a su vez se eviten errores humanos. Para ello, se crearán una serie de scripts diseñados para cada finalidad analizada.
- Centrar todos los scripts creados en una herramienta integral, ejecutable mediante una terminal en Windows, que deberá verificar y reportar cada punto débil o amenaza investigada y crear los servicios adecuados siguiendo una serie de plantillas siguiendo una buena práctica de configuración.

Como objetivo definitivo dotar finalmente de la herramienta **Az Infra Sec** que pueda ser usado por el personal de la organización que dado el análisis previo de los servicios y la implementación manual de los mismos se pueda abstraer la interacción del usuario en la mayor medida posible automatizando las tareas de seguridad en la infraestructura en Azure.

1.4. Metodología del Trabajo

Puesto que finalmente el objetivo es crear una herramienta automatizada para tratar los elementos básicos de seguridad a tener en cuenta en la infraestructura de Azure, se adoptará una metodología Agile, con lo que existirá una mejora incremental en cada iteración dentro de todo el Sprint antes de la entrega final.

Para llevar a cabo este seguimiento utilizaremos un Backlog, estilo Kanban –en el que se representan de forma visual las tareas a realizar, con un peso y estado concreto, por ejemplo, *TO-DO*– ofrecido por Azure DevOps en el que podemos separar las temáticas por Epics, dentro de estas por Features y finalmente por Tasks simples, de esta manera la ejecución del trabajo final será completada de forma ordenada:

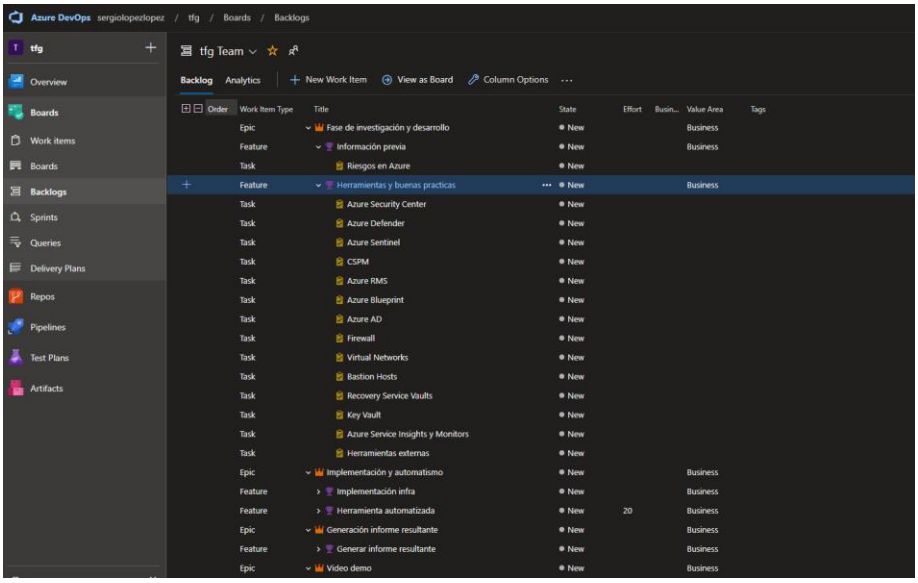


Ilustración 1: Organización Tareas

Primera iteración: Definición del plan de trabajo
Conocer los diferentes servicios que nos provee Azure para las organizaciones y sus propias herramientas de seguridad. Así como la gobernanza y gestión de usuarios para encontrar el punto inicial del que tratar.

Segunda iteración: Listado de malas prácticas
Siguiendo, por un lado, estadísticas reales de problemas de seguridad en el ámbito *cloud* y por otro, configuraciones por defecto que los usuarios pueden adoptar, listar y debatir su problemática.

Tercera iteración: Listado de mecanismos de seguridad
Indicar y analizar los distintos servicios internos que provee Azure y como implementarlos paso a paso para solventar los *insights* que aparezcan. Además, se iniciará el proceso de automatización de dichas implementaciones.

Cuarta iteración: Analizar posibles ataques
Simular dentro de lo posible ataques potenciales o fallas de seguridad, originadas por una mala *praxis* en las configuraciones dentro de los servicios de Azure.

Quinta iteración: Configurar una herramienta automatizada
Analizando los puntos anteriores, realizar una herramienta integral, es decir, “todo-en-uno” en *Powershell*, que los indique e incluso mitigue o solvente.

1.5. Listado de tareas

Para la realización de este trabajo partimos de varias entregas que conjuntamente forman el proyecto completo, en cada entrega se dividirán las tareas en función del avance del mismo.

PEC1:
Se planificará el conjunto del proyecto de manera sintetizada con la finalidad de esquematizar los puntos que se deberán realizar en siguientes presentaciones hasta el trabajo completo.

Para ello, se estudiará el estado actual del contexto a investigar, es decir, el estado del arte, y que dirección se deberá seguir para aportar nuevos datos. Y, a partir de ese punto, se crean los apartados más introductorios.

PEC2:
Durante esta entrega, se desarrollará la introducción a la investigación, es decir, se pondrá en contexto de los servicios más habituales de la nube de Azure y se analizarán los mecanismos que se documentan junto con las buenas prácticas que se deben realizar.

PEC3:
En esta siguiente entrega, el proyecto se centrará en el gran punto del objetivo de la investigación, la creación y uso de la herramienta para automatizar los puntos analizados anteriormente.

PEC4:
Seguidamente, se cerrarán puntos como las conclusiones y posibles mejoras que no se han conseguido realizar. Además, se completará el proyecto escrito que deberá entregarse.

PEC5:
Y finalmente, se ejecutará la preparación del video de presentación y su grabación para ser entregado.

1.6. Planificación temporal

Los hitos principales que se han tenido en cuenta durante la realización del proyecto son los siguientes:

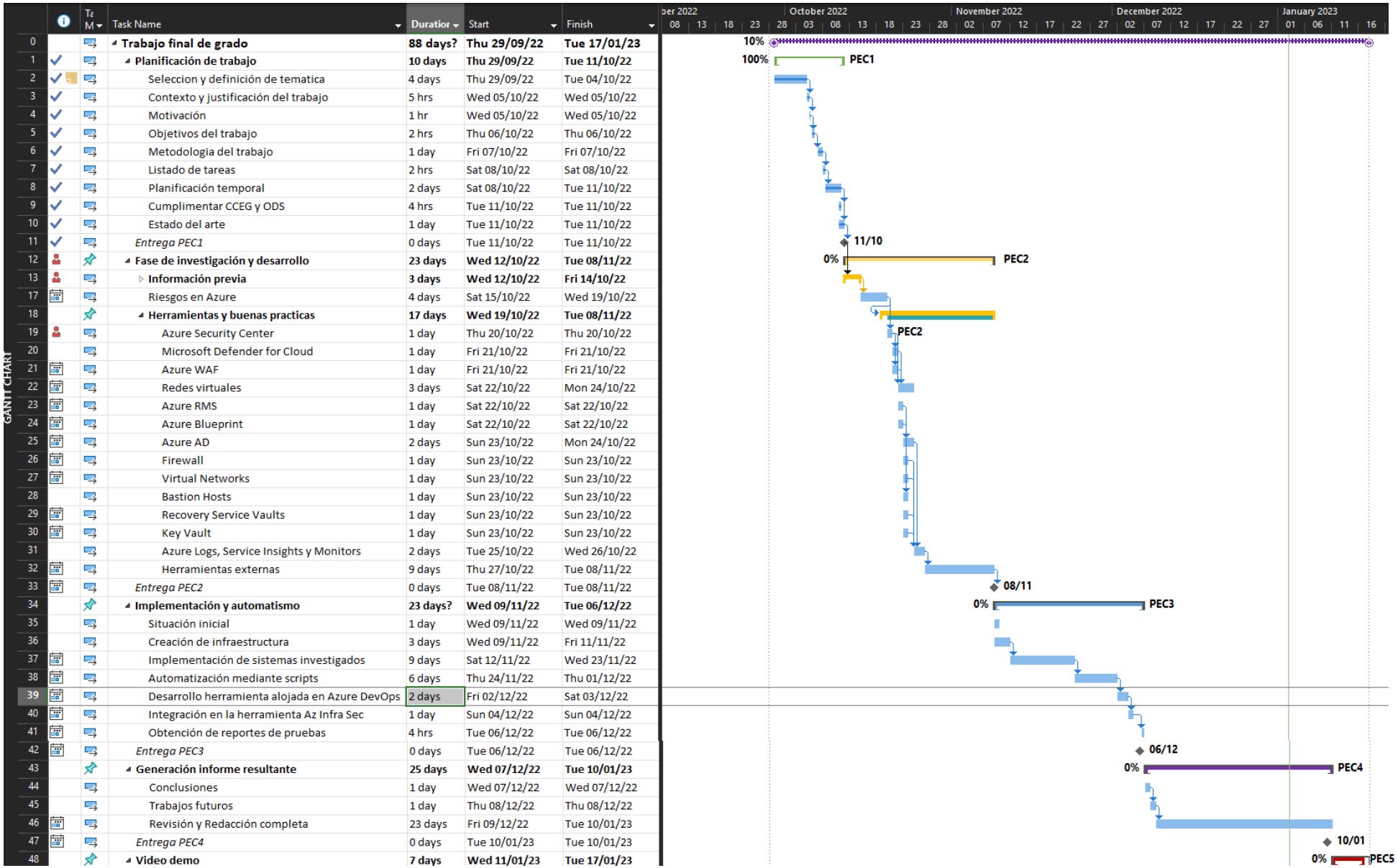


Ilustración 2: Planificación temporal

1.7. Competencia de compromiso ético y global (CCEG) y objetivos de Desarrollo Sostenible (ODS)

Durante este trabajo de fin de grado se dota de importancia una competencia y ética global en dónde se acompaña paralelamente de forma honesta, ética, sostenible, socialmente responsable y respetuosa con los derechos humanos y la diversidad diferenciadas en tres grandes dimensiones, UOC [9]:

- **Sostenibilidad**

  El objetivo final de este trabajo es mejorar la infraestructura que muchas organizaciones alojan en Azure proporcionando mayor seguridad para evitar el uso mal intencionado de la misma y en consecuencia el desgaste en recursos ante un incidente. De hecho, está totalmente relacionada dentro del objetivo 9 del desarrollo sostenible (ODS), Construir infraestructuras resilientes, promover la industrialización inclusiva y sostenible y fomentar la innovación.

  Por otro lado, no hay que obviar que cada vez que se añaden más procesos y servicios alojados al *cloud* se están haciendo mayor uso de energía. De manera que puede existir una mayor huella ecológica frente a no implementar la propuesta.

  Entonces, observando el impacto provocado en la industria cuando es atacada deliberadamente o existiendo un error humano que la comprometa, el coste energético es superior a los mecanismos preventivos que se pretenden impulsar durante este proyecto.

- **Comportamiento ético y de responsabilidad social (RS)**

  Para llevarlo a cabo, hay que tener claro el concepto de responsabilidad, se necesitan una serie de permisos elevados sobre la infraestructura que alojarán aplicaciones o almacenarán datos de varios de usuarios.

  Por lo tanto, se debe abarcar de manera estrecha un marco legislativo que proteja al usuario, bajo las diversas leyes de protección de datos incluidas en la LOPD y reglamentos afines.

  En la otra cara, existe el objetivo de automatizar en la mayor medida posible las tareas continuas o repetitivas que suelen gestionar los administradores u operadores de infraestructura. Por lo que, para favorecer el objetivo 8 del ODS, Trabajo decente y crecimiento económico, estas tareas automatizadas pueden ser un pequeño impedimento para el crecimiento del trabajador.

  Para poder mitigarlo al máximo, la organización tiene que dotar de otras tareas menos automatizables y creativas al equipo de operaciones. De cualquier forma, una tarea automatizada tiene que estar supervisada por algún responsable.

- **Diversidad, género y derechos humanos**
  En este punto, realmente este trabajo no afecta en absoluto, el protagonista en todo caso son las maquinas (ordenadores personales y servidores externos alojados en la nube). No existe por eso ni mujer, ni hombre, ni raza, ni cualquier otra “distinción”.

  Si, por lo contrario, el enfoque es sobre los datos que pueden contener estos servidores, la protección del mismo no distingue el dato en bruto que pueda ser intento de ataque.

1.8. Estado del arte

Existen diversos autores [10-13] que han trabajado sobre esta temática, la seguridad en Azure es muy importante debida a la gran aceptación del *cloud* en nuestra actualidad y en especial al servicio procedente de una gran empresa como Microsoft.

Por lo que, organizar el proyecto a partir de estudios ya realizados es una tarea necesaria con el fin de aportar nuevos enfoques –con la automatización– y con las irrupciones temporales que se van incorporando y modificando métodos de securización poco a poco deprecados.

Se revisarán las comparaciones entre distintos *clouds* que muestran los siguientes *papers*, *A review on AWS, Azure y GCP services* de T. Mufti, P. Mittal y B. Gupta [10] y Review Paper on Cloud Service Provider – AWS, AZURE, GCP de A. Yevge, P. Ghag, C. Solanki y A. Mishra [11].

Para escoger algunos de los estudios más completos y a la vez interesantes se dispondrá del libro de Karl Ots, *Azure Security Handbook* [12], donde se puede analizar el enfoque o guía que detalla para distintas situaciones.

Y revisará también, la gestión de la gobernanza y seguridad en Azure de P. Tender, D. Rendon y S. Erskine [13] donde detallan los distintos mecanismos que provee Azure para protegerse, como son los *Blueprints, Security Center* y *Sentinel*.

Por otro lado, se consultará detalladamente la propia documentación de Microsoft, dado que disponen de varias wikis para los distintos servicios que provee, y de la misma forma, como utilizar la API que los gestiona (para interactuar con Powershell) de la misma manera que lo hace el usuario desde el sitio web, *portal.azure.com* (la interfaz gráfica).

2. Riesgos en la nube de Azure

2.1. Conceptos básicos

Antes de entrar en detalle con los riesgos asociados a la nube pública, se describirán los conceptos básicos sobre que es la nube y sus características más destacables.

La nube (o *cloud*) en su forma más básica de expresión hace referencia a los servidores ubicados en centros específicos (*datacenters*) y accesibles por internet y al software y bases de datos ejecutados en ellos. Permitiendo a los usuarios su acceso desde cualquier ubicación y dispositivo.

Para la existencia de estos servidores cobra especial importancia la virtualización. Que, dicho de forma rápida, se estará creando un ordenador virtual simulando un ordenador físico con su propio hardware, pudiendo gestionarse de manera mucho más eficaz.

Por lo que, partiendo de la idea de estos servidores alojados en la nube, es decir, de esta computación en la nube (*cloud computing*) existen una gama de servicios dependiendo de las necesidades del usuario. Cuya distinción será muy importante a la hora de analizar los riesgos asociados de cada uno, estos son:

- **Infraestructura como servicio** (*Infraestructure as a Service* o IaaS):

  Proporciona la gestión de recursos de cómputo, almacenamiento y redes a petición. Y por ello, es el modelo más similar al concepto de servidor local *on-premise* sin gestionar el mantenimiento físico de los componentes.

  De tal manera, el uso de este modelo de servicio puede acercarse a las necesidades más comunes de los distintos usuarios al realizar la migración y transformación digital. De hecho, según el estudio de mercado realizado por Md K, Beesetty Y, Pramod B, Vineet K prevé que el mercado IaaS exceda los 473 mil millones de euros para el 2030 [14].

  IaaS proporciona un alto nivel de seguridad y redundancia, sin embargo, se debe tener presente que el nivel de abstracción es menor para el cliente, por lo que gran parte de la responsabilidad reside en su gestión. El proveedor comparte en menor parte la responsabilidad, únicamente en la infraestructura física proporcionada.

- **Plataforma como servicio** (*Platform as a Service* o PaaS):

  Proporciona un entorno de desarrollo ya preparado para usar, por lo que el usuario se podrá abstraer de la infraestructura necesaria y centrarse en el desarrollo del código para su negocio.
  Un ejemplo claro son las webs (*App Services*) o los motores de bases de datos, en el que el proveedor proporciona al cliente el servicio para alojar las

aplicaciones. De esta manera, el mismo estudio predice que el mercado en 2030 alcanzará los 313 mil millones de euros.

En PaaS se incrementa la seguridad por parte del usuario final al disponer de menor responsabilidad, ya que esta se gestionará del lado del proveedor. Sin embargo, el código de la aplicación implementado seguirá siendo un punto que deberá analizar el cliente.

- **Software como servicio** (*Software as a Service* o SaaS):

  Proporciona la aplicación ya formada al usuario, por lo que será el proveedor el que sea responsable del desarrollo, alojamiento y la actualización de las aplicaciones o herramientas. El usuario tendrá una gestión muy limitada.

  Los datos del usuario serán el componente protagonista en este modelo, como puede ser el caso de un servicio de correo, donde el usuario accede a la aplicación ofrecida por el proveedor y utiliza las funcionalidades ofrecidas por ellos. Restando al mínimo la responsabilidad de seguridad del usuario.

  En este caso, dada la facilidad para el usuario, el estudio anteriormente mencionado estima que el mercado en 2030 alcanzará los 689 mil millones de euros. Siendo el servicio con mayor crecimiento.

- **Función como servicio** (*Function as a Service* o FaaS):

  Proporciona un servicio sin servidor, es decir, proporcionando al usuario el desarrollo y ejecución de pequeñas piezas de código modulares a respuestas a eventos sin ocuparse de la infraestructura que las aloje.

  En el caso de Azure, existen las *Functions*, parecidas a las *App Services* mencionadas anteriormente, pero abstrae totalmente al usuario final del recurso necesario para la ejecución de su código.

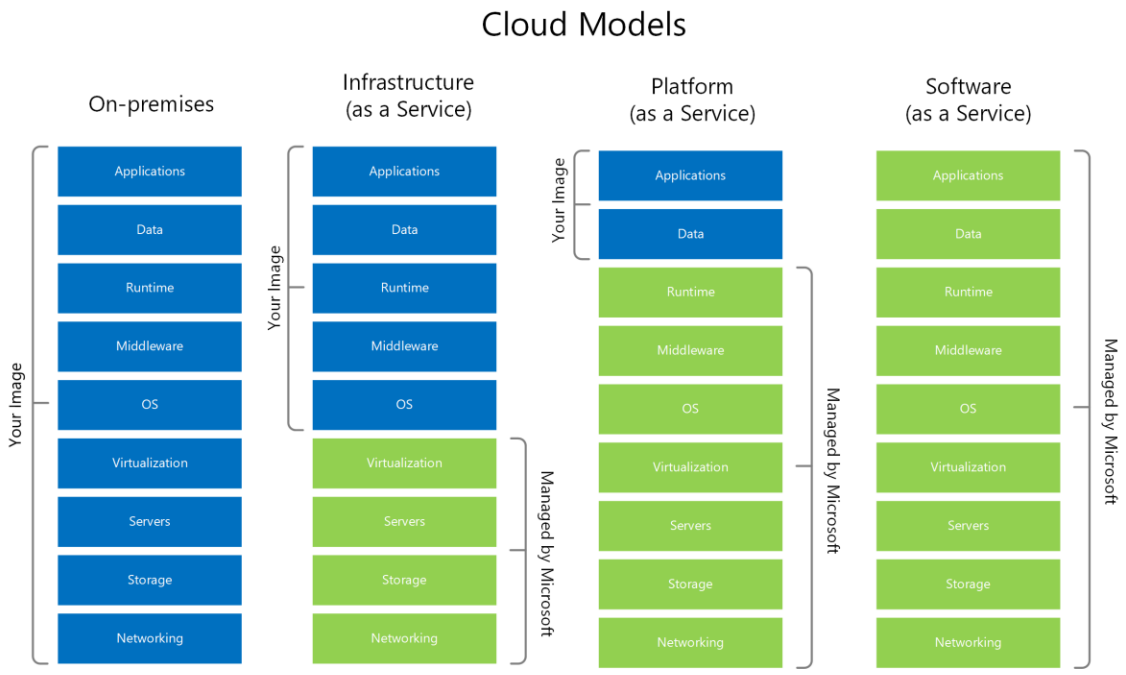


Ilustración 3: Responsabilidad en modelo de nubes [15]

Para finalizar con los conceptos básicos, cabe mencionar que las distintas implementaciones en la nube también dependen de donde estén ubicados los servidores:

- **Nube privada**: en este modelo está centrada la infraestructura totalmente a una organización.

- **Nube pública**: el servicio está gestionado por un proveedor externo, por lo que se comparten con múltiples organizaciones.

- **Nube hibrida**: se combinan ambas, una organización utiliza servicios de una nube pública en combinación con su nube privada.

- **Multinube**: se combinan dos o más nubes públicas otorgando mayor resiliencia en caso de catástrofe general de uno de los proveedores.

Este proyecto se centra únicamente en el caso de la nube pública (o parte de multinube) provisionada por Microsoft, Azure.

2.1.1. Migración a la nube

Durante el proceso de transformación digital, las organizaciones gestionan –de forma autónoma o delegada– el esfuerzo requerido para migrar su infraestructura, datos y aplicaciones hacía la nube.

Por ello, es importante tener en cuenta los riesgos en la seguridad de la nueva implementación del negocio de la organización. El paso hacía este cambio, por ello, se realiza en gran parte por los beneficios ofrecidos en la nube:

- **Escalabilidad**: la computación en la nube puede escalar fácilmente y en ocasiones en caliente, es decir, sin parar la productividad. Para ello existen distintas maneras, verticalmente (aumentando/disminuyendo recursos del servidor, por ejemplo, aumentando los Cores Virtuales) u horizontalmente (aumentando/disminuyendo instancias de servidores que reciban las peticiones).

- **Elasticidad**: con una gran escalabilidad, la nube permite que esta se realice de manera automatizada dependiendo de la carga de trabajo que este recibiendo, permitiendo una gran disponibilidad y un coste acorde.

- **Coste**: el proveedor se encarga de ofrecer una gama de precios en función al uso y al recurso necesario en el momento concreto. Dando mayor flexibilidad de costes al usuario.

- **Funcionamiento**: la funcionalidad mejora debido a las distintas características que ofrece la nube, menor latencia debido a las distintas ubicaciones de los servidores, resiliencia y rápida recuperación a fallos debido de la misma manera a disponer de múltiples servidores alojados en distintas ubicaciones físicas, permitiendo por ejemplo recuperación ante desastre, etc.

- **Flexibilidad y alta disponibilidad**: disponer de los servicios en cualquier momento y en cualquier dispositivo facilita la capacidad de disponibilidad para el usuario, aumentado tiempos de funcionamiento en SLA (*Service Level Agreement*), o hasta incluso permitiendo situaciones de teletrabajo tan comentadas tras la pandemia del Covid19.

Sin embargo, igual que se originan beneficios al cambio existen dificultades que hay que tener en cuenta:

- **Migrar grandes bases de datos**: A pesar de las distintas herramientas que ofrecen los principales proveedores *cloud*, puede ser una difícil tarea dependiendo de la cantidad de datos y las características de los mismos. Además de la gestión y cumplimiento normativo basados en la distintas leyes y normas de protección de datos dependiendo del país donde se alojarán.

- **Integridad de los datos**: los datos migrados deben ser compatibles con los servicios o *frameworks* disponibles en el proveedor *cloud*, a fin de cuentas, de no añadir mayor complejidad a la aplicación o infraestructura anteriormente alojada *on-premise*.

2.2. Entidades de control

El INCIBE (Instituto Nacional de Ciberseguridad de España), analiza de manera básica cuales son las principales amenazas y riesgos que deben tener en cuenta los usuarios u organizaciones que trabajan con cualquier proveedor *cloud*, revisando la posible mitigación y la responsabilidad del usuario.

Además, recalcan la necesidad de revisar los aspectos legales y contractuales con el proveedor seleccionado, comprobando la ubicación de los centros de datos y su gestión cumpliendo la legislación –en este caso, española– con la Ley orgánica de protección de datos (LOPD).

La relación que debe existir entre el usuario y el proveedor *cloud* debe estar regulada por un contrato y un Acuerdo de Nivel de Servicio (SLA por sus siglas en inglés). Este punto es muy importante a la hora de definir que riesgos el usuario está aceptando al contratar el servicio.

En el caso de Azure –a pesar de que puede cambiar dependiendo del tipo de contrato– por defecto, garantiza un alto nivel de porcentaje de SLA, en cada servicio ofrecido existe un nivel distinto, no siendo inferior a un 99,9% (unas 8h al año) [18].

Por otro lado, existen entidades gubernamentales que trabajan para facilitar la gestión del riesgo en el *cloud* con tratados que garantizan la unidad en el uso que se le hace a todo lo que engloba el *cloud computing*. En especial la Comisión Europea mediante la creación de la "*European Alliance on Industrial Data, Edge and Cloud*" [19]. Entre las tareas que se definen en el tratado, se mencionan las siguientes centradas en la seguridad:

- Un libro de reglas de la nube de la UE para los servicios en la nube, que proporcionará un marco europeo único de normas, transparencia sobre su cumplimiento y mejores prácticas para el uso de la nube en Europa
- El Reglamento sobre la libre circulación de datos no personales, que, junto con el Reglamento General de Protección de Datos, aumenta la seguridad jurídica de los usuarios de la nube al garantizar la libre circulación de todos los datos en la UE.
- Portabilidad de los datos: la libre circulación de los datos no personales El Reglamento también genera confianza al facilitar un trabajo de autorregulación sobre el cambio de nube y la seguridad en la nube.
- Ciberseguridad: a petición de la Comisión, la agencia europea de ciberseguridad ENISA está trabajando en un esquema único de certificación europea de ciberseguridad para los servicios en la nube. El sistema proporcionará mayores garantías a las empresas, las administraciones públicas y los ciudadanos de que sus datos están seguros dondequiera que se almacenen o procesen.
- Protección de datos en la nube: la Comisión facilita una plataforma para que la industria desarrolle códigos de conducta para la protección de datos en la nube. Esto ha dado lugar a dos Códigos de Conducta que están siendo revisados por el Consejo Europeo de Protección de Datos.

- Acuerdos de nivel de servicio en la nube (SLA) normalizados que garantizan la calidad de los servicios en la nube en el mercado europeo.

Otra entidad referente multinacional es la Cloud Security Alliance (CSA), encargada de definir estándares, certificaciones y buenas prácticas en todo lo que engloba al entorno *cloud*, y se encarga anualmente de realizar un estudio de las TOP amenazas.

El último reporte, publicado en la conferencia "Seattle and RSA" en San Francisco, el 7 de junio de 2022, *Top Threats to Cloud Computing: Pandemic 11* [20], concluye en que cada vez los problemas tradicionales de seguridad en la nube son cada vez menos preocupantes para el usuario final.

2.3. Amenazas y Riesgos

Para analizar las diferentes amenazas y posibles riesgos que existen en todo lo que engloba al *cloud*, se deben conocer los diferentes modelos descritos en el apartado anterior (IaaS, PaaS, SaaS o FaaS) para delimitar, por un lado, las consideraciones que el usuario debe de tratar, es decir, la responsabilidad y, por otro lado, las facilidades que el proveedor dispone ante el servicio.

Según el recién mencionado reporte anual ofrecido por la CSA sobre los *Top Threats to Cloud Computing: Pandemic 11*, esta es la puntación obtenida en base a una encuesta realizada a un grupo experto en seguridad informática, en el cual se les ofrecía una lista de 26 amenazas, puntuables del 1-10 y en la que finalmente concluyeron el siguiente top 11:

Survey Results Rank	Survey Average Score	Issue Name
1	7.729927	Insufficient ID, Credential, Access and Key Mgt, Privileged Accounts
2	7.592701	Insecure Interfaces and APIs
3	7.424818	Misconfiguration and Inadequate Change Control
4	7.408759	Lack of Cloud Security Architecture and Strategy
5	7.275912	Insecure Software Development
6	7.214493	Unsecure Third Party Resources
7	7.143066	System Vulnerabilities
8	7.114659	Accidental Cloud Data Disclosure/ Disclosure
9	7.097810	Misconfiguration & Exploitation of Serverless & Container Workloads
10	7.088534	Organized Crime/ Hackers/ APT
11	7.085631	Cloud Storage Data Exfiltration

Ilustración 4: CSA Top Threats 2022 [20]

1) Identidad insuficiente, Credenciales, Gestión de Acceso y Claves, Cuentas Privilegiadas

El acceso a los recursos se debe supervisar con una correcta política de control de accesos, ajustándose al rol requerido y necesario para la identidad concreta (o grupo). En oposición, la identidad podrá realizar tareas que no debería realizar bajo su rol y por ende impactará en un mayor nivel de amenaza en caso de uso indebido.

Para ello, se necesita un buen mantenimiento y vigilancia del centro de gestión de identidades, en Azure, pudiéndose gestionar en diferentes niveles jerárquicos, por *Tenant* o inquilino (mayor nivel) > subscripción > grupo de recursos > recurso concreto en el panel IAM.

Las cuentas con privilegios elevados deben analizarse llevando un control reportado, para situaciones de cambios de rol del personal (o incluso salida) se evite la exfiltración de datos o la probabilidad de compromiso.

Esta es una amenaza que depende únicamente del usuario, la responsabilidad no se comparte con el proveedor dado que es totalmente administrable por el mismo. Y puede afectar a todos los modelos de datos recién mencionados (SaaS, PaaS e IaaS).

El impacto negativo sobre esta amenaza puede llegar a ser grave, dependiendo del rol de cuenta o credencial que se ha expuesto, pudiendo existir accesos indebidos a los servicios con lo que todo eso puede conllevar, modificación de otras identidades, dejando inaccesible para el personal autorizado o incluso el cifrado de datos mediante un *ramsomware*.

2) Interfaces y APIs inseguras

Partiendo de la base de la definición de API (Application Programming Interfaces) como un conjunto de protocolos y definiciones utilizados para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones.

Debido a la gran utilidad que brindan las APIs, según un informe de Akamai de 2021, han aumentado su uso un 53% tras el año anterior, por lo que, al ser una herramienta tan utilizada entre diferentes componentes, también son un gran punto de entrada y por ello un vector de ataque a analizar.

De modo que, deben revisarse vulnerabilidades debidas a malas configuraciones, malas prácticas de codificación, falta de autenticación y autorización inapropiada o débil.

Esta situación puede tener distintos puntos de vista en función de la responsabilidad, por un lado, el usuario debe gestionar esta interfaz o API llevando una buena política de desarrollo de código sin vulnerabilidades –al

menos publicadas– y diseñar un modelo de autenticación y autorización acorde a las necesidades, junto con un cifrado seguro con certificados SSL correctamente configurados provenientes de una entidad certificadora de confianza.

Y, por otro lado, el proveedor puede tener responsabilidad en caso de no controlar las últimas versiones de la plataforma y el *framework* o *runtime* donde se implante la API, por ejemplo, como un *frontend* con una versión de .NET con vulnerabilidades.

3) Mala configuración y control de cambios inadecuado

En cualquier entorno, ya sea *on-premise* o en este caso *cloud* las malas configuraciones han sido un gran agujero de seguridad, a pesar de las capas de protección que se implementen, estas serán saltadas (*bypasseadas)* por una mala configuración.

La diferencia puede venir dada en como esta mala configuración o detección de cambios es detectada en el *cloud* a diferencia de los sistemas tradicionales, dado que, en muchas ocasiones, son necesarios múltiples comprobaciones en distintos recursos para encontrar la causa del problema que en sistemas *on-premises* eran de más fácil acceso.

Además, en sistemas tradicionales, dado que los servicios e infraestructura tardaban más días en llevarse a producción era más sencillo encontrar estos agujeros en entornos de desarrollo o preproducción, en general, en la nube, la automatización viene dentro de sus características, por lo que el llevar a producción un error en la configuración será más rápida.

Esta mala configuración trae consigo amenazas inherentes con permisos excesivos, credenciales y ajustes por defecto, controles de seguridad estándares y monitorización desactivados y gestión de secretos insegura.

La responsabilidad puede estar compartida entre usuario y proveedor, debido a mala praxis o errores en la propia configuración gestionada por el propio proveedor y afecta a todos los modelos.

4) Problemas de seguridad en la arquitectura

Esta amenaza viene dada por la falta de conocimiento debida a la gran cantidad de servicios y recursos ofrecidos por el proveedor, además de estar en constante cambio.

Un buen estudio y diseño del entorno a crear es clave para evitar posibles fallos mencionados anteriormente, como accesos no autorizados o roles innecesarios, además de una buena segmentación de redes entre los servicios y otras cuestiones similares de arquitectura de sistemas.

Claramente afecta a todos los modelos (SaaS, PaaS e IaaS) y tiene responsabilidad el diseñador de la arquitectura ante los puntos anteriormente mencionados.

5) Desarrollo de software inseguro

Partiendo de la base que el desarrollo de software inseguro es una amenaza ya en su inicio, transportado a un entorno *cloud*, donde las tecnologías tienden a aumentar la complejidad, por lo que propician funciones involuntarias que permiten la creación de *exploits* y desconfiguraciones.

Para ello, el desarrollador que suba el código a cualquier servicio, como pueden ser en páginas web (*App Services*), deberían utilizar herramientas internas para evitar subir esos errores de código.

Afecta tanto a PaaS, como a IaaS y SaaS también, y la responsabilidad a pesar de que parezca únicamente del usuario o desarrollador por subir código inseguro, puede ser también del propio proveedor por su implicación en servicios que también disponen de código utilizados por el usuario, por ejemplo AKS (*Azure Kubernetes Services*) en el que exista un conocido *exploit* en la versión que proporcione el proveedor y que luego el usuario implementará en sus aplicaciones contenerizadas, con el agujero de seguridad existente.

6) Recursos de terceros inseguros

Es una amenaza difícil de detectar debido a la poca transparencia que en general pueden brindan los recursos que se añadan de terceros, estos pueden ser integrados por el propio usuario o incluso por el propio proveedor *cloud*. Un claro ejemplo en el caso de Azure es la integración del recurso *Databricks* para tratamiento de datos ETL dentro del entorno, siendo precisamente un recurso de terceros.

En muchas ocasiones la integración con estos recursos de terceros puede estar tan afianzadas al flujo habitual de trabajo del usuario que si existe alguna vulnerabilidad conocida (y/o anunciada) del recurso en cuestión la substitución se hace muy complicada sin alterar la actividad habitual del negocio. Dando en ocasiones una puerta de entrada, mediante un pivoteo, a los demás recursos propios del cliente que aparentemente no tienen vulnerabilidades.

Por lo tanto, la responsabilidad puede estar compartida, el recurso de tercero es inseguro pero la adopción de este puede estar gestionada por el propio proveedor y no tan solo por el usuario. Además, afecta a todo tipo de modelo de servicio.

7) Vulnerabilidades del sistema

Las vulnerabilidades que se dan en el sistema son fallos en la propia plataforma de los servicios alojados en el *cloud*, pudiendo ser explotadas en un

intento de comprometer la confidencialidad, la integridad y la disponibilidad de los datos o incluso interrumpiendo la disponibilidad del servicio.

Existen 4 categorías de vulnerabilidades del sistema:

- **Vulnerabilidades 0day (de día cero):** estas vulnerabilidades se les llama de día 0 porque el fabricante o desarrollador de la aplicación o componente no es conocedor de las mismas por lo que se explotan por un atacante de forma libremente. Sucedió recientemente con Log4Shell [21], en la que se afectó una librería muy utilizada en Java.

- **Falta de parches de seguridad:** la actualización de versiones con parches de seguridad, o también conocidos como KBs (Microsoft Knowledge Base) en sistemas Windows parchean vulnerabilidades conocidas y, por ende, solucionables en la versión desplegada.

- **Vulnerabilidades basadas en la configuración:** estas vulnerabilidades vienen dadas por el despliegue de sistemas con ajustes por defecto o mal configurados –que se analizarán más detalladamente en apartados posteriores–, como son protocolos de seguridad heredados, cifrados y permisos débiles o interfaces mal protegidas.

- **Credenciales débiles por defecto:** una política de credenciales por defecto o débiles permite a los atacantes un fácil acceso a los recursos con ataques de fuerza bruta o diccionarios. Por otro lado, contraseñas almacenadas de forma insegura y no tratadas como secretos, por ejemplo, en *Key Vaults* –servicio *cloud* para el almacenamiento de secretos de forma segura– pueden ser filtradas con mayor facilidad.

Pueden afectar a todos los modelos de servicio de *cloud* y además puede ser una responsabilidad compartida debida a la naturaleza de la vulnerabilidad, ¿ha habido una correcta actualización de versiones por parte del usuario? –siendo claramente responsabilidad de este– o ¿existe una familia de máquinas virtuales que utiliza un controlador en la tarjeta gráfica con algún 0day? – por lo que responsabilidad del proveedor– o ambas situaciones.

8) Revelación accidental de datos

En ocasiones, el cambio de la propiedad física de los datos hacía el *cloud* puede producir una falta de gobernanza de la seguridad y del control de los mismos.

También, al disponer de un mayor número de configuraciones de los recursos en los distintos proveedores, pudiendo diferir bastante entre ellos, propicia un sistema más común de configuraciones erróneas que conduzca a fugas de datos involuntarias, los famosos *data leaks* que se están publicando

recientemente por mala configuración en el servicio de Azure Blob Storage [22] y que se analizarán posteriormente.

Similar a la amenaza de errores en las configuraciones puede llegar a ser de responsabilidad compartida entre usuario-proveedor, de la misma manera que puede pasar en cualquier modelo de servicio (SaaS, Paas o IaaS).

9) Desconfiguración y explotación de las cargas de trabajo sin servidor y en contenedores

El uso de tecnologías y computación en el *cloud* permite un mayor avance de metodologías agiles y comportamientos más DevOps –cambios en el comportamiento entre equipos habitualmente aislados, desarrolladores, operaciones, *testers* y seguridad hacía una coordinación y colaboración para evitar tiempos innecesarios de más y errores en las implementaciones– hacen que roles que anteriormente eran en silos, se compartan y puedan acarrear errores por inexperiencia.

Cuando el usuario despliega infraestructura como código (IaC), permite dedicar su tiempo a otras tareas menos repetitivas, sin embargo, una mala configuración de este despliegue puede ser más grave y difícil de detectar que un despliegue manual. Cabe decir que, a la inversa, con un nivel de control de código esto precisamente evita errores comunes y el factor humano.

Por otro lado, el proveedor puede ofrecer FaaS (un servicio sin servidor) por el cual la responsabilidad es asumida por el mismo y permite al usuario desplegar en contenedores de corta duración, es decir, que se crean, realizan su tarea y se destruyen, evitando así que una posible vulnerabilidad y/o *exploit* pueda afectar gravemente al servicio. Sin embargo, siempre existe la mala configuración por desconocimiento o inexperiencia de bloquear que este servicio contenerizado se elimine al terminar, pudiendo permitir que el entorno si se vuelva inseguro y por ello bajo la responsabilidad del usuario.

10) Crimen organizado, hackers y APT

Las amenazas persistentes avanzadas (o APT) es la denominación que se utiliza para describir una campaña de ataque en la que un intruso/s (o crimen organizado) establece una presencia ilícita a largo plazo en una red para extraer datos altamente sensibles, normalmente después de una intrusión instalando una puerta trasera (*backdoor*).

Las APTs establecen sofisticadas tácticas, técnicas y protocolos (TTPs) para infiltrarse en los objetivos y permanecer indetectados, pivotando en muchos casos entre otros sistemas de la misma red.

Dada los múltiples tipos de APTs y hackers delincuentes que estén detrás el resultado es muy diverso, en general se busca un objetivo reputacional de la

compañía, económico o incluso político. Y afecta a todos los modelos de servicios siendo responsabilidad de cualquier tipo o incluso compartida.

11) Exfiltración de datos

La exfiltración de datos es un incidente que implica la perdida de información sensible, protegida y confidencial, pudiendo ser liberados, robados, modificados o eliminados de forma ilícita por un intruso.

Suele ser un objetivo principal de un ataque dirigido como consecuencia de una previa explotación de una vulnerabilidad o mala configuración previamente tratada en anteriores apartados.

Dentro de la exfiltración de datos existen distintos niveles de clasificación de los datos, en función del carácter identificativo y de la definición que describe la ley orgánica de protección de datos, LOPD.

Estos deben de tratarse de una manera u otra en función de su clasificación, siendo públicos, personales no especiales (identificativos como: DNI, dirección, imagen, etc.), personales especiales (relativos a la salud, vida u orientación sexual, religión, etc.) y confidenciales [23].

Debida a la importancia de los datos, posiblemente un activo muy importante del usuario u organización y la facilitad que proporciona la computación en el *cloud* para su almacenamiento (configuración, elasticidad, resiliencia, etc.) es un factor muy importante que analizar, y por ello configurar adecuadamente.

Una exfiltración puede suceder tanto en una máquina virtual de IaaS, como un servicio PaaS (*Storage Account* mal configurado), un servicio SaaS (una cuenta de correo vulnerada) con responsabilidades en ambas partes.

3. Seguridad en Azure

Como se ha podido ver en el punto anterior, existen varias amenazas explotables en un entorno *cloud* como Azure –siendo extrapolable a todos los demás proveedores– que ya eran reproducibles en entornos *on-premise.*

Se parte de la base de la que la infraestructura alojada en el *cloud* no deja de ser en poco o gran parte inherente a la infraestructura tradicional y, por lo tanto, existen los mismos problemas de malas configuraciones, fuga de datos, *APTs* y malos diseños de infraestructuras y software trasladados a este tipo de arquitecturas.

En gran parte, la diferencia frente los entornos del *cloud computing* el control y la responsabilidad entre el proveedor y el cliente y la menor madurez de los equipos. Y, por lo tanto, el primer punto deberá ser conscientes de las mismas y conocer las buenas prácticas que proporciona Azure.

3.1. Responsabilidad compartida

En comparación con el *cloud* privado o infraestructura local *on-premise,* donde la responsabilidad está aislada al propio propietario de la misma, en cambio, en el caso del *cloud* público no siempre es así.

En ocasiones, esta responsabilidad sigue siendo del propietario, es decir del proveedor Microsoft, a bajo nivel del hardware ofrecido, en malas configuraciones, productos vulnerables u otras situaciones mencionadas en el apartado 2.

Por otro lado, está la vertiente en la que la responsabilidad recae en el usuario, alojando aplicaciones vulnerables, no cifrando datos o conexiones o mal configurando los servicios proporcionados. En cualquier caso, dependerá del modelo de servicio usado.

Y finalmente, existen casos de responsabilidad compartida, en la que los puntos son tratados tanto por el proveedor como el cliente, dependiendo de la misma forma del tipo de modelo objetivo.

<table>
<tr><th>Responsabilidad</th><th>On-Premise</th><th colspan="2">IaaS</th><th colspan="2">PaaS</th><th colspan="2">SaaS</th></tr>
<tr><td>Cuentas</td><td>Cliente</td><td colspan="2">Cliente</td><td colspan="2">Cliente</td><td colspan="2">Cliente</td></tr>
<tr><td>Información y datos</td><td>Cliente</td><td colspan="2">Cliente</td><td colspan="2">Cliente</td><td colspan="2">Cliente</td></tr>
<tr><td>Clientes y dispositivos</td><td>Cliente</td><td colspan="2">Cliente</td><td colspan="2">Cliente</td><td>Cliente</td><td>Azure</td></tr>
<tr><td>Identidad y gestión de acceso</td><td>Cliente</td><td colspan="2">Cliente</td><td>Cliente</td><td>Azure</td><td>Cliente</td><td>Azure</td></tr>
<tr><td>Control de aplicación</td><td>Cliente</td><td colspan="2">Cliente</td><td>Cliente</td><td>Azure</td><td colspan="2">Azure</td></tr>
<tr><td>Redes virtuales</td><td>Cliente</td><td>Cliente</td><td>Azure</td><td colspan="2">Azure</td><td colspan="2">Azure</td></tr>
<tr><td>Infraestructura alojada</td><td>Cliente</td><td>Cliente</td><td>Azure</td><td colspan="2">Azure</td><td colspan="2">Azure</td></tr>
</table>

Hipervisor	Cliente	Azure	Azure	Azure
Servidores y almacenamiento	Cliente	Azure	Azure	Azure
Redes físicas	Cliente	Azure	Azure	Azure

Tabla 1: Responsabilidad proveedor-cliente

3.2. Formación del equipo

El siguiente paso, siempre que se adopta una nueva tecnología en el uso diario del negocio es la formación al usuario o usuarios de como adaptarse a usar esta nueva tecnología.

En ocasiones pueden aparecer nuevos roles y cambios culturales en las que protagonizan cambios de responsabilidades en las tareas a realizar a partir de ese momento. Esto requiere una nueva mentalidad y enfoque.

Microsoft, en su documentación, proporciona una guía de nuevas funciones y tareas con una responsabilidad clave para una organización que adopte un entorno *cloud*. Basándose en una visión DevOps, adoptando la seguridad internamente al concepto (*DevSecOps*), en un objetivo global del equipo formado.

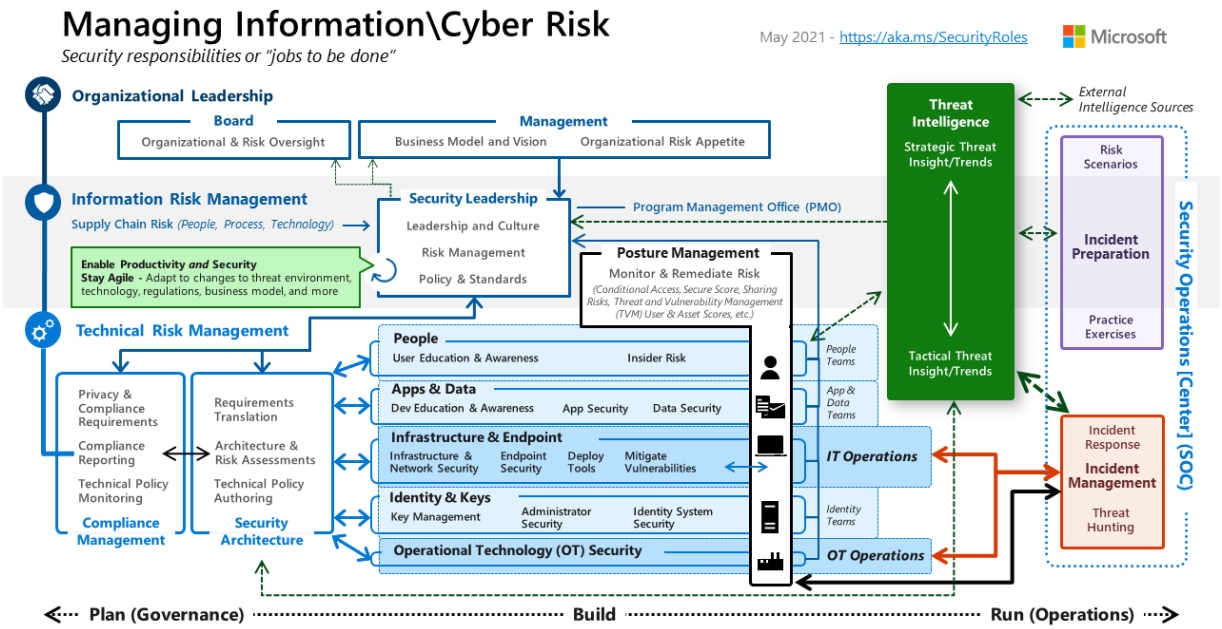


Ilustración 5: Funciones de seguridad Azure [24]

Que, analizando esta ilustración sobre las funciones de Seguridad en el *cloud* de Microsoft [24] se diferencian de forma sintetizada:

- **Función de los estándares y la directiva de seguridad**
  Este rol tiene como objetivo la de crear, aprobar y publicar los estándares y la directiva de seguridad para guiar las decisiones de seguridad dentro de la organización.

Se entiende como estándares a el uso seguro de las plataformas en el *cloud*, de los modelos DevOps e inclusión de aplicaciones y APIs, control de red (estrategia de segmentación), etiquetado de recursos y definición de los procesos de evaluación de seguridad.

En el caso de la directiva se debe reflejar los objetivos sostenibles a largo plazo con la estrategia de seguridad en la organización, como es el cumplimiento normativo, referencia cultural y los procedimientos recomendados del sector.

Las directivas deben permanecerse estáticas pero los estándares deben ser dinámicos y revisarse continuamente para mantenerse al día de los cambios en la tecnología del *cloud*. Siendo un rol más dirigido a perfiles de tecnología de la información, auditores legales y PRL (prevención de riesgos laborales).

- **Función de administración del cumplimiento**
  Complementando la función anterior, el objetivo de esta función es la gobernanza de datos, velando por que la organización cumpla los requisitos normativos y las directivas internas, realice un seguimiento del estado y lo notifique a los interesados.

  Para facilitar la creación de informes de seguimiento de estos registros y directivas, Microsoft facilita la siguiente herramienta: *Microsoft Purview Compliance Manager*.

- **Función de operaciones**
  Su objetivo principal es detectar de forma proactiva, responder y solventar los ataques que afecten a los recursos con herramientas para ese fin.

  La evolución de estas funciones recae principalmente en el SOC (Centro de operaciones de Seguridad) monitorizando y registrando los eventos que vayan sucediendo y analizándolo herramientas como un sistema SIEM (Sistema de información y eventos de seguridad), descrito por la empresa Gartner, como la combinación entre un SEM y SIM.

  De esta manera, estas herramientas proporcionarán al equipo una visión de gran calidad de la situación de su entorno que complementadas con tecnologías de IA y *machine learning* pueda ayudar ante la respuesta contra incidentes comparando comportamiento con una *bigdata* de casos parecidos y acciones mitigadoras apropiadas.

- **Función de la inteligencia sobre amenazas**
  La inteligencia sobre amenazas proporciona una respuesta basada en un catálogo de casos conocidos almacenados en un *bigdata* que permite aplicar una acción sobre un ataque activo y amenazas potenciales.

- **Función de la arquitectura de seguridad**
  La idea principal del rol es la de implementar los objetivos de negocio en una documentación y diagramas a seguir para guiar las decisiones técnicas de seguridad en la infraestructura en el *cloud*.

  Por ello, el rol de arquitecto de seguridad tendrá que involucrarse junto con los equipos de desarrollo y operaciones en guiar las decisiones de seguridad durante todo el ciclo de vida del producto, desde su nacimiento y de forma iterativa (una vez más, metodologías ágiles + DevOps).

  Un buen plan de arquitectura previene retrabajos en el futuro en caso de amenazas no conocidas o malas configuraciones, esto por lo tanto aplica a todos los modelos de servicios. Además, deben especificar los permisos y las identidades que accederán a los recursos, gestionando el Azure Active Directory + Permisos IAM + roles RBAC que se analizarán en puntos futuros.

- **Función de seguridad de las personas**
  En la seguridad informática, el eslabón más débil es el usuario, comprobado en un estudio realizado por I. Arce en IEEE Security & Privacy [25]. Por muchas capas de seguridad que se añadan, siempre existen técnicas como la ingeniería social o el propio error humano que dejan un agujero de seguridad accesible al atacante.

  Por lo que esta función es ampliamente importante para promover un entorno seguro. Y para ello, involucrar a todos los usuarios o equipo de la organización con técnicas como las formaciones, comunicaciones, talleres, *hackatons* y el uso de ludificación y gamificación –aprendizaje con pequeñas recompensas–.

  Además de toda la intención formativa de esta función, existe la monitorización con registros de toda la actividad realizada por los usuarios, pudiendo detectar situaciones anómalas como, por ejemplo, la eliminación de N recurso por la identidad X durante un periodo fuera de su jornada laboral.

- **Función de seguridad de las aplicaciones**
  En este punto, la función se centra en la seguridad de las aplicaciones alojadas y desplegadas, por lo que entra protagonismo *DevSecOps* mencionado anteriormente.

  Este modelo de desarrollo implica no tan solo la creación de aplicaciones seguras si no todo el ecosistema de integración e implementación continua (CI/CD) que incrementa el *feedback* durante el flujo de despliegue de nuevas funcionalidades y arreglos que facilita una gestión de la seguridad por las siguientes características:

    - Testeo anticipado, al existir un despliegue continuo, se van realizando pruebas continuas donde el usuario podrá encontrar los errores de seguridad en fases tempranas, incluso antes de su salida a producción.

- La propia integración continua (CI/CD) que permite desplegar pequeños cambios –en muchas ocasiones en caliente sin parar el entorno– de manera que existe un mayor control de seguridad de la aplicación o servicio desplegado.
- Testeos de seguridad de código alojado de forma estática y/o dinámica, es decir, sin ejecutar código o ejecutándolo, evitando por ello la deuda técnica (el mencionado costo del retrabajo)

- **Función de la seguridad de datos**

  Esta función tendrá como objetivo principal la de proteger mediante controles y configuraciones de seguridad los datos de cualquier tipo de la organización.

  En este punto entra en juego cumplimentar los estándares de seguridad del dato según la legislación estatal y europea, mencionados anteriormente como el reglamento general de protección de datos y la LOPD.

- **Función de la infraestructura y puntos de conexión**

  El equipo con este rol proporcionará la protección de seguridad, la capacidad de detección y los controles de respuesta en los componentes de infraestructura y red que usarán los usuarios y las aplicaciones. Los objetivos son los siguientes:

  - Detección de errores de configuración e inventario
  - Administración de vulnerabilidades
  - Tecnologías en contenedores como *Azure Kubernetes Services*.
  - Instalación y mantenimiento de agentes de seguridad
  - Creación de listas permitidas y excluidas (*whitelists* y *blacklists*) de aplicaciones y protocolos.
  - Uso de infraestructura como código (IaC) con plantillas, como ARM y Biceps de Microsoft o de terceros, como *Terraform*–se analizarán en apartados posteriores–.
  - Accesos temporales y suficientes a los recursos basados en los principios de privilegios mínimos.
  - Administración de puntos de conexión unificada para gestionar todos los dispositivos que acceden a los entornos *cloud*.
  - Un diseño y arquitectura bien definido.

  Esta función es muy importante que se defina y se escoja al equipo debidamente, partiendo de un perfil más técnico, como un arquitecto u operaciones, el mismo equipo SOC mencionado anteriormente y algún integrante con experiencia en cumplimiento y auditoria.

- **Función de administración de identidades y claves**

  El objetivo es controlar de manera correcta, por un lado, el acceso con autenticación y autorización de usuarios (personas), servicios, dispositivos y otras aplicaciones y, por otro lado, administrar las claves, secretos y certificados para las operaciones criptográficas.

Los controles de identidad es uno de los perímetros de seguridad principal de las aplicaciones en el *cloud*, de hecho, reemplaza en gran medida a la autenticación basada en claves dada su versatilidad. Puede autenticarse un servicio sin interacción humana, sin almacenar las claves en ningún sitio (evitando fugas de datos sensibles) y automatizando procesos.

El primer paso para empezar a controlar las identidades de los servicios es habilitando las identidades administradas de Azure Active Directory, esto es dar la opción a autenticar a un recurso con un token de identidad sin necesidad de administrar credenciales. En la ilustración 6 se observa un ejemplo del uso de identidad para que una web acceda a una cuenta de almacenamiento de datos sin personificar con usuarios.

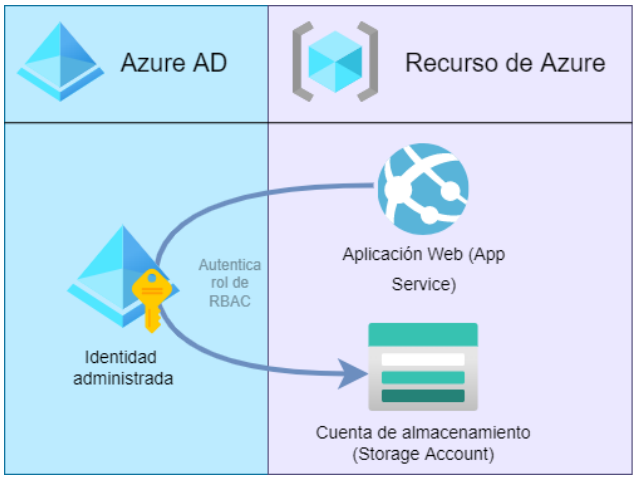


Ilustración 6: Identidad administrada

- **Función de preparación de incidentes**

  La función principal en este caso es la de tener la habilidad de preparar informes de incidentes de seguridad en caso de que exista un ataque o amenaza aparente.

  No solo aplica a la realización de casos de incidentes ya sucedidos, si no también planes de prevención ante incidentes analizando métricas y datos que puedan indicar un posible agujero en el entorno. Por ejemplo, uno de los pasos iniciales a la hora de migrar hacía el *cloud* –decidir si una decisión correcta para el negocio– es la propia preparación de un DAFO, un esquema de debilidades, amenazas, fortalezas y oportunidades como se muestra en la tabla 2 ejemplificando un reporte de incidente de seguridad:

	FORTALEZAS	DEBILIDADES
Origen INTERNO	Baja inversión inicial	No transparencia en modelos de servicios
	Despliegue casi inmediato	
	Escalabilidad, resiliencia y disponibilidad	Costes de infraestructura diferentes a los tradicionales
	OPORTUNIDADES	AMENAZAS
Origen EXTERNO	Incremento en la estabilidad de las aplicaciones	Distintos competidores con servicios semejantes.
	Competencia con otros negocios similares	Reducción de la gobernanza de los datos.
	Capacidades de seguridad externos	

Tabla 2: DAFO entorno cloud

Por otro lado, una vez ocurrido un incidente de seguridad, el equipo que lleve esta función deberá realizar comunicarlo dentro de la organización y a las partes interesadas, realizar un análisis forense post mortem y redactar planes de autoprotección.

Todas estas funciones se pueden compartir entre equipos dependiendo de las capacidades y numero de los integrantes dependiendo del tamaño de la organización.

3.3. Servicio para las buenas prácticas de diseño

La mejor forma de crear un entorno seguro por defecto es utilizando de base un estándar de seguridad mínimo, de manera que, todos los recursos que se creen en la cuenta del usuario deban cumplir unos requisitos establecidos. Y, por lo tanto, parten de una política global de configuración.

Para ello existe una herramienta, *Azure Blueprints*, en la que el usuario podrá crear plantillas y reglas de manera declarativa para ser usadas repetidas veces asegurando la implementación idónea.

Comprobando las funciones propias de este servicio, se puede deducir que realmente es una mejora a las ya existentes plantillas (templates) y a las existentes políticas de Azure. Lo que difiere de estos dos servicios ya presentes es el modo en que se integra, *Blueprints* existe una sincronización una vez ya creado el recurso entre la plantilla declarada y el propio recurso creado. De la misma manera, ocurre con la creación de directivas y políticas creadas a mano, permitiendo la gobernanza de forma estandarizada.

Por otro lado, hay que tener en cuenta que *Blueprints* necesita una serie de permisos sobre los recursos para crear/eliminar/asignar (*write/delete*) las plantillas o planos que deberán aplicarse.

Dentro de las opciones que permite este servicio está la de crear plantillas de ejemplo ofrecidas por Azure que son de interés común, algún ejemplo son los siguientes:

- Plantilla de directivas comunes aplicables a una subscripción
- Configuración de roles grupos de recursos
- Directivas para abordar controles, como la ISO 27001
- Configuración de redes y NSG

Los pasos a seguir para configurar una serie de directivas se mencionan en los anexos [págs. 90-92]. De la forma que se ahí se puede comprobar, esta herramienta puede aplicarse para validar una serie de buenas prácticas, no tan solo de seguridad, si no directamente directivas y necesidades propias de negocio.

3.4. Herramientas y Servicios de seguridad

En este apartado, la idea es analizar los distintos servicios y recursos que Azure proporciona para que el usuario pueda aplicar un nivel estándar de seguridad en su entorno.

Para ello, se seguirá la documentación que proporciona Azure [26], en la que se indica que herramientas se proporcionan en función de dos distinciones ilustradas en la imagen 7, por un lado, por filas, objetivo de seguridad (prevención, detección y respuesta) y, por otro lado, por columnas, tipo de recursos a proteger (identidad/acceso, infraestructura/red, datos/aplicaciones y acceso clientes). Las cuales se diferenciarán seguido a la ilustración.

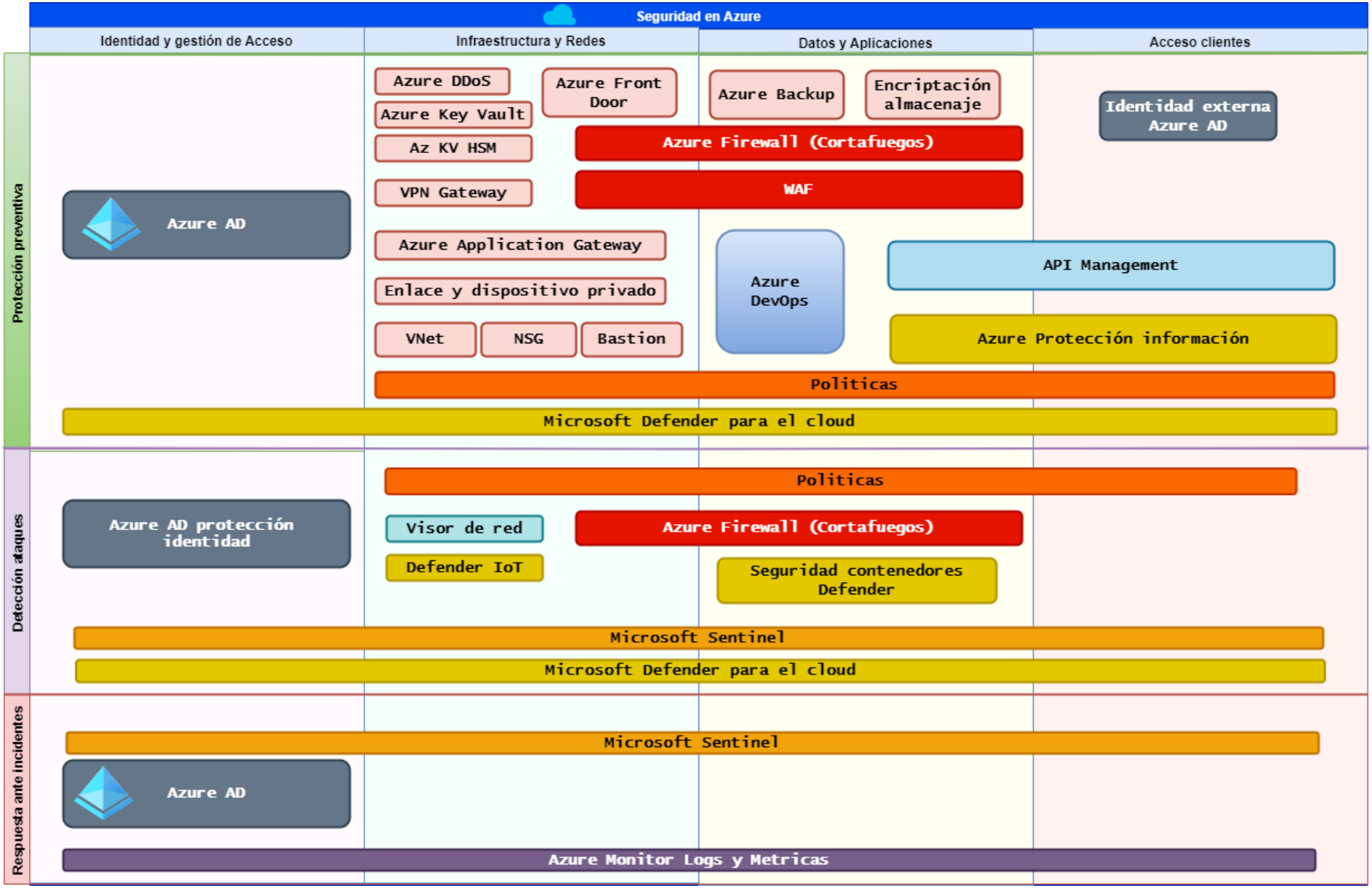


Ilustración 7: Herramientas de seguridad

3.3.1. Protección y prevención

Los servicios de este apartado tienen el objetivo proteger mediante capas los recursos de manera que se prevenga el máximo de ataques posibles que pueda sufrir el entorno. Desde una correcta gestión de usuarios, roles y accesos, gestión de los dominios del usuario, a la segmentación de redes que abarcarán recursos como máquinas virtuales, gestión de secretos, copias de seguridad de datos y aplicaciones hasta las conexiones con servicios de terceros. Se analizarán en siguientes apartados.

- **Azure Active Directory**
  Es el servicio de administración de acceso e identidades en el *cloud* (IDaaS, solución de identidad como servicio), por lo que es una importante capa dentro del perímetro de seguridad de toda la red a tener en cuenta.

  De la misma manera que en el computo *on-premise* existe *Active Directory Domain Services* (AD DS), normalmente alojado en el controlador de dominio, uno de los objetivos a atacar más suculentos dado el poder en las cuentas gestionadas vulneradas y su posterior explotación total de los sistemas.

  La ventaja que proporciona Azure frente al AD DS, por un lado, no solo permite el *single sign-on* (SSO) –uso de una cuenta federada para acceder a varios servicios–, sino también accesos terceros externos a Microsoft a sus servicios. Y, por otro lado, a nivel de responsabilidad de seguridad, Azure AD es un servicio proporcionado por el proveedor y es el cliente el que tiene el deber de gestionar los accesos, con RBAC, y evitar la mala configuración (o por defecto) para evitar accesos indebidos.

- **Microsoft Defender for Cloud**
  Es un recurso de administración de la posición de seguridad en el *cloud* (CSPM), para implementar procesos continuos y automatizados de seguridad y cumplimiento ayudando a prevenir vulnerabilidades y la plataforma de protección de cargas de trabajo (CWPP) que amplía la visibilidad de los recursos para proteger las cargas de trabajo.

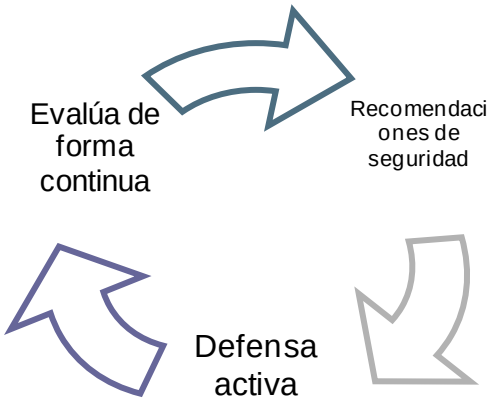


Ilustración 8: Flujo Microsoft Defender

a) **Evalúa de forma continua**
El servicio va evaluando constantemente los recursos que dispone el usuario analizando las posibles vulnerabilidades reportando los resultados con una puntuación según su criticidad (afecta no solo el riesgo ocasionado si no también el número de recursos afectados)

b) **Recomendaciones de seguridad**
Se priorizan los resultados según su criticidad obtenida en el apartado a) y se crea una lista de recomendaciones de cómo actuar para resolver o mitigar la amenaza detectada. También permite la opción de establecer unas directivas de seguridad marcadas y de aplicar un estándar de seguridad.

c) **Defensa activa**
Existe una política por defecto de seguridad que se provee durante el aprovisionamiento de recursos, de manera que se aplica ya una defensa en el estado inicial. Por ejemplo, sucede con la instalación de Microsoft Defender en todas las máquinas virtuales tipo Windows u otros controles de aplicaciones adaptables con listas blancas.

- **Azure DDoS**
Los ataques por denegación de servicio distribuido son uno de los problemas de seguridad y disponibilidad más extendidos según un reporte de Michael Vizard [27].

Es por ello, que Azure con este servicio (junto con otras reglas en las configuraciones y diseño de aplicaciones) permiten una capa de protección frente estos ataques.

Este servicio está integrado en otros pertenecientes dentro de la capa de red virtual, como son: Azure Front Door, WAF y el cortafuegos, entre otros.

- **Azure Front Door**
Es un servicio de CDN (Content Delivery Network) que proporciona acceso rápido, confiable y seguro a un contenido web estático y dinámico global.

De esta forma, el usuario puede acceder a un sitio web alojado en la nube pasando por este servicio con las ventajas que esto conlleva, no tan solo a nivel de seguridad, como lo ya mencionado anteriormente, protección ante ataques DDoS, sino también el control y estabilidad para el sitio web.

- **WAF**
El cortafuegos de aplicaciones web (WAF), protege a los sitios webs contra vulnerabilidades de seguridad habituales, por lo que habilita una capa más de protección perimetral para la aplicación.

Para entrar mínimamente en detalle, con este servicio el usuario añade una capa de protección ante los siguientes posibles ataques indicados en el TOP 10 OWASP [28] –la fundación de software libre OWASP realiza reportes de seguridad anuales con la finalidad de mejorar los estándares de seguridad–:

- Inyecciones de SQL, NoSQL, OS o LDAP
- Autenticación comprometida
- Control de acceso comprometido
- Exposición a data sensible
- Entidades XML externas
- Mala configuración de seguridad
- Cross-Site Scripting (XSS)
- Deserialización insegura
- Componentes con vulnerabilidades conocidas
- Insuficientes registros y monitoreos

- **Cortafuegos de Azure**
  Azure dispone de un servicio de seguridad de cortafuegos (firewall) de forma nativo que protege contra amenazas de forma global en toda la red. Sin embargo, existen distintos niveles (SKUs) con distintos costes, partiendo del básico, estándar y premium.

- **Azure Key Vault y HSM**
  Estos dos servicios de protección de claves están relacionados, la diferencia reside en como las protege criptográficamente, a nivel de software o hardware (aunque en ambos casos en reposo están protegidas por hardware con HSM).

  Con el uso de estos dos servicios –que dependerá de la necesidad del usuario para seleccionar uno de los dos, con mayor o menor coste– el almacenaje de claves, secretos y certificados se realiza de forma segura con mecanismos de accesos controlados con identidades y roles.

- **Puerta de enlace VPN**
  Usando puertas de enlace el tráfico viaja cifrado entre la red (o redes) virtual de Azure y una ubicación local por un medio inseguro como puede ser Internet.

  Existen 3 tipos de conexiones, VPN de sitio a sitio, VPN de punto a sitio y VPN entre redes virtuales. La elección de cada una de ellas dependerá de la necesidad especifica. Por ejemplo, en el primer caso, se utiliza habitualmente para conectar una red *on-premise* con la red virtual de Azure. En el segundo caso, está más enfocado a la conexión entre puntos de accesos –un teletrabajador con un cliente VPN– hacia la red de Azure.

- **Puerta de enlace de aplicación (APP GW)**
  Es un balanceador de carga de tráfico web en capa de transporte (4) OSI, para administrarlo hacia las aplicaciones.

Además, permite añadir de nuevo una capa de seguridad con un WAF y una protección DDoS ante la aplicación o aplicaciones a las que apunta finalmente.

- **Enlace y dispositivo privado**
  Este servicio permite acceder a servicios PaaS (una base de datos SQL) por un punto de acceso de conexión privado de la red virtual, de manera que se bastiona la conexión entre un lado y el otro con un vínculo privado por la red.

  Además de la ventaja clara de seguridad –evitando *sniffeo* de red–, debido a su naturaleza, también existen mejoras en el ancho de banda y la protección contra la perdida de datos por posible ruido.

- **Red Virtual (Vnet)**
  Es un bloque de creación fundamental de una red privada en Azure y el punto central de muchos de los servicios mencionados anterior y posteriormente en esta definición de servicios.

  Permite, por lo tanto, la comunicación entre recursos de forma segura entre ellos, con internet y con otras redes.

- **Grupo de seguridad en red (NSG)**
  Se utiliza para filtrar el tráfico de red entre los recursos de Azure de una red virtual mediante reglas de seguridad similares a "iptables" como sucede en Linux.

  Para ello se puede indicar distintos niveles de gestión, desde un simple recurso, grupo de recursos hasta una red virtual entera. Y permitiendo o denegando por distintas prioridades, orígenes o destino, protocolo usado o puertos.

- **Bastion**
  Es un servicio PaaS que permite la conexión segura a una máquina virtual mediante el explorador y el portal web de Azure (o por un cliente RDP o SSH).
  De esta manera, con Bastion se protegen los puertos de RDP/SSH de manera pública con una red privada virtual gestionada por Azure. El esquema se muestra en la ilustración 9:

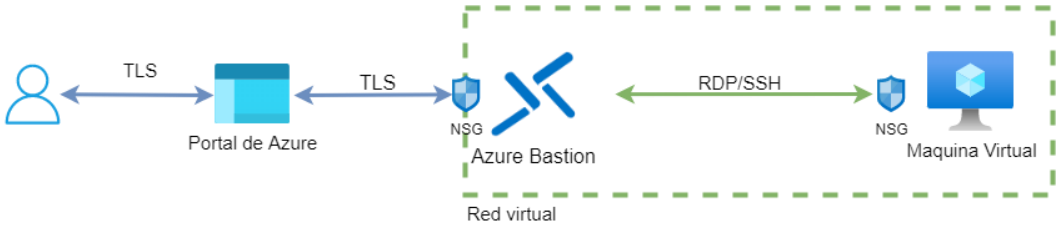


Ilustración 9: Bastion

- **Políticas de seguridad**

Tal y como indica su nombre, este servicio ayuda a aplicar políticas mediante reglas previamente definidas, las directivas. Estas se asignan a distintos grupos de administración, subscripciones o recursos.

3.3.2. Detección de ataques

Servicios enfocados al análisis constante de amenazas y vulnerabilidades encontradas en los entornos, de tal manera que se identifiquen actividades sospechosas para que el usuario deba gestionar. En algunos casos ya tratados en el apartado anterior, protección y prevención, debido a su versatilidad de acciones.

- **Microsoft Sentinel**
  Es un servicio nativo de Microsoft Azure que permite la administración de eventos e información de seguridad (SIEM) y la respuesta automatizada de orquestación de seguridad (SOAR) de forma inteligente con los siguientes pasos:

  a) Recopilación de datos de todos los usuarios, dispositivos, aplicaciones y toda la infraestructura alojada en el *cloud*.
  b) Detección amenazas con análisis inteligentes.
  c) Investigación de las amenazas detectadas y actividades sospechosas.
  d) Respuesta a los incidentes con orquestación y automatización de tareas integradas.

- **Visor de red de Azure (Azure Network Watcher)**
  Una serie de herramientas de supervisión, diagnosis de métricas y registros de los recursos en la red virtual de Azure, pudiendo observar el espectro de red de por ejemplo una máquina virtual.

- **Microsoft Defender para IoT y contenedores (Kubernetes)**
  Existen módulos específicos de Microsoft Defender para supervisar tipos de dispositivos IoT (internet of things) o contenedores como *kubernetes* que examinan las imágenes que se alojan y ejecutan en Azure.

3.3.3. Respuesta ante incidentes

El objetivo del tercer punto es como proceder ante un incidente reportado, estos servicios extraen datos de registro para analizar actividades sospechosas y puedan generar un reporte de las tareas realizadas y por realizar.

- **Logs y métricas de Azure Monitor**
  Es un servicio que recopila y analiza la telemetría de los entornos y recursos alojados (dentro y fuera) del *cloud*. De esta manera, el usuario conoce el rendimiento de las aplicaciones y los posibles problemas o cambios que ha habido que puedan afectar al servicio.

Existen los siguientes servicios con capacidad de gestionar este monitoreo de forma nativa graficados en la ilustración 10:

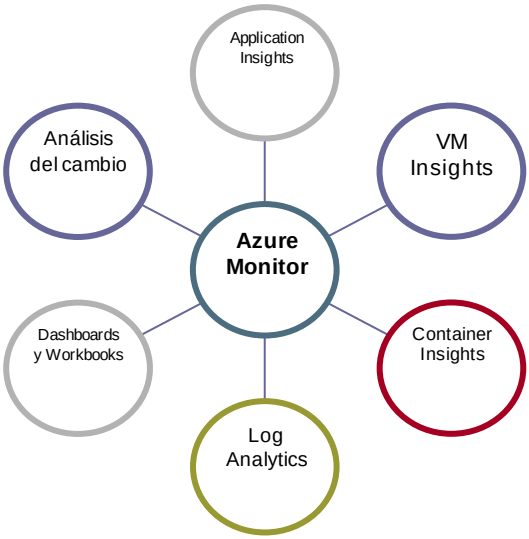


Ilustración 10: Servicios con Azure Monitor

4. Implementación de seguridad

En este punto, una vez descritos los principales servicios que Azure proporciona para contrarrestar las principales amenazas mencionadas en el apartado 2, se implementará un entorno o infraestructura desde el principio que cubran los requisitos generales de seguridad. Para ello, se analizará en el siguiente apartado un caso genérico de organización tipo *startup*, con un nivel de experiencia y unas capacidades formativas básicas en el *cloud*.

4.1. Situación inicial

Para proponer un buen ejemplo de la correcta implementación de seguridad en el *cloud* se debe mencionar el motivo por el cual se selecciona de partida una organización tipo *startup* y cual suele ser su situación inicial.

Este término, se emplea a una empresa con poco tiempo de creación, con una base tecnología o su actividad es innovadora. Y por ello, el motivo es evidente, se parte de una situación inicial nula por lo que la creación del entorno tecnológico de cero, pudiendo por ello, arrancar con un estándar de seguridad por defecto (término acuñado como *security by default*).

Además, estudios conocidos como el publicado recientemente en el País (octubre 2022) indica que el 92% de las empresas tipo startups se mantienen activamente desde su creación entre el año 2016 y 2021 frente a las empresas tradicionales sin base tecnológica [29]. Por lo que es interesante analizar este tipo en concreto.

En el caso ejemplo, se trata de una startup centrada en el desarrollo y distribución de soluciones de ciberseguridad para otras empresas, la cual cuenta con una estructura organizativa pequeña dada sus características como startup, se incluye organigrama ilustrativo en la ilustración 59 de los anexos.

En las siguientes pruebas se simulará la creación de la infraestructura por parte del encargado de operaciones de infraestructura (usuario6.3, en el organigrama) que realizará en gran parte todos los roles o funciones descritos anteriormente en el apartado 3.2, junto con el CEO (usuario1).

Para llevar a cabo la implementación de la infraestructura, es necesario listar los **requisitos tecnológicos** del negocio para dar el servicio a sus clientes, en las siguientes tablas esquema 3 y 4:

Requisitos base internos	
Objeto	Descripción
Usuarios acceso por VPN	Para que cada trabajador pueda llevar a cabo sus tareas dentro de la red empresarial fuera de la localización
Acceso con usuario y contraseña (y doble factor de autenticación)	De manera que los trabajadores necesiten una contraseña y otro mecanismo para acceder. (algo que conocen, algo que tienen y/o algo que son)
BD empleados	Tener un control de los trabajadores, horarios, vacaciones, etc.
Web intranet	Poder tratar los datos alojados en la BD de empleados con una web accesible solo en la red empresarial de la startup
Unidad de red compartida	Almacén de archivos para los trabajadores

Tabla 3: Requisitos internos

Requisitos base negocio	
Objeto	Descripción
Aplicación 1	Aplicación web que comercializa la startup, sistema EDR (solución de seguridad diseñada para detectar y bloquear amenazas a nivel de dispositivo) tipo SaaS para sus clientes
BD clientes	Almacena datos de clientes necesarios para dar servicio a su aplicacion1

Tabla 4: Requisitos negocio

4.2. Creación del entorno en Azure

4.2.1. Cuentas de usuario

Como primer requisito interno e igual que cualquier entorno empresarial, donde los empleados trabajarán para desarrollar su trabajo y crearán las unidades de negocio, es necesario crear una gestión de usuarios en el entorno de Azure para acceder a los

distintos servicios, aplicando un rol de acceso necesario para cada equipo y aplicación, es decir, aplicando una serie de buenas prácticas.

Para ello, con la cuenta administradora del directorio de Azure para la startup, durante todo el trabajo de investigación, directorio: UNIVERSITAT OBERTA DE CATALUNYA.

Microsoft recomienda no crear distintos directorios para separar usuarios y subscripciones, sino que esta se deberá hacer con roles RBAC que se analizarán como indica en su documentación de identidades [30]. Dicho esto, a partir de ese directorio se crearán los siguientes objetos:

a) **Usuarios**: se debe seleccionar si son usuarios internos de la organización o invitados, esta distinción se realizará dependiendo del tipo de usuario que se quiera añadir. En general, para los trabajadores del organigrama (internos) se añadirán como miembros, mientras que los trabajadores temporales u externos se deberán añadir como invitados.

   En el caso de tipo de usuario invitado, este puede partir de un dominio distinto al usado por la startup, por lo que, una primera medida a tomar es la de analizar la necesidad de gestionar este tipo de usuarios si en el caso de esta organización todo el personal suele ser interno. Y, para ello, desde los ajustes de usuarios, se limitarán los siguientes parámetros de las ilustraciones 11 y 12:

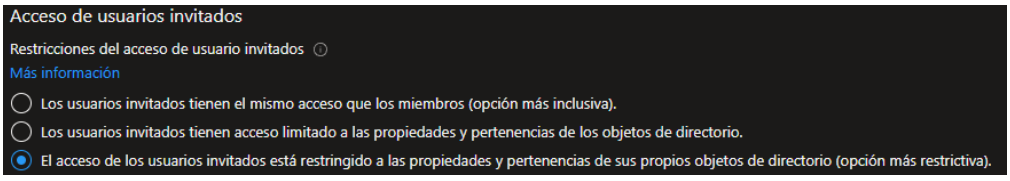


Ilustración 11: Acceso invitados

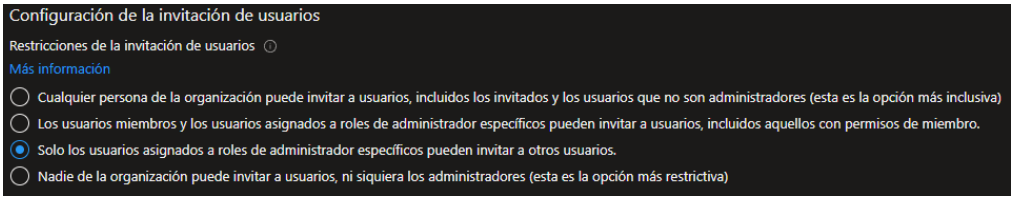


Ilustración 12: Invitaciones a usuarios

   Y en caso de necesidad, por ejemplo, la realización de algún proyecto con una consultora externa cabe la posibilidad de permitir el envío de invitaciones a los dominios especificados:

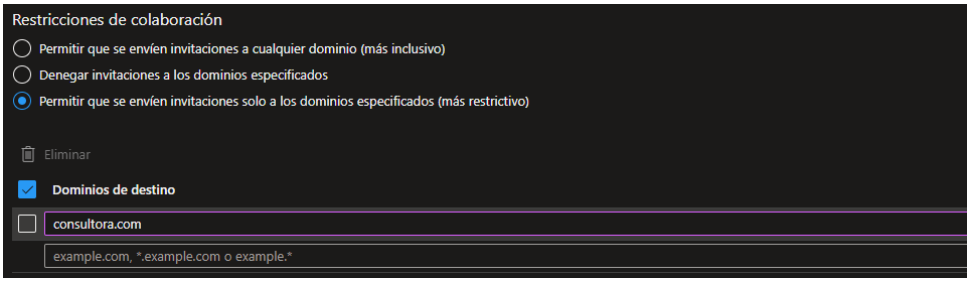


Ilustración 13: Invitación dominios específicos

Además, el usuario0, con el que se ha creado el directorio y los siguientes usuarios –cuya capacidad es de administrador global– debe limitarse su uso para ocasiones puntuales, serán los llamados "cuentas Break-the-Glass" por su similitud al mensaje de "romper en caso de emergencia". Y en su lugar, usar usuarios nominales sin rol de administrador global como se ilustra en la imagen 14.

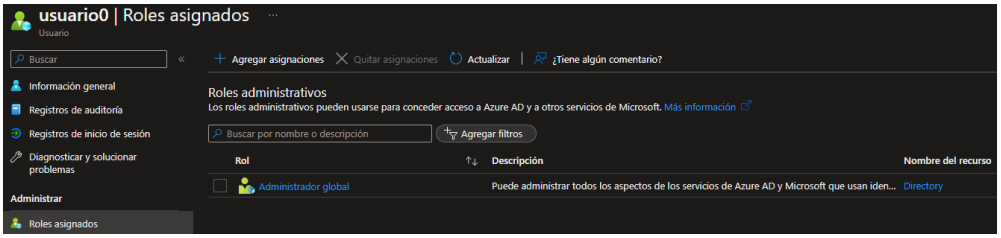


Ilustración 14: Rol global admin

b) Grupos de seguridad: para poder gestionar usuarios que corresponden a los mismos tipos de roles y equipos se puede crear grupos en los que se incluyan a estos mismos y se gestionen los accesos de forma centralizada.

Para facilitar esta gestión, se crearán directivas de nomenclatura, de tal manera que el administrador pueda controlar cada grupo para cada permiso requerido por equipos, ilustración 15:

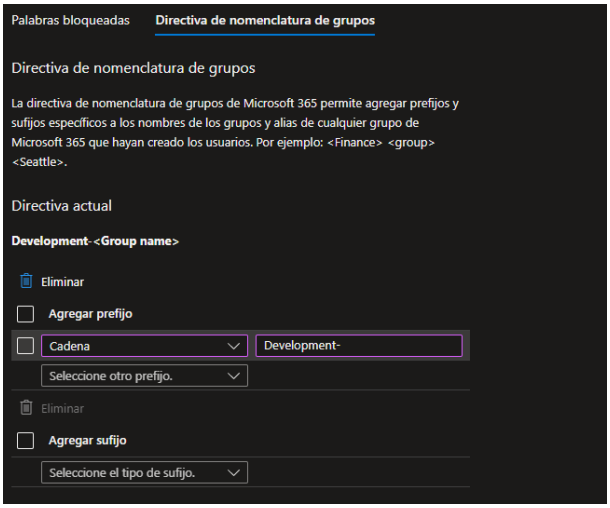


Ilustración 15: Política de nomenclatura de grupo

c) Políticas de acceso condicionales

- **MFA**: una restricción muy importante es la de habilitar el doble (o múltiple en este caso) factor de autenticación. De esta manera, el trabajador deberá utilizar un método de acceso a parte del usuario y contraseña (algo que conoce), junto con algo que tiene (su teléfono móvil o una llave de hardware) y algo que forma parte del trabajador (por ejemplo, huella biométrica dactilar).
  Se puede configurar con una política nueva que aplicará a todos los usuarios del directorio, por grupos o a usuarios concretos. Es recomendable realizarlo a todos los usuarios.

- **Horario de logins:** de momento, no existe la limitación, pero es posible auditar los accesos y de esta manera controlar el acceso de los usuarios en lapsos de tiempo no permitidos, por lo que se considera su automatización en el punto 5.

- **Lugar de login**: es posible bloquear o permitir el acceso por país de origen o rango de IP, de esta manera, siendo una pequeña limitación dada la posibilidad de *bypaseo* (saltarselo) mediante proxys o VPNs como se filtra en la imagen 16.

  En los anexos se ha realizado una comprobación de esta política accediendo a un usuario desde una localización distinta a España, ubicado en la pág. 92.

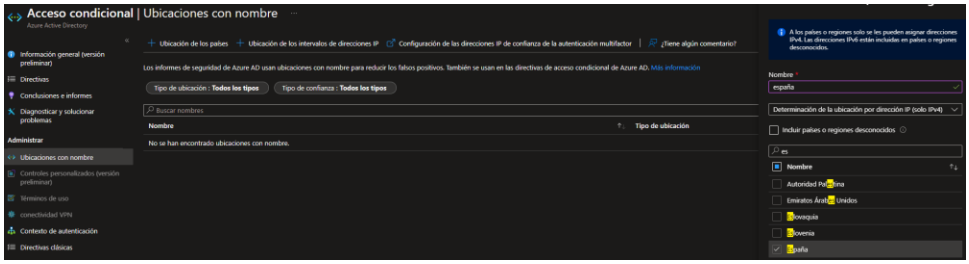


Ilustración 16: Filtro por ubicación de login

4.2.2. Roles de acceso

Para acceder a cualquier recurso (scope), se necesita explícitamente garantizar dicho acceso mediante el control de acceso con roles (RBAC) mencionado en numerosos apartados anteriores.

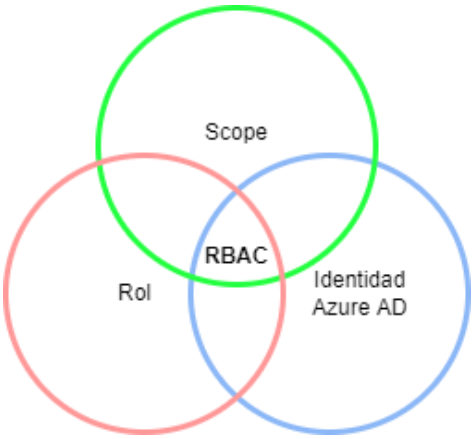


Ilustración 17: Dia. venn RBAC

Scope: es el recurso objetivo por dar permisos (subscripción, grupo de recursos, recurso o sub-recurso concreto). El *scope* actúa de manera jerárquica, por lo que, si por ejemplo se asigna a nivel de subscripción, todo lo que le precede tendrá ese rol.

Identidad: el sujeto al cual dar permisos (usuario, grupo o aplicación). En caso de que un trabajador salga de la startup este se sincronizará con el Azure AD y no podrá acceder al recurso (a pesar de no haber eliminado el permiso explícitamente)

Rol: basándose en una lista de acciones –por APIs– permitidas o denegadas del recurso en cuestión estas se pueden aplicar habilitando o deshabilitando capacidades hacia el recurso. Por ejemplo, para una *webapp*, usando la siguiente lista, o en concreto, wildcard, "Microsoft.Web/*" se habilita cualquier acción hacía ese recurso, de manera que se crea un rol personalizado. Y, sin embargo, existen roles ya previamente definidos: lector, propietario, etc. para evitar esta tarea.

En combinación a los 3 puntos, se gestiona un rol de accesos, sin embargo, existen diversas consideraciones a tener en cuenta durante la administración para evitar posibles amenazas de seguridad:

a) **Evitar la manipulación de cuentas**

Debido a esta acción un usuario malintencionado puede crear un rol –utilizando letras similares para su creación, lo que se conoce como *homoglyphs* https://gist.github.com/StevenACoffman/a5f6f682d94e38ed804182dc2693ed4b – que parezca inofensivo pero que está aplicando mayores privilegios, por lo que se **debería limitar los permisos para crear roles al administrador global** únicamente.

Se puede reproducir de la siguiente manera, el usuario mal intencionado con permisos elevados puede crear un rol personalizado, ilustración 18, cuya plantilla se muestra en la 19:

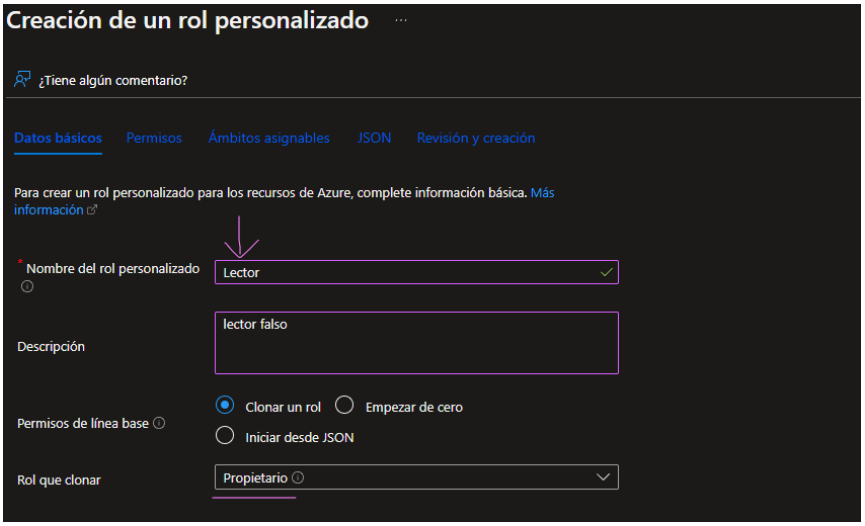


Ilustración 18: Rol malicioso

```
{
    "properties": {
        "roleName": "Lector",
        "description": "lector falso",
        "assignableScopes": [
            "/subscriptions/54dfc161-80fb-
4ff3-b33e-706d401a84c1"
        ],
        "permissions": [
            {
                "actions": [
                    "*"
                ],
                "notActions": [],
                "dataActions": [],
                "notDataActions": []
            }
        ]
    }
}
```

Ilustración 19: Plantilla rol malicioso

Se puede comprobar en la ilustración 20 que en esta situación se ha asignado dos tipos de lectores al recurso, a pesar de que uno es el falso con permisos de propietario.

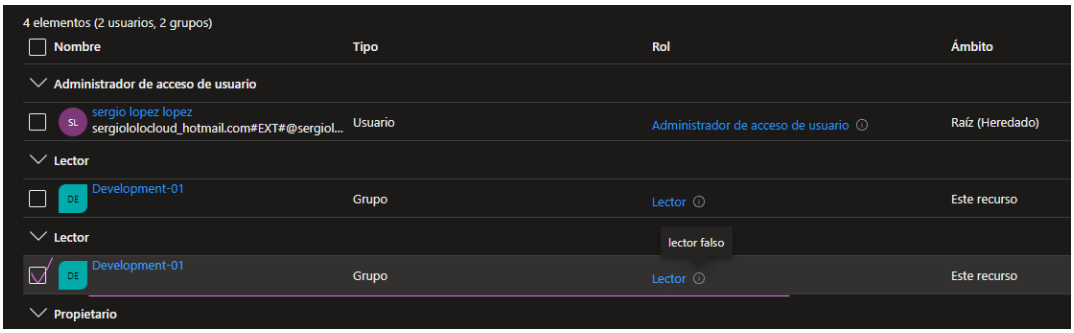


Ilustración 20: Asignación rol malicioso

b) Limitar roles por grupos de usuarios

La siguiente tarea por realizar es la de gestionar los permisos a los recursos para cada grupo en función de la necesidad y no otorgar mayores permisos a los no necesarios para la función.

Es una tarea difícil de gestionar de forma automática, por lo que el administrador de la startup deberá analizar de manera exhaustiva los requisitos necesarios de cada grupo al recurso.

Al equipo de desarrollo, se le deberán otorgar permisos de lectura a la subscripción de DEV (desarrollo), y únicamente permisos de contribución a los recursos jerárquicamente sub-nodulares a los que deban necesitar, por ejemplo la *webapp* de la aplicación como se muestra en la ilustración 21.

Ilustración 21: Asignación rol a grupo

c) Uso de identidades gestionadas y service principals

Para el correcto funcionamiento del flujo de negocio de las aplicaciones alojadas en Azure, se usarán *Managed identities* (identidades gestionadas) para gestionar los accesos entre recursos, y en cualquier otro caso, hacer uso de *service principals*.

Esto quiere decir que, si por ejemplo la *webapp* debe acceder a una cuenta de almacenamiento para recuperar un fichero que se solicita, este acceso se puede otorgar mediante la identidad de la propia web hacía la cuenta de almacenamiento con un rol necesario como se muestra en la figura 22.

Ilustración 22: Asignación rol a identidad

En otras situaciones, si la aplicación se debe impersonar, por ejemplo, en una base de datos, es posible realizarlo con el uso de un *service principal* (ilustración 23) que simula un usuario y contraseña, con un secreto (ilustración 24), en este caso para realizar una consulta, por ejemplo.

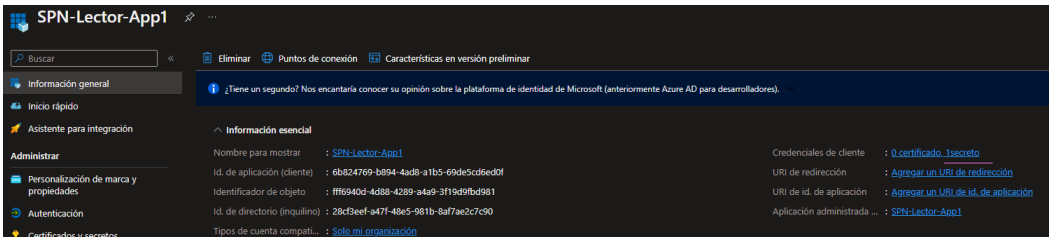


Ilustración 23: Service Principal Name para SQL

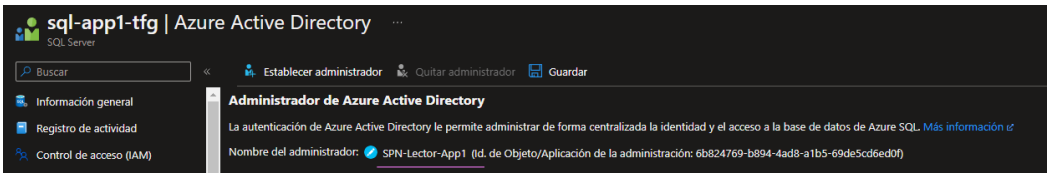


Ilustración 24: Asignación admin SQL al SPN

4.2.3. Redes virtuales

El siguiente punto que se debe abordar es la securización durante la creación de las redes virtuales de la infraestructura de la startup. Cuáles son las buenas prácticas que seguir para delimitar el alcance entre todas ellas y el *workstation* del trabajador.

Azure brinda distintas soluciones para que el tráfico del cliente esté conectado a la WAN (red de área amplia) de Microsoft, como es el caso de Azure *ExpressRoute* –habilita la conexión privada entre la red de la organización y la de Microsoft para ofrecer mayor seguridad, confiabilidad y menor latencia–, pero dado el tamaño de la startup y el gran coste del recurso, se opta por un *peering* simple entre las redes (local y virtual de Azure) con el servicio de VPN.

En el momento en el que se crea la primera red virtual, cualquier recurso *deployado* dentro de la misma estará aislado de cualquier otra si no está explícitamente configurado mediante un *peering*, por lo que se debe analizar qué necesidad hay para crear esta relación.

Por ejemplo, en el caso de entornos de preproducción o producción no debe de existir ningún *peering* directo hacía ninguna otra red que no sea la habilitada para acceder por máquinas de salto (*steping stones*). Sin embargo, como se verá en la implementación, usando maquinas bastionadas, explicadas anteriormente, se puede evitar la exposición hacía ninguna otra red (ni públicamente).

a) VNet y Subnets

Una buena práctica para segmentar las redes es crear una red por cada subscripción (desarrollo, preproducción y producción) a no ser, que existan distintas aplicaciones en el negocio y se necesite aislarse entre ellas. En el caso de la startup, existe una aplicación duplicada en las distintas fases hasta producción, aplicando metodologías *devops*.

Durante el proceso de creación se deben agregar las subredes necesarias dentro del rango seleccionado. Para ello, una buena *praxis* es crear una subred por cada servicio o incluso por tipo (*frontend* y *backend*), de esta manera existe una división lógica entre los recursos.

Además, en este punto se elegirá disponer de una subnet bastionada para crear el acceso seguro web hacía un puerto de administración (RDP/SSH) de las maquinas que se desplieguen en la red.

Por defecto, se habilita en la protección frente a ataques DDoS. Pero se debe configurar a mano el firewall en una subred dentro de la red virtual que se cree, como se muestra en la ilustración 25.

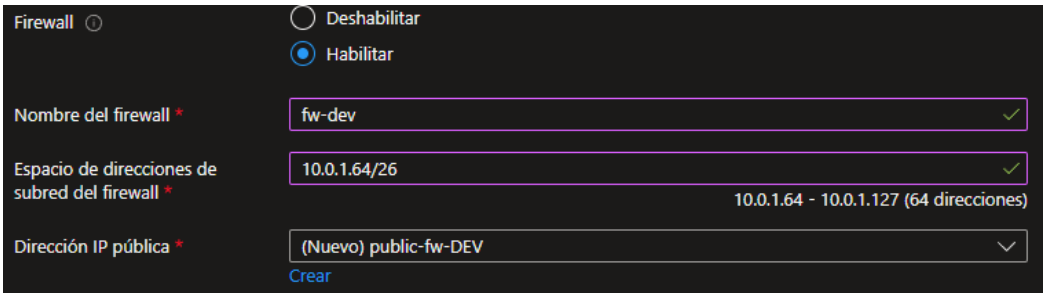


Ilustración 25: Direccionamiento subred firewall

b) VPN

Para conectar de forma privada al trabajador con la red de desarrollo recién creada, se utilizará una VPN Point-to-Site (imagen 26), es decir, se creará una conexión entre la red virtual de Azure y un cliente VPN instalado en el usuario y accesible con credenciales.

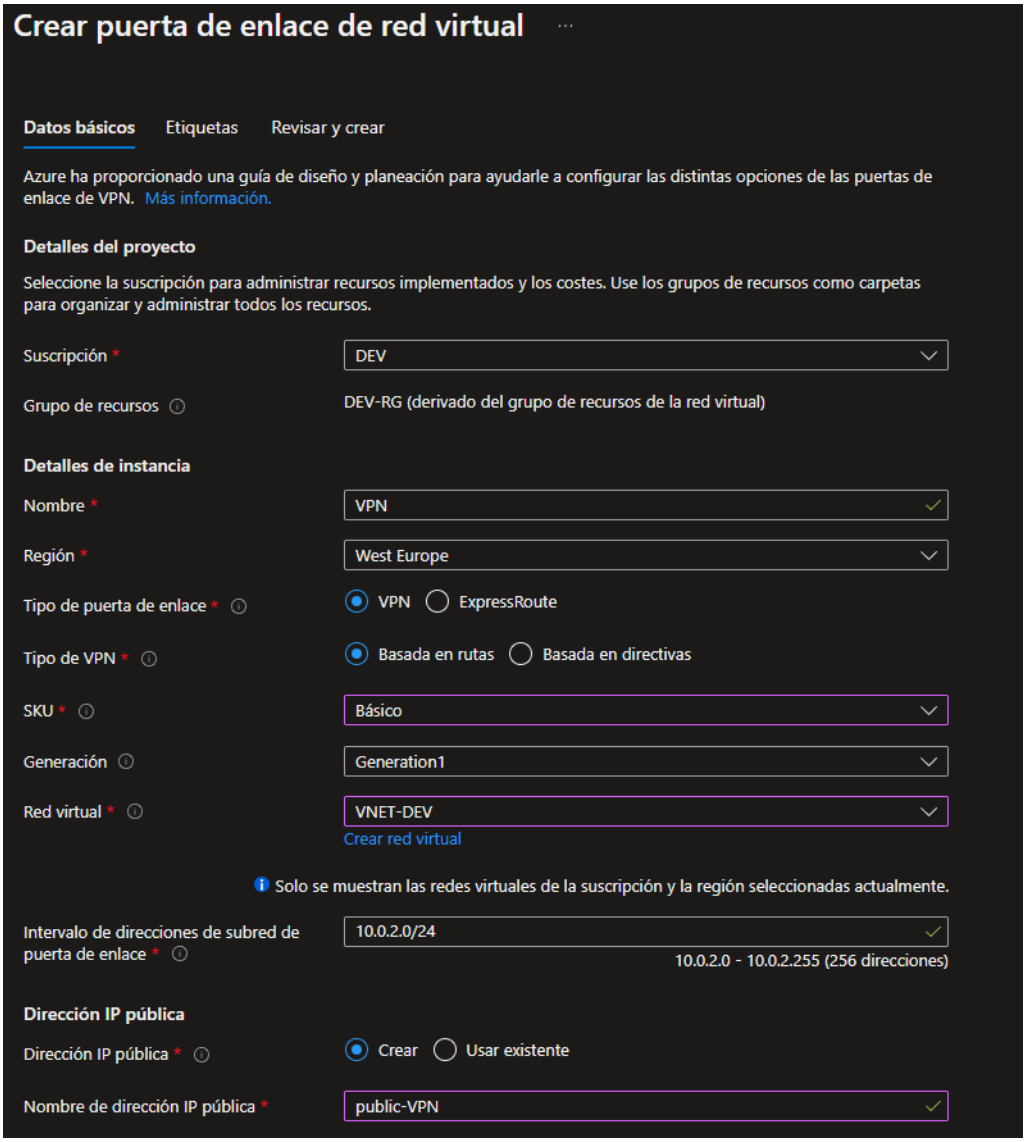


Ilustración 26: Creación VPN P2S

De esta manera, todo el tráfico de red del dispositivo del trabajador estará circulando por internet de forma cifrada utilizando el protocolo IKEv2 (Internet Key Exchange versión 2) u OpenVPN dado que es lo recomendable para una conexión VPN segura y estable frente a SSTP.

c) Network Security group

Similar a una ACL (lista de control de accesos), con la NSG se puede gestionar el tráfico de entrada y salida de los recursos o incluso de toda la red. Las especificaciones son las siguientes:

- **Acción**: Denegar / permitir
- **Origen y destino del tráfico**: dirección IP (o múltiple) / servicio (o grupo de servicios)
- **Protocolo**: TCP / UDP / ICMP / ESP / AH o todos
- **Rango de puertos**
- **Prioridad**: 100<4096

Por defecto, los NSG permiten el tráfico saliente hacía Internet desde cualquier puerto, por lo que se deberá valorar la necesidad de conectividad hacía el exterior dependiendo de, por ejemplo, la función de la máquina virtual actuando como *backend*.

d) Endpoints privados

Mediante el uso de endpoints privados, el recurso seleccionado obtendrá una IP privada dentro de la VNet indicada, permitiendo con ello el tráfico interno entre los servicios, está por ello debe ser la opción que realizar en cualquier servicio que NO deba estar expuesto públicamente a Internet.

Para crearlo se indicará la ruta que el tráfico de Azure debe resolver con entradas en Zonas de DNS privadas, de esta manera se resuelve el dominio internamente sin consultarse públicamente.

e) Firewall de Azure y WAF

Aplicar un firewall es esencial, este crea una protección en las capas 3-7 del modelo OSI recomendadas para controlar el tráfico de la red. Y para ello, la startup tiene que valorar el coste-beneficio que se obtiene teniendo en cuenta la capacidad de FW seleccionada. Otros servicios interesantes que puede ofrecer el FW son las reglas de *port-forwarding*.

En el caso ejemplo, durante la creación de la VNet se ha creado el FW, que, por costes, se ha mantenido una versión básica, donde ya permite habilitar una primera capa de protección frente los ataques, que por defecto no viene activada, como se muestra en la ilustración 27.

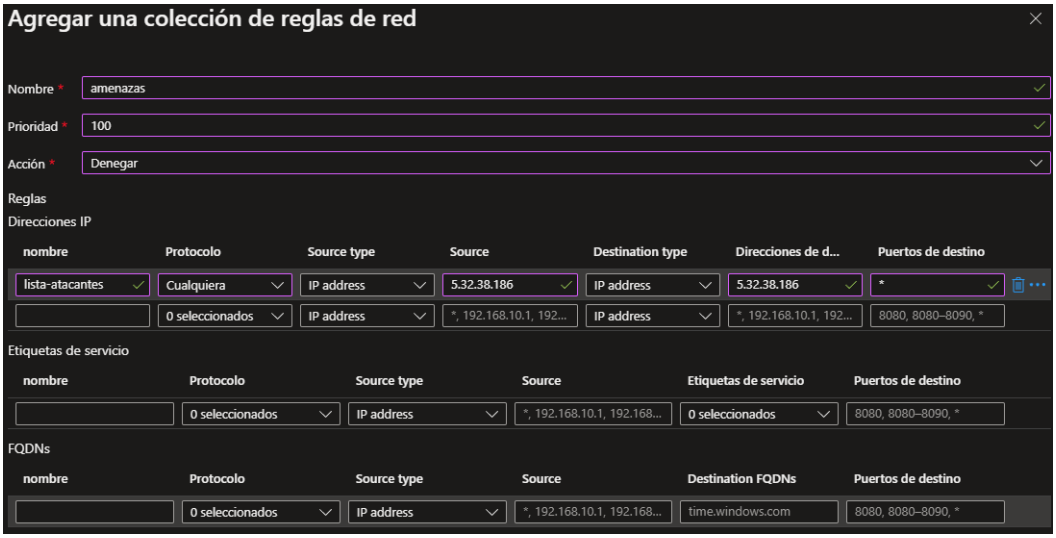


Ilustración 27: Creación manual regla de firewall

El Web Application FW o WAF actúa en la capa 7, protegerá las aplicaciones web frente a múltiples ataques: XSS, Inyecciones SQL entre otros.

4.2.4. Computo

En la mayoría de las organizaciones se hace necesario el uso de máquinas virtuales, esto puede ser debido a varios motivos: la aplicación alojada proviene de *on-premise* y, por ende, esta tiene mayor adaptación dada la similitud de la arquitectura, además existe un modelo de responsabilidad similar (el cliente tiene mayor control) frente a un servicio PaaS o SaaS y en ocasiones, sin un desarrollo profundo previo, no es posible utilizar otro servicio en el *cloud* que no sea una máquina virtual.

En el caso de la startup se creará una máquina de ejemplo –destinada a alojar servicios de cálculos e ingestas de datos de la aplicación principal de la compañía (tipo *backend*)–, con el portal de Azure seleccionando la configuración que más segura acorde a la actividad que llevará a cabo, como se indica en la tabla 5.

Sección	Nombre Config	Valor	Descripción
Datos básicos	Región	*Cualquier zona acorde a la necesidad*	Es interesante seleccionar una zona concreta para todos los recursos por varios motivos: - Movimiento de datos entre recursos se sitúe dentro de la misma zona geográfica (por costes y latencias) - Duplicidad de infraestructura alojada en otra zona para facilitar la creación de un Disaster Recovery en otra zona totalmente distinta
	Opciones de disponibilidad	Zona de disponibilidad	Permite separar el recurso en diferentes zonas físicas de Azure para garantizar la disponibilidad en parches o incidencias
	Tipo de seguridad	Máquinas virtuales de inicio seguro	Protege frente a ataques persistentes y avanzados con arranque seguro y Módulo de plataforma segura virtual (vTPM)
	Reglas de puerto de entrada	Ninguno	Por defecto viene activado el acceso público por el puerto RDP en caso de las Windows o SSH en Linux
Discos	Tipo de disco del sistema operativo	SSD Premium	En máquinas de producción se aconseja para evitar errores en el disco. En el caso de ejemplo no es necesario, se usará SSD estándar. En cualquier caso, viene cifrado por defecto (en reposo).
Redes	IP pública	Ninguno	El acceso a la máquina es exclusivo por red privada. En casos de máquinas tipo *frontend* este acceso debe seguir siendo igual, sólo accesible con balanceadores y *traffic managers*
	Grupo de seguridad de red de NIC	Opciones avanzadas	Se creará un NSG que contendrá las reglas necesarias

Admin.	Identidad	Activado	De esta manera la VM crea una identidad y es usada para acceder a los servicios con RBAC (sólo si es necesario)
	Opciones de orquestación de revisiones	Actualizaciones manuales	Para tener el control, con un *Automation* de las actualizaciones de la máquina. De esta manera se seleccionan los KBs, en el caso de Windows, que instalar.

Tabla 5: Configuración validada VM

Una vez creada la máquina virtual a la que el equipo de desarrollo deba acceder para configurar la aplicación, habilitaremos su acceso mediante el servicio de Bastion, usando el usuario configurado en el momento de creación de la máquina (ilustración 28 y 29) obteniendo un escritorio remoto vía web, para siguientes accesos se debe habilitar el login con los usuarios de dominio de AzureAD:

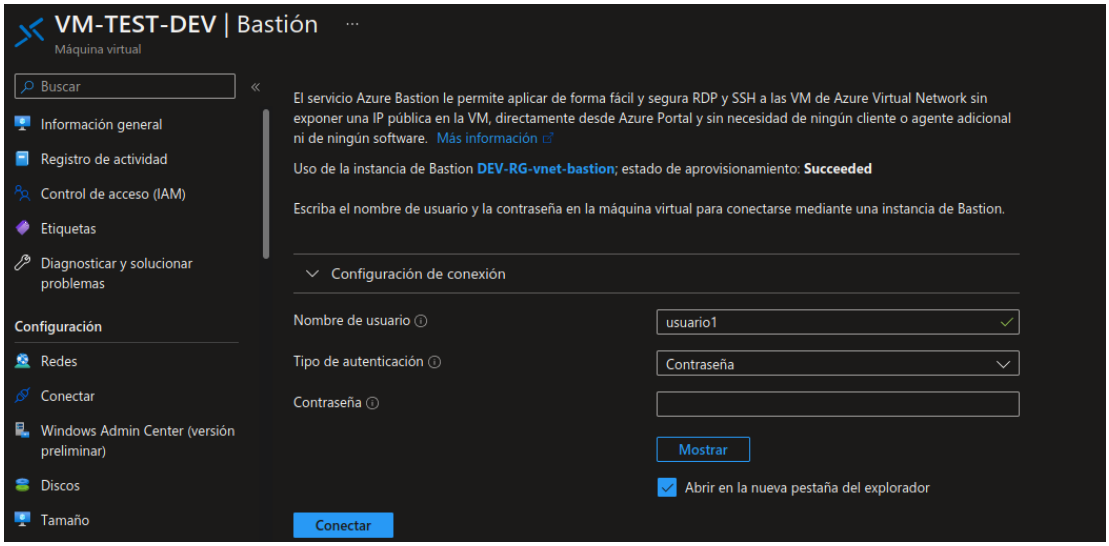


Ilustración 28: Login Bastion usuario1

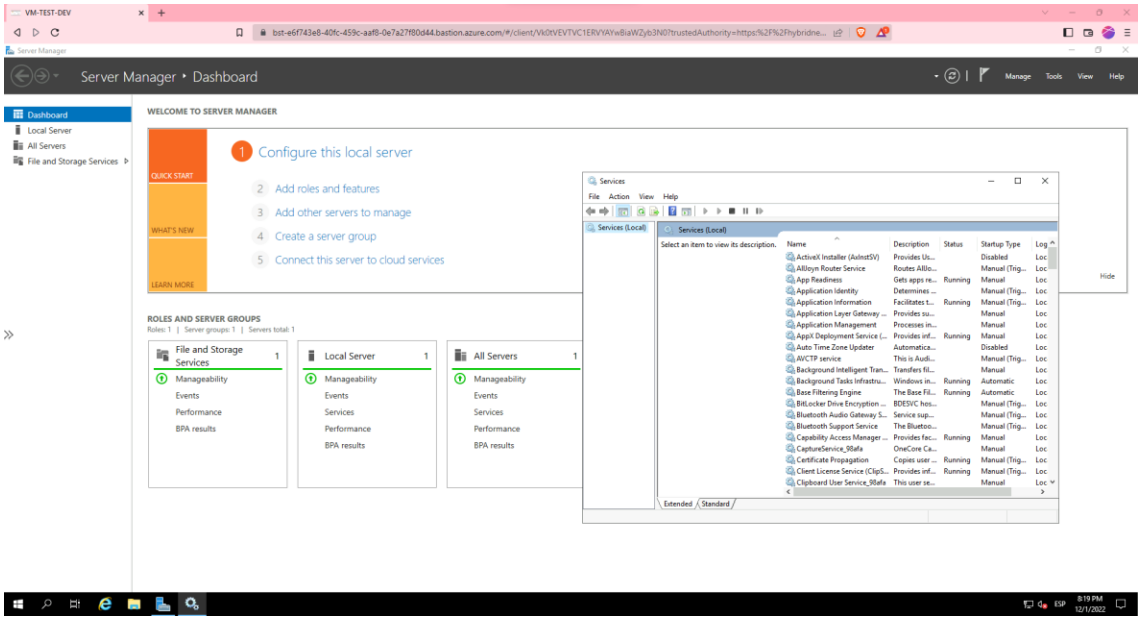


Ilustración 29: Escritorio remoto a través de Bastion

Además, podemos comprobar, esnifando el tráfico con alguna herramienta como *Wireshark,* que la conexión por https con bastion mediante la URL: https://*.bastion.azure.com/#/client/* apunta a la siguiente IP Pública 20.166.37.72, ilustrada en la imagen 30 con una traza *ICMP* por *PING*.

```
PS C:\Users\sergi> ping bst-e6f743e8-40fc-459c-aaf8-0e7a27f80d44.bastion.azure.com

Haciendo ping a bst-e6f743e8-40fc-459c-aaf8-0e7a27f80d44.bastion.azure.com [20.166.37.72] con 32 bytes de datos:
Control-C
```

Ilustración 30: Obtención IP pública a partir del FQDN del Bastion

En dicha captura ofrecida por la herramienta Wireshark, en la imagen 31, solo existen paquetes TLS 1.2 durante toda la comunicación y por ello, toda viene cifrada y segura. De esta manera, como se ha mencionado anteriormente, no es necesario levantar o abrir los puertos de administración, como el puerto RDP (o SSH) en la máquina para poder trabajar con ella, favoreciendo su protección externamente.

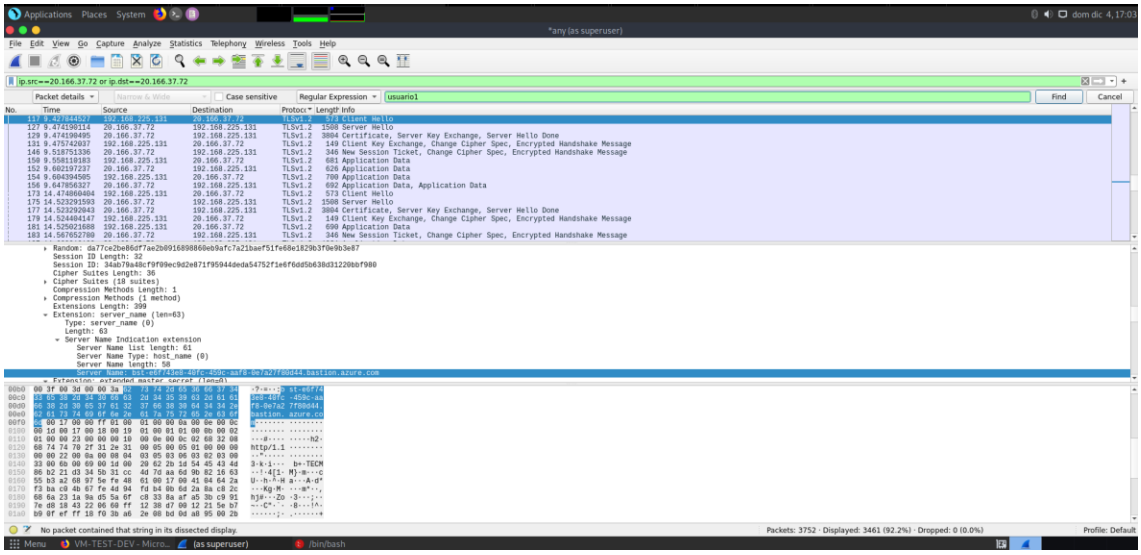


Ilustración 31: Captura Wireshark analizando el tráfico TLS entre local y red bastion de la VM

4.2.5. Servicios PaaS

Cualquier servicio PaaS, por ejemplo, Azure App Service o Azure SQL Server por defecto son alcanzables por cualquier persona que esté dentro del tráfico de Azure, es decir, cualquier cliente del proveedor puede conocer el servicio.

a) Azure App Service

Las aplicaciones Web son otro punto clave dentro de cualquier infraestructura cloud, por lo que se considera interesante administrar algunas de las características que aportan seguridad en la web de la startup.

Por defecto, al crear un App Service se hace de forma pública, es decir, es accesible a todo internet con un dominio propio de Azure, *azurewebsites.net*. Es por ello que, como primer paso, se debe analizar la naturaleza de la aplicación, si estará destinada a desarrollo, si debe ser interna –una intranet– o si será una web pública que deba llevar un dominio y certificado personalizado.

En este punto se van a proponer 3 simulaciones para cada caso, en los que se prima un nivel de seguridad u otro en función de la selección:

- **Aplicación para desarrollo:** en este supuesto caso, Azure ya provee una gran calidad a nivel de seguridad para la aplicación web, esta dispone de un dominio propio de Azure con certificado TLS incluido, copias de seguridad constantes (por defecto cada hora) y dependiendo del plan seleccionado, espacios de trabajos distintos.

- **Aplicación interna:** para este caso, ya existe una intención de aislar la aplicación web por lo que no es posible dejar los ajustes por defecto.

Para ello, seleccionar un dominio personalizado junto un certificado propio es decisión cuestionable, pero no requerida, por lo que para la startup se mantiene la aplicación con el dominio obtenido de Azure. Los enfoques principales de este punto son el aislamiento por la red y el acceso con credenciales.

- **Aislamiento red:** en la aplicación existe un espacio para redes en el que se deben filtrar tanto el tráfico entrante como también el saliente, como se muestra en la ilustración 32 y el esquema de la tabla 6 y como resultado, un acceso no permitido, *ERROR 403* (imagen 33).

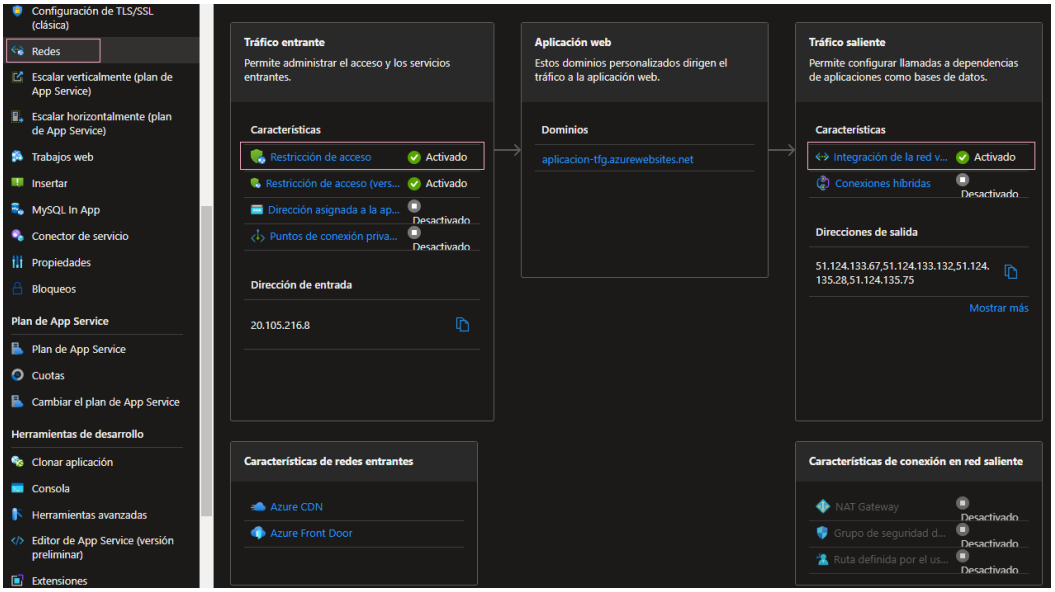


Ilustración 32: Aislamiento red en App

Trafico de entrada	Tráfico de salida
Regla tipo NAT	Integración con la VNet
Puntos conexión privada	-

Tabla 6: Tipo de filtros de red en App

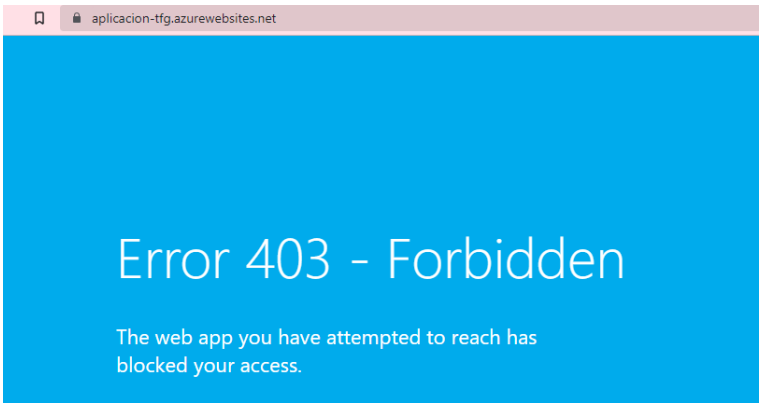


Ilustración 33: Error 403 App

- **Acceso con credenciales:** añadiendo un proveedor de identidades –como el propio Microsoft, Google, Apple, Facebook, GitHub, Twitter u OpenID– es posible administrar el acceso a la aplicación, ilustración 34.

  Para el caso de prueba, si se selecciona Microsoft, el servicio se vincula con los usuarios configurados dentro del Azure AD facilitando la gestión de accesos a la intranet, como se muestra en la imagen 35 y se comprueba su eficacia en la 36.

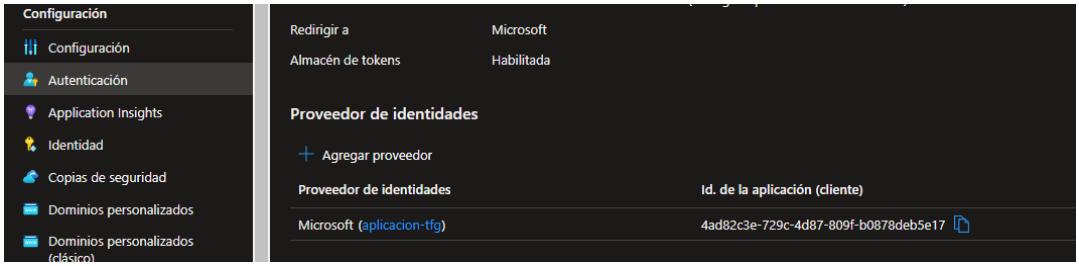


Ilustración 34: Proveedor identidad App

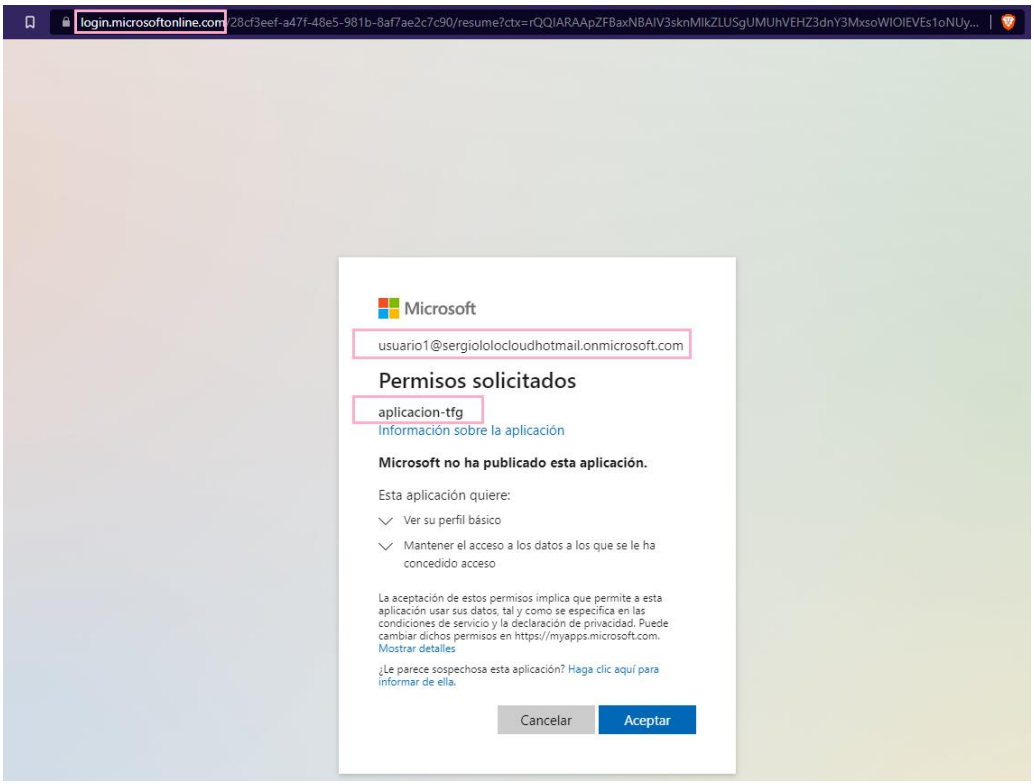


Ilustración 35: Proveedor Microsoft App

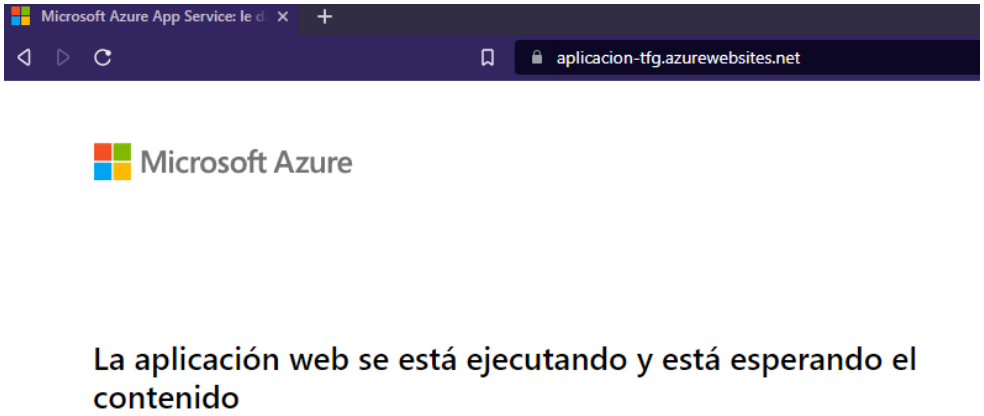


Ilustración 36: Acceso correcto App

- **Aplicación pública:** en el caso de una aplicación a la que la intención principal sea recibir el máximo tráfico de clientes, dado que no se deberá ni limitar por red interna, ni bloquear por acceso no autenticado, los puntos a tratar serán totalmente distintos.

  Para este supuesto, se gestionará un dominio personalizado junto con un certificado validado por una entidad certificadora de confianza (como puede ser el propio Azure) y por otro lado, bloquear el acceso inseguro forzando a HTTPS, imagen 37.

  Asimismo, se limitará el cifrado de TLS mínimos (imagen 37) a 1.2 –a pesar de que exista actualmente la v1.3– y se evitarán los *ciphers* inseguros que por defecto Azure a día de hoy ya están aplicados.

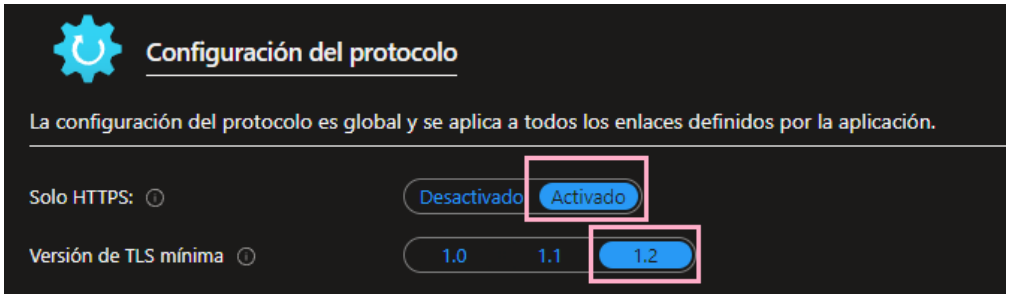


Ilustración 37: Protocolos seguro App

Por otra parte, para garantizar la seguridad en las peticiones entrantes a la aplicación, será aconsejable añadir una capa extra como el servicio Azure Front Door / CDN y/o Application Gateway (puerta de enlace) con posibilidad de limitar a TLS1.3. Además, incluye, si es necesario, la capacidad de Cortafuegos (WAF) como se muestra en la imagen 73 añadida en los anexos.

En todos los supuestos mencionados anteriormente no se valoran las vulnerabilidades mencionadas en apartados anteriores debido a errores en el código o módulos de terceros vulnerables.

b) Azure Key Vault

Este servicio, como se ha introducido previamente, permite almacenar secretos solo accesibles con previa autenticación con Azure AD, por lo que, es indispensable un buen control de usuarios y roles.

Existen distintos permisos para crear un rol personalizado en función de la necesidad, se dividen en permisos para claves, secretos y certificados, y conllevan distintas actividades posibles, desde obtención del valor del secreto, eliminación, copias de seguridad, etc. Para la aplicación web, debida la necesidad de obtención de certificado y un parámetro secreto se le otorga lo siguiente (principio del mínimo permiso necesario) como se muestra en la imagen 38.

Ilustración 38: Directivas acceso KV

Por defecto, al crearse permite el acceso desde cualquier red (con un usuario autorizado), por lo que, se deberá valorar dos casuísticas; deshabilitar el acceso público y crear un punto de conexión privado (lo más restrictivo posible) o seleccionar un listado de redes virtuales a las que debe conectar con el Key Vault y como añadido, permitir el *bypass* de los servicios internos de Microsoft, por ejemplo, una cuenta de almacenamiento (cualquier cliente de Azure tendrá conexión con este KV). Esta última opción es la escogida debida a la necesidad de la startup, consultable en la imagen 39:

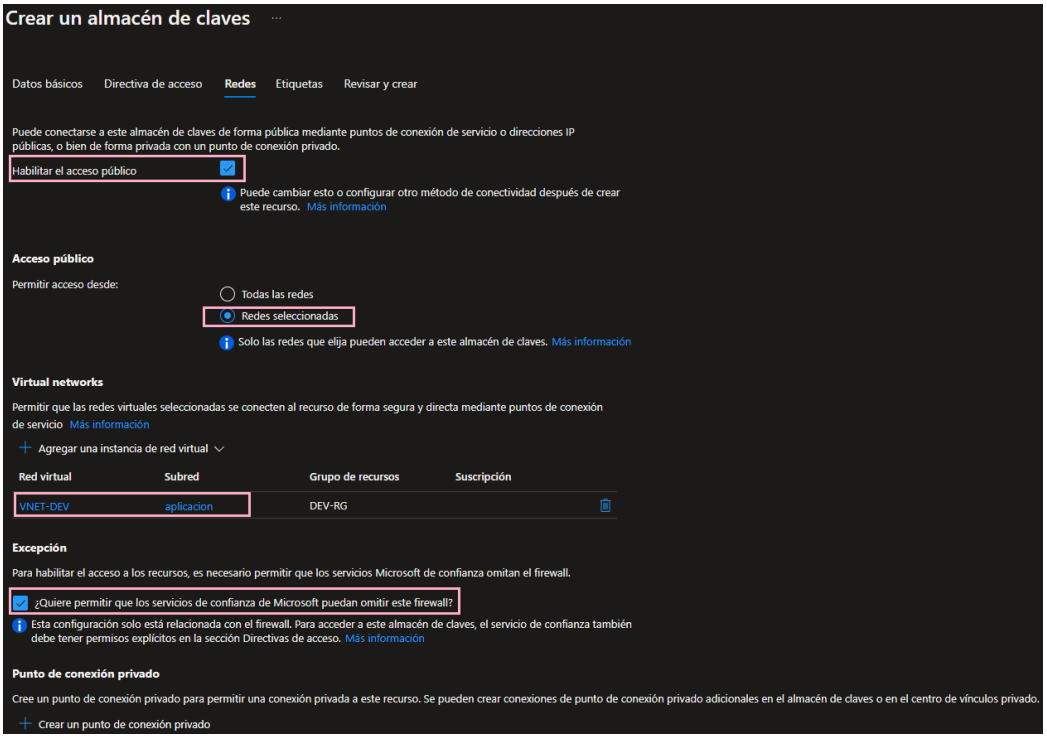


Ilustración 39: Configuración KV

Finalmente, como la app está dentro de los servicios de confianza –a parte de su integración en la misma vnet– y dispone de los permisos necesarios para obtener el secreto y el certificado, se está abstrayendo de almacenar un valor que debe de estar oculto y cifrado, permitiendo a su vez actualizar la versión del propio valor gracias a la capacidad de rotación de secretos del KeyVault, aumentando la seguridad.

c) Azure Storage Account

La cuenta de almacenamiento es un servicio necesario para el negocio de la startup, por un lado, servirá para almacenar todos los archivos compartidos dentro de la organización, con una unidad de red por SMB interna, y por otro lado, para alojar cualquier tipo de archivos de datos desestructurados de la aplicación de clientes utilizando un Blob. Adicionalmente, todos estos datos estarán encriptados utilizando AES 256-bit.

Por defecto, se configuran una serie de parámetros que securizan el almacenamiento de datos: transferencia segura, versión TLS mínima 1.2, encriptación, recuperación de archivos, etc.

Sin embargo, existen un valor a revisar, **Habilitar el acceso público a blobs**, activo por defecto. Esta mala configuración, nombrada como *BlueBleed*, ha permitido una gran fuga de datos de 65.000 compañías y 548.000 usuarios según un reporte de SOCRadar [31].

Por lo tanto, si no existe la necesidad de transferencia de datos para usuarios anónimos (no autentificados) se debe bloquear el acceso. En caso contrario, durante la creación de contenedores de datos –simulando jerárquicamente directorios de archivos–, por defecto, se crean privados y no son accesibles de forma anónima como se muestra en estas imágenes de ejemplo.

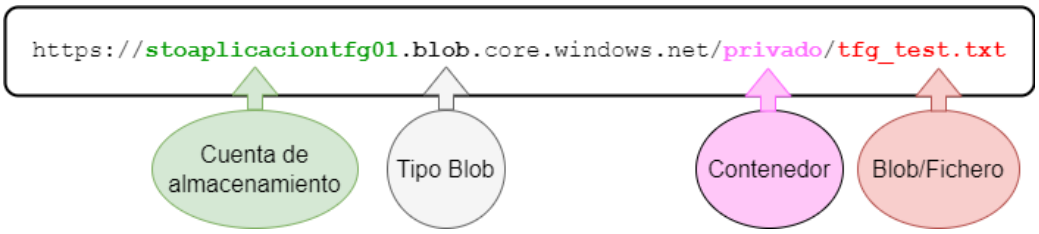


Ilustración 40: Esquema URI de un Blob

Podemos comprobar en la siguiente ilustración 41, mediante una herramienta de testeo de API Rests como *Postman*, la URI de descarga de un blob en un storage privado:

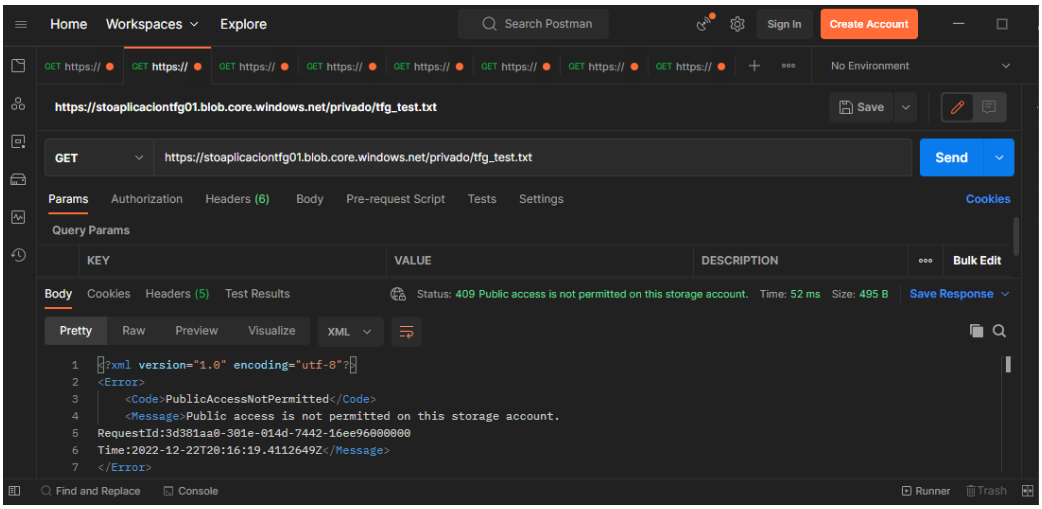


Ilustración 41: Postman storage privado

En caso de realizar una creación de contenedores con acceso público existen dos niveles distintos:

- **A nivel de Blob:** un usuario anónimo disponiendo de la URL directa al blob concreto puede descargar el archivo, sin embargo, no puede listar las propiedades del contenedor en busca de otros blobs.

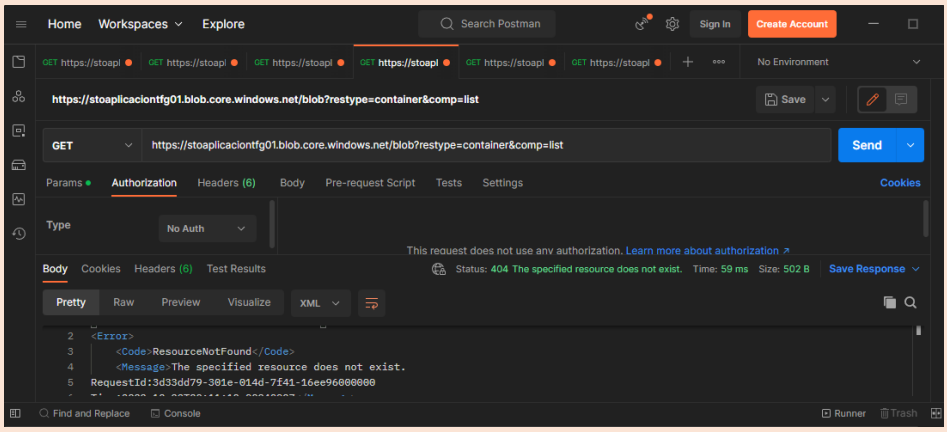
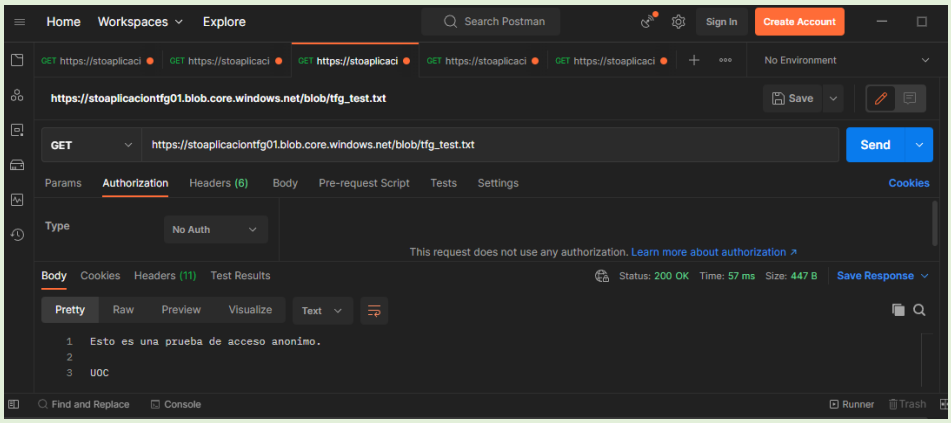
Acción	Captura de la llamada	Resultado
Listar		✗
Descargar		✓

Tabla 7: Test Storage con Blob público

- **A nivel de Contenedor:** un usuario anónimo puede realizar ambas acciones, listar todo el contenedor y descargar los ficheros.

Acción	Captura de la llamada	Resultado

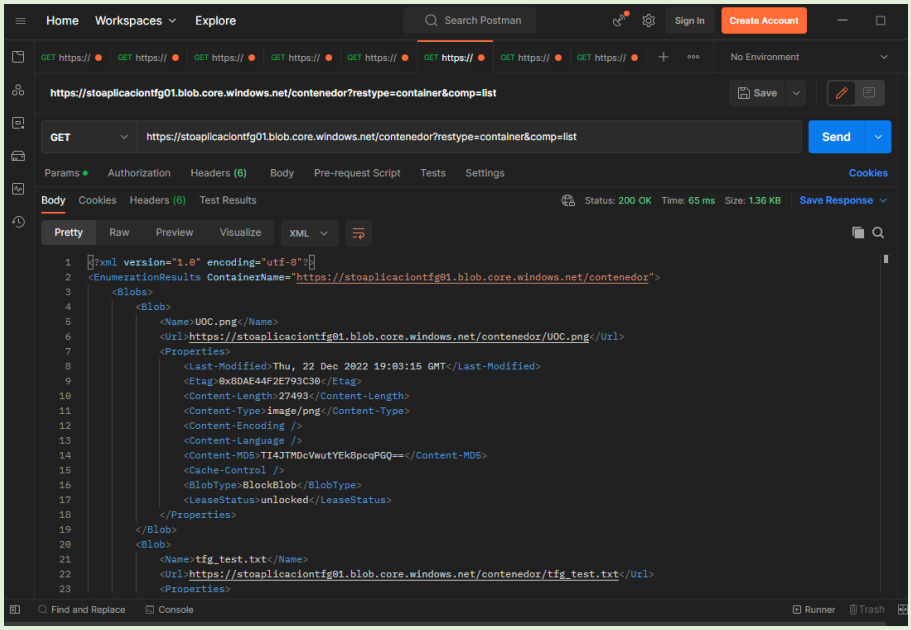
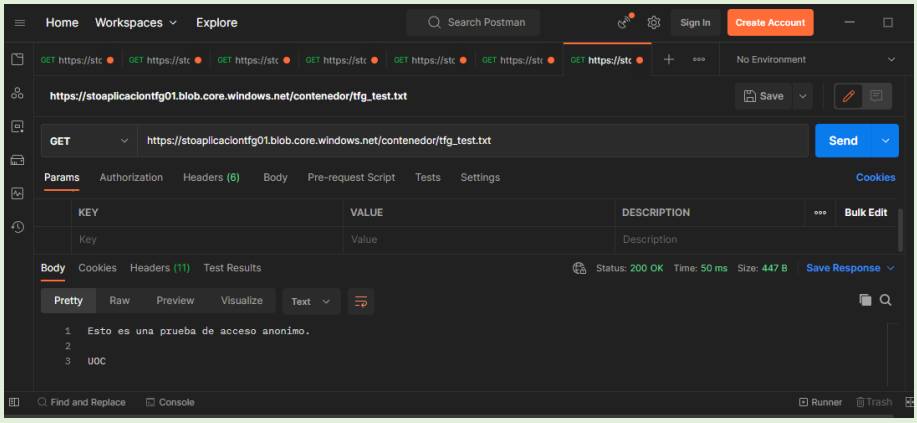
Listar		✓
Descargar		✓

Tabla 8: Test Storage con Container público

Para el negocio de la startup, se desactivará el acceso público, por lo que no habrán storages accounts accesibles de forma anónima, para ello, la mejor manera es mediante plantillas ARM que se analizarán en apartados anteriores –como alternativa, se puede crear una política en *Blueprints* para desactivar la creación pública por defecto–. Además, operador cloud realizará un automatismo, en el apartado 5.2, para auditar todas las cuentas de almacenamiento de la organización para evitar la posibilidad.

Por otro lado, para crear la unidad compartida de red (con SMB versión 3+) que utilizarán los trabajadores de la startup se empleará el servicio de *FileShare* dentro de la cuenta de almacenamiento. En este, hay dos maneras de autenticarse, por Active Directory (on-premise, AAD DS o AAD Kerberos) o con clave de acceso (más inseguro).

Para las necesidades del personal interno, se selecciona la opción por AAD DS dado que es un servicio tipo SaaS para gestionar el dominio de la startup y otorgar acceso a los recursos compartidos a partir del Azure Active Directory mencionado anteriormente. En los anexos se ilustra la configuración necesaria en la imagen 75.

5. Automatización del proceso

Como objetivo final, la idea del proyecto está enfocada en automatizar lo máximo posible las tareas descritas o mencionadas en el apartado anterior.

Luego entonces, en este proceso de automatización no solo se centra en la creación de la infraestructura, siguiendo las buenas prácticas para la securización, si no también, en implementar mecanismos de auditoría y control proactivo de varios vectores de ataque que defender.

Es por ello, que existirán dos puntos distintos dentro de este apartado de automatización, la creación automática de infraestructura necesaria para la startup mediante plantillas de *Azure Resource Manager* (ARM) combinado con llamadas a la API de Azure con módulos *Powershell* en la versión 5.1 y automatización de tareas de control con scripts ejecutados de forma planificada mediante el servicio de Automatización de Azure, ejecutados de igual manera en *Powershell* 7.

5.1 Creación infraestructura

Para facilitar la creación de los recursos de la infraestructura como código (IaC) mencionados en apartados anteriores, se habilita la opción de su creación mediante plantillas ARM propiamente personalizadas por el administrador de la startup, de modo que, tanto la creación como la replicación se realizará de manera sencilla y priorizando la seguridad.

a) Máquina Virtual

La plantilla ARM para las máquinas virtuales sigue la misma estructura que la creación a mano desde el propio portal de Azure analizado en el apartado 4.2.4, con la ventaja de ser desplegada como código reduciendo las configuraciones por defecto que puedan ser vulnerables a ataques.

Para concretar, en esta plantilla existen varios parámetros de entrada que el usuario deberá cumplimentar (en algunos casos con condiciones de validación):

Parámetro	Descripción	Condiciones
nombre	Nombre del recurso	Se añade el entorno
entorno	Entorno o subscripción	
nombreusuarioAdmin	Nombre de usuario de la máquina virtual	

passwordAdmin	Password del usuario de la máquina virtual	Más de 12 caracteres (combinaciones entre letras mayus-minus y números
VersionSO	La versión de windows del sistema operativo de la máquina virtual	Listado de versiones de Sistemas operativos disponibles
tamano	tamaño de la máquina virtual	
localizacion	localización de los recursos	El mismo que el grupo de recursos
VNet-Nueva-o-Existente	Booleano que indica si es nueva (true) o existe (false) la vnet	Crea o usa una red virtual
NombreVNet	red virtual	Se añade el entorno
DireccionamientoVNet	Prefijo de direccionamiento de la VNet	
DireccionamientoSubNet	Prefijo de direccionamiento de la Subnet	
RGVNet	grupo de recursos de la red virtual	
NombreSubnet	SubNed	

Tabla 9: Plantilla parámetros VM

Durante la creación de la máquina virtual, la plantilla crea otros recursos necesarios para el funcionamiento de la misma. En cada recurso existen valores fijados para proteger su seguridad, son los siguientes:

Network Security Group	
Configuración	Descripción
Habilitado para la VM	De esta manera se añade el NSG que por defecto no se realiza
Desactivados los puertos de conexión por defecto	Se crea vacío, sin reglas extras de entrada/salida
Máquina Virtual	
Configuración	Descripción
Última versión del SO	Se indica “latest” por defecto
Disco del SO	Está condicionado dependiendo de si el entorno es de desarrollo, crea SSD Local o entorno productivo, crea SSD replicado

Tabla 10: Configuración seguridad plantilla VM

Acceso a la plantilla:
https://dev.azure.com/sergiolopezlopez/tfg/_git/tfg?path=/plantillas%20ARM/vm.json

b) App Service

Durante la creación de la aplicación web se debe considerar incluir en la plantilla el plan de servicio que la contiene, por lo que, por un lado, se mencionarán las características de la app y por otro, del *serverfarm* (plan).

Para entrar en detalle en la propia plantilla como en el caso anterior, existen varios parámetros de entrada a describir:

Parámetro	Descripción	Condiciones
nombrePlan	Nombre del Service Plan	Se añade el entorno
entorno	Entorno o subscripción	
tamano	tamaño de la máquina virtual	
nombreApp	Nombre del Service Plan	Se añade el entorno
RGPlan	grupo de recursos de la service plan	
RGVNet	grupo de recursos de la red virtual	
NombreVNet	red virtual	Se añade el entorno
localizacion	localización de los recursos	El mismo que el grupo de recursos

Tabla 11: Plantilla parámetros App

Además del App Service, es necesario crear el Service Plan que contendrá la app, en el que no se especifica ningún parámetro interesante a mencionar, pero, sin embargo, sí que se detalla en el caso de la app y de la conexión a la VNet:

App Service	
Configuración	Descripción
Sólo por HTTPS	Redirige todo el tráfico inseguro hacía la app forzando TLS
HTTP 2.0 habilitado	Mayor calidad y velocidad en la comunicación al utilizar la versión 2 de HTTP
Versión TLS mínima 1.2	Evitar protocolos vulnerables al usar versiones inferiores
Sólo conexión por FTPS	Uso del protocolo FTPs, es decir, transferencia de archivos de manera cifrada
Identidad de sistema	Permite utilizar la identidad de la app para otros servicios de Azure (con AAD)
Conexión a VNet	
Configuración	Descripción

Conectado a VNet	Activar internamente la conexión entre la app y la subnet seleccionada dentro de la VNet

Tabla 12: Configuración seguridad plantilla App

Acceso a la plantilla:
https://dev.azure.com/sergiolopezlopez/tfg/_git/tfg?path=/plantillas%20ARM/app.json

c) Cuenta de almacenamiento

Es interesante crear cuentas de almacenamiento con plantillas para evitar configuraciones que por defecto no se habilitan de forma correcta en el portal de Azure. Por ejemplo, la activación de las cuentas públicas.

En la siguiente tabla se pueden analizar los distintos parámetros que se deben introducir para la creación correcta del recurso:

Parámetro	Descripción	Condiciones
nombre	Nombre del recurso	Se añade el entorno
entorno	Entorno o subscripción	
localización	localización de los recursos	El mismo que el grupo de recursos

Tabla 13: Plantilla parámetros Storage

Las configuraciones que caben remarcar en la siguiente plantilla son las siguientes, todas ellas pertenecientes al recurso de la cuenta de almacenamiento.

Cuenta de almacenamiento	
Configuración	Descripción
Versión 2	Mejora la performance y el coste del recurso frente a la versión 1
Versión TLS mínima 1.2	Evitar protocolos vulnerables al usar versiones inferiores
Permitir el acceso público desactivado	Evitar el acceso público analizado en el apartado anterior
Lista de acceso de red restringida a una VNet, todo el tráfico restante denegado	Activar internamente la conexión entre la cuenta de almacenamiento y la subnet seleccionada dentro de la VNet
Sólo por HTTPS	Redirige todo el tráfico inseguro hacía la cuenta forzando TLS

Tabla 14: Configuración seguridad plantilla Storage

Acceso a la plantilla:
https://dev.azure.com/sergiolopezlopez/tfg/_git/tfg?path=/plantillas%20ARM/storage.json

5.2 Automatización de tareas

Para llevar a cabo esta función, Azure dispone de varios servicios de automatizaciones: *Logic Apps* (automatización de tareas en modo gráfico con *workflows*) o Cuentas de automatización (automatización de procesos o *runbooks* mediante Powershell o Python).

En la realización de este proyecto se selecciona la cuenta de automatización por la semejanza y cercanía con los scripts ya utilizados por administradores de sistemas en lenguajes de programación de tipo *scripting*.

Además, para su consulta, se suben todos los códigos de los scripts en el siguiente repositorio de Azure DevOps (hecho público): https://dev.azure.com/sergiolopezlopez/_git/tfg y se añaden en los anexos del proyecto.

a) Actualización de directivas y políticas de seguridad de red en el firewall

Un mecanismo para garantizar una buena gestión del tráfico de la red en Azure es la utilización de un FW como se ha comprobado anteriormente, sin embargo, sin una creación de políticas de red adecuada el FW no utiliza el cien por cien de sus características. Es por ello, que, para el ejemplo de organización, se aplicará la **denegación del tráfico proveniente de fuentes que se han reportado como maliciosas** en el siguiente automatismo "ActualizarPoliticasFireWall" de la imagen 42. Por otro lado, se describe de forma esquemática el flujo del script en la pág. 101 de los anexos.

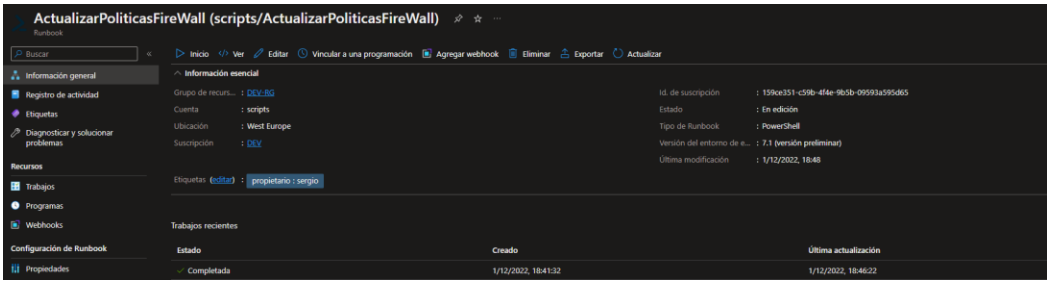


Ilustración 42: Runbook tarea politicasFW

En esta simulación, el administrador de la startup ha encontrado un repositorio público con un listado de IPs maliciosas, que se actualizan de forma periódica, https://pastebin.com/RVArjEKJ. Por lo que, se configura un automatismo que vaya consultando la API del servicio de *pastebin* –es un servicio web que permite a los usuarios subir pequeños textos– y en caso de cambio, aplique las nuevas reglas en el firewall de la compañía "fw-dev".

Una vez obtenido el listado de IPs se crea una función que comprueba las políticas de reglas actual y esta se combina con otra función que realiza un

bucle por cada IP encontrada, añadiendo dicha política a las reglas del FW y bloqueando las principales IPs publicadas.

El resultado del script deja un log en el firewall que indica la creación/actualización de la política originaria por el runbook alojado en la cuenta de automatización “scripts” y añade las reglas mencionadas, ilustración 43:

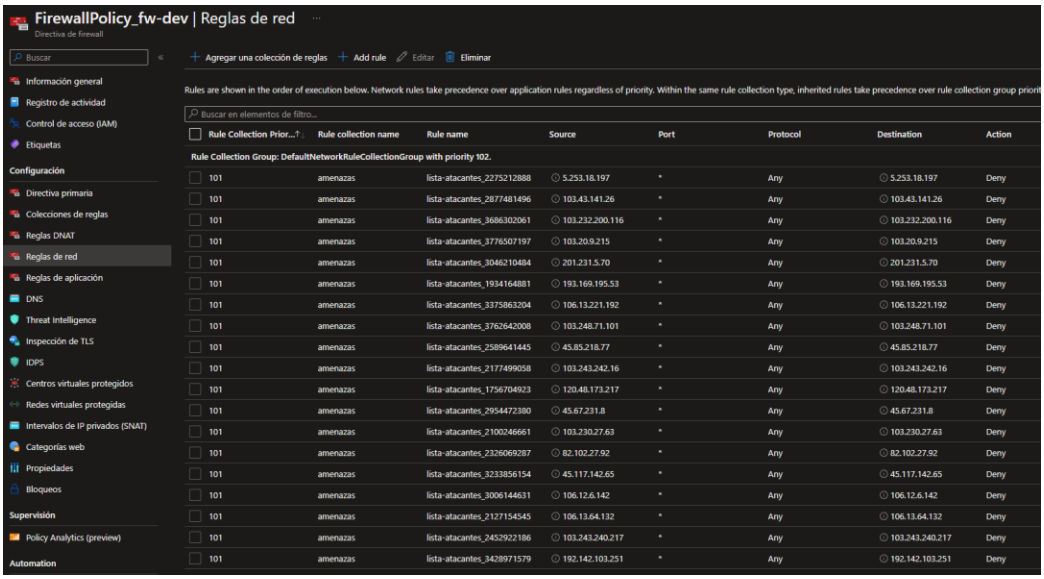


Ilustración 43: Resultado politicasFW

El automatismo tiene la tarea para ejecutarse de forma regular, siguiendo el siguiente calendario semanalmente, ejemplificado en la ilustración 44:

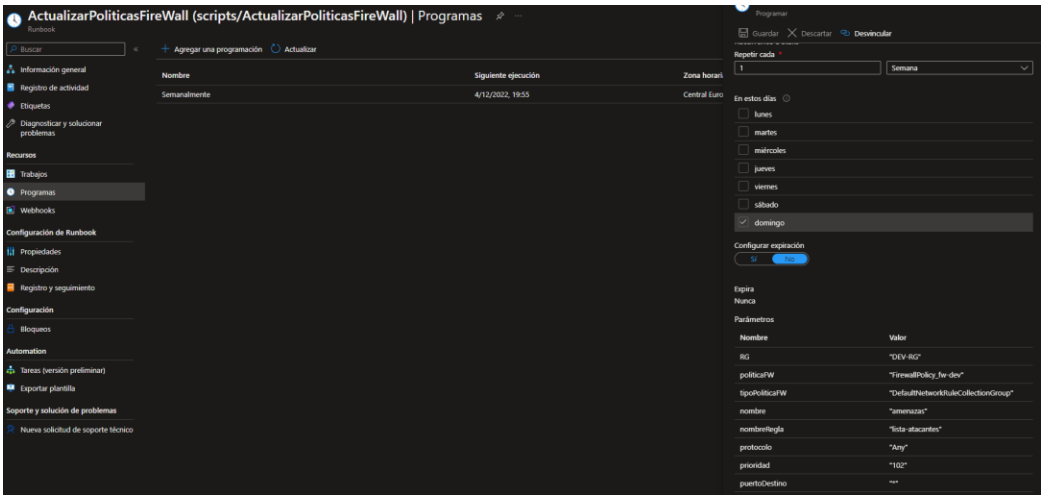


Ilustración 44: Calendario de ejecución politicasFW

Enlace directo al código del script (realizado en Powershell 7): https://dev.azure.com/sergiolopezlopez/_git/tfg?path=%2Frunbooks%2FActualizarPoliticasFireWall.ps1&version=GBmain.

b) Monitor de conexiones de usuarios

Dado la carencia en la monitorización de los *logins* de los usuarios del Azure AD, que a día de hoy no está gestionada de forma nativa con políticas de gestión de usuarios, el administrador considera crear un script para dicho cometido.

En esta primera versión, la automatización realizará una comprobación diaria, analizando todos los accesos sobre el directorio de la organización y alertará en caso de que existan las siguientes condiciones resultantes en la ilustración 46:

- Acceso fines de semana *(marcados con 1)*
- Acceso fuera del horario laboral semanal conocido (pueden existir sub-condiciones para usuarios concretos que deban estar excluidos por causas verificadas, guardias, etc). *(marcados con 2)*
- Acceso desde otras localidades distintas a las validadas por recursos humanos por cada trabajador. *(marcados con 3)*

En siguientes versiones, se actuará de forma bloqueante, expulsando al usuario de la sesión, aunque en fases de pruebas, se recomienda utilizarlo en un modo de auditoría, únicamente.

Para llevarse a cabo esta tarea, se ha creado el siguiente automatismo tipo *runbook*: "ObtenerLoginsFueraHoras" (ilustración 45) que tiene un comportamiento esquematizado en los anexos, pág. 102.

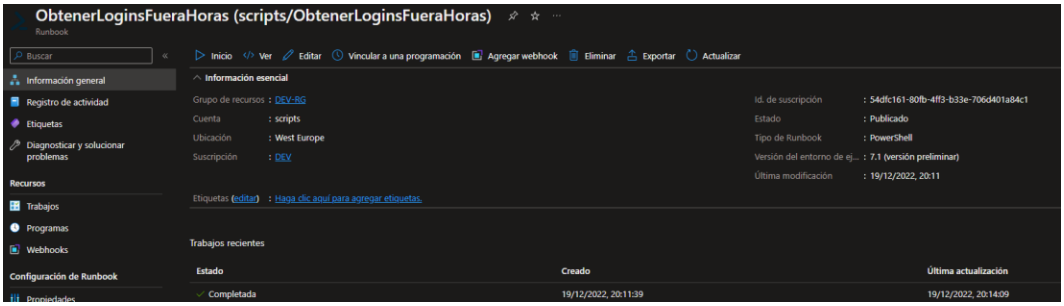


Ilustración 45: Runbook obtención logins

Se ejecutó como : Usuario Instantánea de origen : Ver instantánea de origen

Entrada **Salida** Errores Advertencias Todos los registros Excepción

Nombre	Email	FechaAcceso	Dia	IP	SistemaOperativo	Buscador	Localidad	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:48:59	Sunday	195.181.167.194	Windows 10	Chrome 108.0.0	Madrid - ES	1
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:49:55	Sunday	195.181.167.194	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:49:58	Sunday	195.181.167.194	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:53:22	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:57:10	Sunday	5.8.16.168	Windows 10	Chrome 108.0.0	Sankt-Peterburg - RU	3
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:57:25	Sunday	5.8.16.168	Windows 10	Chrome 108.0.0	Sankt-Peterburg - RU	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:57:50	Sunday	5.8.16.168	Windows 10	Chrome 108.0.0	Sankt-Peterburg - RU	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:58:22	Sunday	156.146.50.1	Windows 10	Chrome 108.0.0	Kyiv Misto - UA	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 12:58:56	Sunday	195.181.167.198	Windows 10	Chrome 108.0.0	Madrid - ES	1
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 13:00:03	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 13:00:16	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 13:01:19	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 13:01:21	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 13:01:23	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 16:57:08	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 16:57:11	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	18/12/2022 16:57:13	Sunday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	19/12/2022 0:09:55	Monday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	2
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	19/12/2022 0:09:58	Monday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	
usuario1	usuario1@sergiolololocloudhotmail.onmicrosoft.com	19/12/2022 0:09:59	Monday	86.127.229.77	Windows 10	Chrome 108.0.0	Madrid - ES	

Ilustración 46: Resultado del runbook de obtención de logins

Nota a considerar: es requisito disponer de una licencia de Azure Active Directory Premium (para la investigación, se selecciona el periodo gratuito de 30 días P2).

Enlace directo al código del script (realizado en Powershell 7): https://dev.azure.com/sergiolopezlopez/tfg/_git/tfg?path=/runbooks/ObtenerLoginsFueraHoras.ps1

c) Control de recursos inseguros: Cuentas de almacenamiento públicas y App Services con TLS desprotegidos

El siguiente automatismo regula dos de las inseguridades que se argumentan apartados b y c del punto 5.1, cuentas de almacenamiento con configuración por defecto, habilitando el acceso anónimo público y aplicaciones web con algún parámetro mal configurado, relacionado con el protocolo TLS (uso de mínima versión 1.2, no uso de HTTP versión 2, FTP sin cifrado y no forzado el acceso por HTTPS).

Para ello, se muestra el automatismo tipo *runbook* capturado en la ilustración 47 “recursosInseguros” y esquematizado en los anexos, pág. 103.

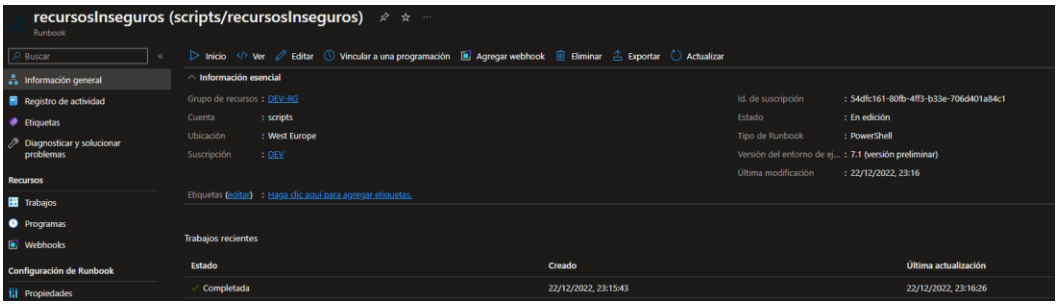


Ilustración 47: Runbook recursos inseguros

La tarea de este automatismo es la siguiente: como primer paso, lanza una consulta al *Azure Graph* –servicio de datos de administración de recursos– filtrando, por un lado, cuentas de almacenamiento con la propiedad de

“permiso de acceso público” con la *flag* en verdadero y, por otro lado, simplemente un listado general de todas las *app services* de la organización.

Seguidamente, se obtiene el primer informe formateado mediante una búsqueda en bucle de cada uno de las cuentas de almacenamiento y el segundo informe formateado en un segundo bucle por cada app que cumpla las condiciones de inseguridad del protocolo TLS.

Finalmente, ambos reportes se ordenan por nombre de cada servicio y se muestran por pantalla, ilustración 48.

```
Entrada   Salida   Errores   Advertencias   Todos los registros   Excepción

Revisando: stoaplicaciontfg01 (dev-rg)

Subscripcion ResourceGroup NombreStorageAccount BlobAccesoPublico NombreContainer ContainerAccesoPublico
------------ ------------- -------------------- ----------------- --------------- ----------------------
DEV          dev-rg        stoaplicaciontfg01   [!] True          contenedor                   Container
DEV          dev-rg        stoaplicaciontfg01   [!] True          blob                              Blob


Revisando: APP-INTRANET-TFG-DEV (dev-rg)
Revisando: appmintls01 (dev-rg)
Revisando: appmintls03 (dev-rg)
Revisando: apponlyhttps01 (dev-rg)

Subscripcion ResourceGroup NombreAppService MinTlsVersion Http20Enabled FtpsState HttpsOnly
------------ ------------- --------------- ------------- ------------- --------- ---------
DEV          dev-rg        appmintls01     [!] 1.0       [!] False     FtpsOnly       True
DEV          dev-rg        appmintls03     [!] 1.1       [!] False     FtpsOnly       True
DEV          dev-rg        apponlyhttps01  1.2           [!] False     FtpsOnly  [!] False
```

Ilustración 48: Resultado de recursos inseguros

Enlace directo al código del script (realizado en Powershell 7): https://dev.azure.com/sergiolopezlopez/tfg/_git/tfg?path=/runbooks/recursosInseguros.ps1

d) Exportar reporte de seguridad de Microsoft Defender

Microsoft Defender for cloud es el recurso de administración de seguridad en la nube de Azure (CSPM), entre otras cosas, tal y como se ha mencionado en el apartado 3.3.1.

Por consiguiente, es el mejor sistema de consulta y valoración de la protección de todas las subscripciones que el administrador deberá consultar para encontrar los puntos débiles de la startup que se generen durante todo el ciclo de vida del negocio.

No obstante, el informe que genera es consultable de forma muy manual con el portal web, e impide el correcto seguimiento de todos los eventos interesantes a tratar, a pesar de la posibilidad básica de exportar de forma limitada el resultado por una hoja de cálculo en formato *csv* demasiado desordenado y sin elaborar. Es por ello, se valora la creación el siguiente automatismo “ReportMicrosoftDefender”, ilustración 49 y el esquema del flujo del programa en la pág. 104 de los anexos. Además, de esta manera se mejora el ciclo de los informes, permitiendo añadir avisos recurrentes y eventuales.

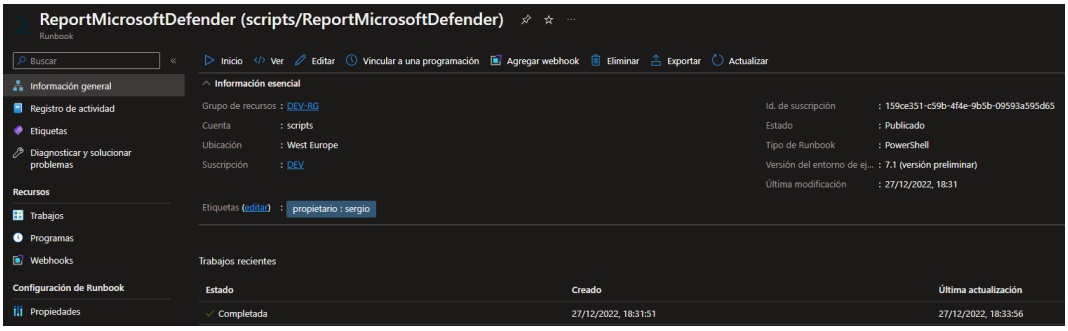


Ilustración 49: Runbook Report Microsoft Defender for cloud

El funcionamiento es el siguiente, el script se divide en dos tareas fundamentales, por un lado, consultar todas las recomendaciones de seguridad para cada subscripción, simplemente llamando a la API de Azure encargada para dicha función, mostrando el resultado en una tabla formateada por pantalla (ilustración 50) y por otro lado, la generación y actualización del reporte en Excel, ilustración 51, en formato *xlsx*, con dos gráficas que amplían la visualización de los recursos afectados resultantes (solo disponible en la herramienta portable y no en el automatismo, debido a la necesidad de la subida del resultado del Excel a una maquina o cuenta con el programa instalado, no necesario para la automatización).

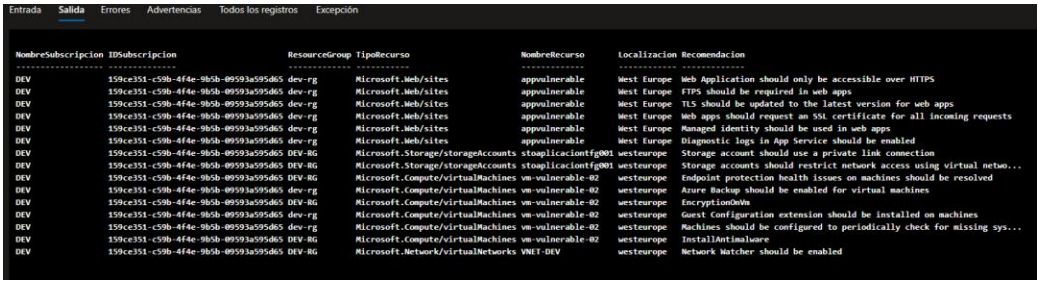


Ilustración 50: Resultado formatado Report Microsoft Defender

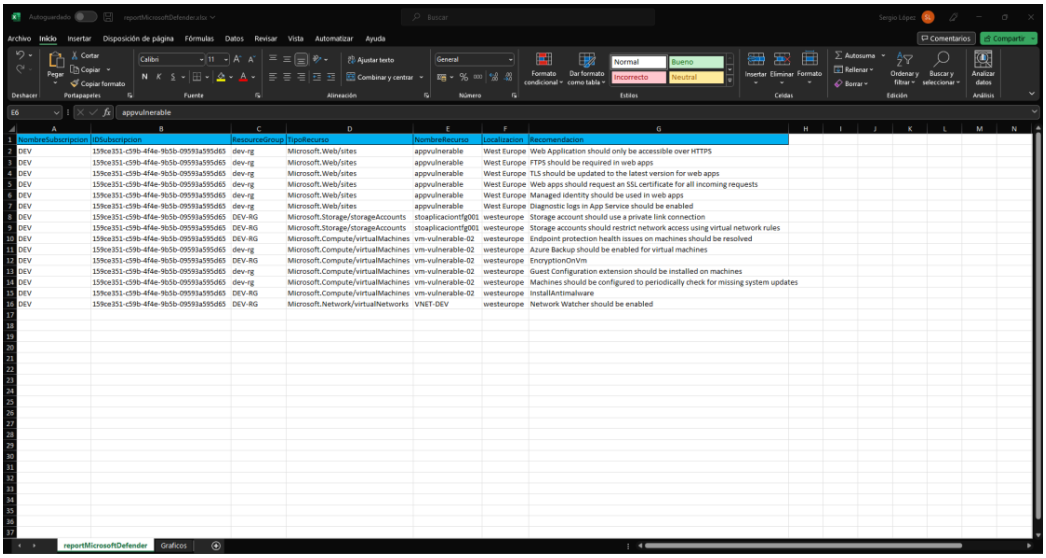


Ilustración 51: Resultado en Excel Report Microsoft Defender

Pudiendo organizar en una siguiente pestaña (ilustración 52) en cada ejecución una actualización en tablas dinámicas, que a su vez muestran dos gráficas tipo circular y por barras. En el primer caso, se agrupan la cantidad de recursos afectados por tipo de recursos (máquinas virtuales, cuentas de almacenamiento, etc) y en el segundo caso, cuantas recomendaciones se dan distintas, de tal manera que el administrador deberá centrar los recursos para solventar las amenazas sin duplicar la tarea posteriormente.

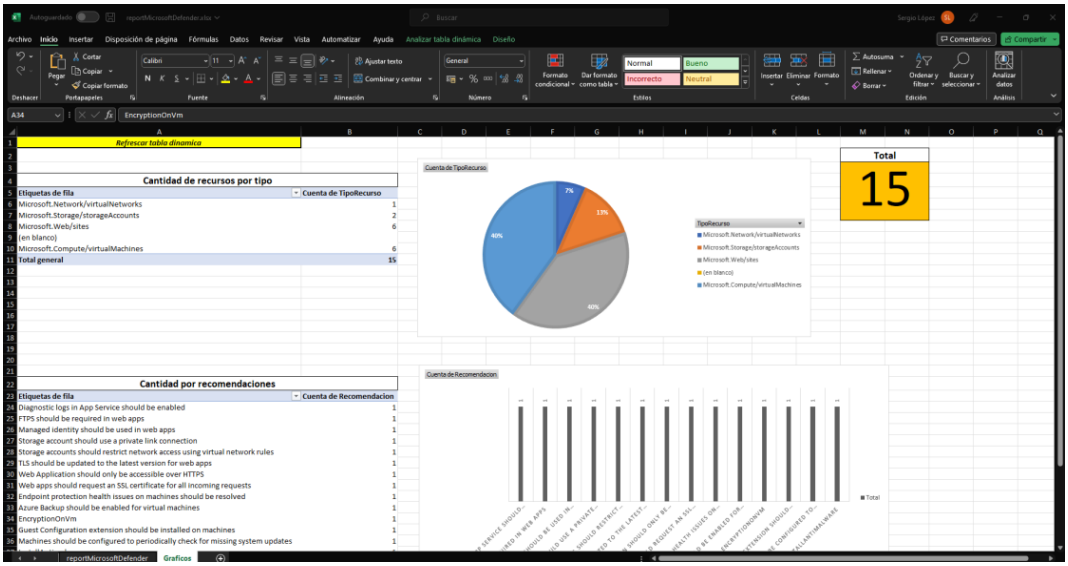


Ilustración 52: Resultado con gráficas de Report Microsoft Defender

Enlace directo al código del script (realizado en Powershell 7): https://dev.azure.com/sergiolopezlopez/tfg/_git/tfg?path=/runbooks/ReportMicrosoftDefender.ps1

5.3. Herramienta portable

Además de los automatismos creados en los apartados anteriores, estos se integran, componiendo una herramienta portable, con la finalidad de que el administrador le aporte un set a ejecutar con los scripts ya creados, pero bajo demanda.

De igual manera, cuenta con una opción para crear los recursos bajo plantillas, tratados en el apartado 5.1, de manera cómoda, siguiendo los parámetros de seguridad ya mencionados, es decir, pudiendo crear una máquina virtual, una app service o una cuenta de almacenamiento al vuelo desde la propia herramienta.

La herramienta, **Az Infra Sec** (combina las siguientes palabras, Azure, Infraestructura y Seguridad) está alojada en el mismo repositorio público de Azure DevOps, pudiéndose descargar de la siguiente manera:

```
git clone
https://sergiolopezlopez@dev.azure.com/sergiolopezlopez/tfg/_git/tfg
```

Y, una vez descargado, el siguiente paso, será abrir una terminal de *Powershell (v5.1)* y, seguidamente, ubicarse en la ruta adecuada y ejecutar el script:

```
Set-location '.\tfg\Herramienta AzInfraSec\'
.\herramienta-seguridad.ps1
```

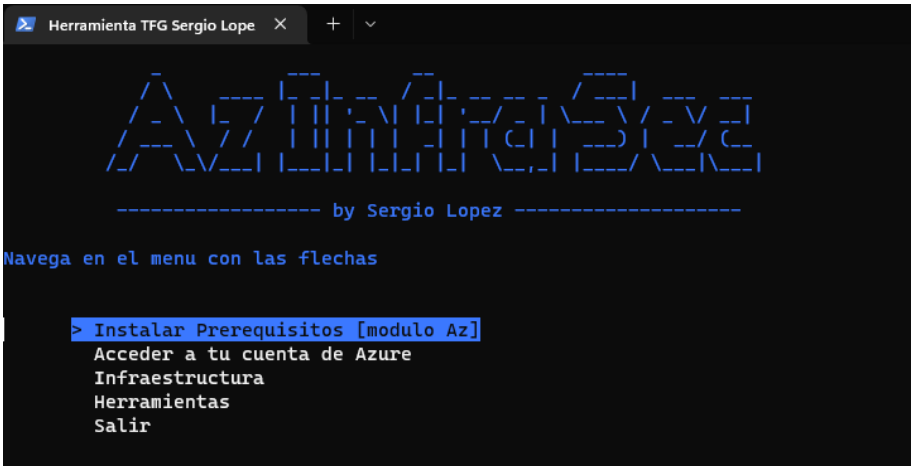


Ilustración 53: Menú Az Infra Sec

Las opciones de menú, ilustración 49, son las siguientes:

a) Instalar prerrequisitos [módulo Az]
Instala los módulos de Azure *powershell* necesarios para la ejecución de la herramienta. Más información en los anexos.

b) Acceder a la cuenta de Azure
Realiza un login a la cuenta de Azure del usuario, de esta manera se podrá seguir usando las siguientes opciones del menú.

c) Infraestructura

Hace uso de las plantillas ARM creadas en el apartado 5.1, desplegando según los criterios del usuario.

- Máquina Virtual
- App Services
- Cuenta de almacenamiento

Más información en los anexos.

d) Herramientas

Ejecuta los scripts diseñados anteriormente en el apartado 5.2 para la cuenta de automatismo de Azure, sin embargo, estos se lanzan directamente en la máquina del usuario, otorgándole mayor autonomía.

- Políticas de Azure Firewall
- Auditoria de accesos
- Recursos inseguros: Cuentas de almacenamiento públicas o *AppServices* con TLS desprotegido
- Exportar informe de Microsoft Defender

Más información en los anexos.

e) Salir

Cierra la herramienta correctamente.

6. Conclusiones

A lo largo de esta investigación se han mencionado y descrito algunos conceptos clave referentes al estado actual de la nube pública, cuáles son las amenazas y los riesgos más recurrentes que afectan a este tipo de infraestructura y qué servicios o recursos provee uno de los principales proveedores, Microsoft con su solución *cloud*, Azure.

Asimismo, el siguiente objetivo de la investigación ha sido la implementación de una infraestructura a modo ejemplo que cumpliese los estándares de seguridad estudiados en los primeros apartados y del mismo, comprobar de forma paralela alguno de los ataques y afectaciones vinculadas a malas configuraciones.

Como se ha marcado en el objetivo principal, la idea resultante ha sido realizar un mecanismo automatizado relacionado con la securización de la infraestructura desplegada, que, a diferencia de lo mencionado anteriormente, se realiza de forma autónoma con la mínima interacción humana posible. De tal manera, lo que se ha intentado conseguir es la explotación de los beneficios de automatizar.

Cabe recordar, la intención inicial del proyecto, motivado en potenciar la calidad de la seguridad de los entornos en Azure, dado el auge en el uso de tecnologías *cloud* en la mayoría de las organizaciones, que si bien, comparando las principales amenazas existentes suceden de forma similar en entornos tradicionales *on-premise*. Pero, sin embargo, existía la hipótesis de que factores como las configuraciones o la responsabilidad compartida podían distinguirse entre ambos entornos.

Por ello, ha sido crucial analizar cada escenario enfocado al *cloud,* con la finalidad de poder conocer que opciones permite el proveedor Azure a sus clientes para hacer frente dichas amenazas y que otras capas son responsabilidad del propio proveedor. Igualmente, según el reporte realizado por una de las entidades más influyentes en este sector, ya estudiados en el apartado 2 sobre los riesgos, cada vez la línea que separa las amenazas en entornos tradicionales con entornos *cloud* es menor, pero con la diferencia de que este último cuenta con grandes equipos de seguridad que evitan que al cliente la tarea de mantenimiento de su propio negocio, propiciando entornos con configuraciones por defecto más seguras, que durante el trabajo ha sido un factor clave a tratar.

A pesar de esta abstracción en las configuraciones que Microsoft brinda con sus soluciones, existen muchas opciones que por la tipología del negocio deban ser ajustadas para su correcta funcionalidad y por ello, pierdan el grado de seguridad teórico que disponían. Por lo tanto, es de vital importancia que la organización conozca y pueda elegir que, si necesita utilizar una cuenta de almacenamiento de acceso público sea debido a su requisito y pueda aislarlo de otros recursos, es decir, en todo caso sea consciente del riesgo que supone y pueda gestionarlo debidamente.

Durante la creación de tres recursos ejemplos con plantillas ARM se destaca el gran potencial que proporciona Azure frente a la creación con el portal web, permitiendo configurar parámetros que en ocasiones no estaban disponibles (fuera de versiones no

preliminares), utilizando estructuras tipo JSON o las no mencionadas plantillas BICEP, dado el éxito de la infraestructura como código, pero con una sintaxis más simple.

El uso de scripts en el lenguaje Powershell, cercano al sistema operativo, han permitido gracias a la buena documentación de Microsoft, usar una serie de módulos para gestionar los recursos de Azure de forma completa, permitiendo a su vez, la estandarización de las acciones tratadas a fin de securizar la infraestructura utilizada de ejemplo en el proyecto, automatizar el bloqueo de IPs en el cortafuegos, controlar los accesos de los empleados, auditar vulnerabilidades vistas en dos tipos de recursos comunes y hacer uso del potencial de CSPM, *Microsoft Defender for Cloud* para generar informes de las recomendaciones de seguridad que Azure que por defecto encuentra y no debe pasarse de largo, automatizando su ejecución de forma recurrente.

Y concluir con un kit de herramientas, denominada íntegramente como "AzInfraSec", potencialmente útil para los responsables de infraestructura y seguridad para utilizar el listado de *scripts* pero, bajo demanda, que, si bien durante la investigación se han encontrado alternativas externas muy completas, Azure debería proporcionar su propia herramienta complementando sus características de seguridad.

Sin embargo, gracias a este tipo de trabajos de investigación, estas herramientas ya existentes –como la creada para este proyecto– proporcionan un plus para todas aquellas organizaciones que durante su transformación digital al paradigma *cloud* necesiten crear y verificar la correcta configuración de sus recursos de forma constante, pero como se ha analizado, dada la variedad de amenazas confirmen la hipótesis, aplicando la herramienta resultante de este proyecto, configurar su infraestructura correctamente.

Tras la realización de la investigación, debido al gran abanico de servicios que se deben proteger y de su constante actualización y versiones, aparecen limitaciones que se pueden afrontar en futuros trabajos, que seguidamente se introducen.

Durante el estudio de las cuentas de almacenamiento, se puede profundizar en otras configuraciones, como son las claves SAS o la vinculación con el dominio de *Kerberos* de *Active Directory* para trabajar con los ficheros almacenados de forma segura.

Otros de los recursos muy utilizados y los cuales se pueden estudiar a fondo son los servidores de bases de datos, que debida a su gran extensión ha quedado fuera del objetivo de la investigación, pero es sumamente importante debido a su capacidad de explotación.
Además, no se han tratado métodos de protección más reactivos, es decir, post-amenaza, como son el uso de copias de seguridad y restauración con *Recovery Service Vaults* o el uso de grupos de *failover* para casos de desastres, con planes de *Disaster Recovery*.

Este trabajo no se ha centrado en explotar las características inherentes que Azure provee para la seguridad, como son las numerosas opciones de *Microsoft Defender for Cloud* o *Sentinel*, las cuales su estudio permite enfocar este proyecto por otra rama,

más auditora, sin embargo, la idea principal ha sido encontrar los agujeros de seguridad a tratar de manera más tradicional utilizando metodologías de scripting conociendo internamente la causa y el motivo a tratar.

7. Glosario

Término	Definición
.NET	Plataforma para desarrolladores de código abierto, creada por Microsoft, para construir aplicaciones.
ACL	Lista de control de acceso (ACL) es un conjunto ordenado de reglas para filtrar el tráfico.
AD	Siglas de Active Directory, servicio de directorio desarrollado por Microsoft para redes de dominio de Windows.
AES	Siglas de *Advanced Encryption Standard*, es un estándar de encriptación avanzado para cifrar las comunicaciones de datos.
Agile	Metodología de trabajo centrada en la implementación rápida, eficiente y flexible de planear el flujo de trabajo por iteraciones.
API	Interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones.
APT	Amenaza persistente avanzada. Tipo de ciberataque a gran escala, con el objetivo de robar datos a empresas u organismos públicos o efectuar el espionaje de sus sistemas.
Azure	Plataforma informática en la nube operada por Microsoft para la gestión de aplicaciones por centros de datos distribuidos.
Backdoor	Puerta trasera que se utiliza para dar acceso a un equipo infectado de manera remota.
Backend	Servidor que provee, procesa y almacena la data que solicita un cliente.
Backlog	Listado del trabajo pendiente, ordenado por prioridad.
Ciphers	Conjunto de algoritmos que ayudan a asegurar una conexión de red.
Click&run	Término usado para referir un programa o acción que se ejecuta sin previa configuración.
Cloud	Uso de una red de servidores remotos conectados a internet para almacenar, administrar y procesar datos, servidores, bases de datos, redes y software.
Cores	Núcleo de un procesador.
Covid19	Enfermedad infecciosa provocada por el virus SARS-CoV-2.
CSPM	Siglas de *Cloud Security Posture Management*, conjunto de herramientas y prácticas de seguridad destinadas a identificar y corregir problemas de configuraciones incorrectas.
CWPP	Siglas de *Cloud Workload Protection Platform*, herramienta de seguridad que detecta y elimina las amenazas en el software.
DAFO	Herramienta de estudio que analiza características internas y situación externa en una matriz cuadrada.
Datacenters	Lugar donde se concentran los recursos necesarios para el procesamiento de la información.
Dataleaks	También conocido como filtrado o escape de datos.
DDos	Siglas de ataque de denegación de servicio distribuido. Ataque en masa a un sistema causando la caída de un servicio o recurso.
DevOps	Conjunto de prácticas que agrupan el desarrollo de software y las operaciones de TI.
DevSecOps	Integración de la seguridad en la metodología *DevOps*, desde principio a fin del desarrollo de software.
ETL	Siglas de *Extract*, *Transform* y *Load*, proceso que permite mover datos de múltiples fuentes, procesarlas y almacenarlas.
Exploits	Fragmentos de código o programas especializados que aprovechan una vulnerabilidad de software o un defecto de seguridad.
FaaS	Siglas de Función como servicio.
FQDN	Nombre de dominio completo.

Frameworks	Estructura base utilizada como punto de partida para elaborar un proyecto con objetivos específicos.
Frontend	Servidor tipo web con la que el usuario interactúa.
HSM	Dispositivo hardware resistente a manipulaciones que aseguran los procesos criptográficos generando, protegiendo y administrando claves utilizadas para cifrar y descifrar datos.
IaaS	Siglas de Infraestructura como servicio.
IaC	Siglas de Infraestructura como código.
IAM	Siglas de *Identity and Access Management*. Servicio de control de identidades de recursos.
ICMP	Siglas de *Internet Control Message Protocol*. Protocolo de diagnosis de fallos en comunicación en red.
IDaaS	Siglas de Identidad como servicio.
Insights	Muestra información interna del uso de la aplicación (APM).
Kanban	Metodología centrada en la mejora continua, donde las tareas se extraen de una lista de acciones pendientes en un flujo de trabajo constante.
KB	Siglas de Microsoft Knowledge Base. Tecnología para almacenar información utilizada por un sistema.
Kerberos	Protocolo de autenticación que permite a los sistemas y usuarios probar su identidad por un tercero de confianza.
Kubernetes	Plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios.
Legacy	Sistema, tecnología o aplicación de software antiguo o desactualizado que sigue en uso
Log4Shell	Vulnerabilidad catalogada como crítica que afecta a la librería de Java Log4J v2 que permite la ejecución de código remoto.
LOPD	Siglas de Ley Orgánica de Protección de Datos.
MFA	Siglas de *Multiple Factor Authenticator*.
OAuth	*Open Authorization*, protocolo de otorgación de autorizaciones a los usuarios.
On-premise	Tipo de instalación realizada de manera local.
PaaS	Siglas de Plataforma como servicio.
Peering	Intercambio de datos entre redes de Internet permitiendo la conexión entre dos o más redes para transferir tráfico entre ambas.
PING	Envío de paquetes ICMP para el diagnóstico de la red.
Pivotar	En el argot de la ciberseguridad, es el proceso de salto entre una maquina a otra, dentro de una red.
Powershell	Solución de automatización de tareas multiplataforma formada por un *shell* de línea de comandos, un lenguaje de *scripting* y un marco de administración de configuración.
Proxy	Servidor que hace de intermediario en las peticiones de recursos que realiza un cliente a otro servidor.
Python	Lenguaje de alto nivel de programación interpretado multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional.
Ramsomware	Tipo de *malware* que impide a los usuarios acceder a su sistema o a sus archivos personales exigiendo el pago de un rescate por ello.
RBAC	Siglas de *Role based access control*. Modelo de seguridad que permite asignar funciones y autorizaciones basado en roles a recursos informáticos.
Runtime	Intervalo de tiempo en el que un programa de computadora se ejecuta en un sistema operativo.
SaaS	Siglas de Software como servicio.

Serverfarm	Grupo de servidores para ejecutar tareas más complejas que con un solo servidor.
SIEM	Siglas de *Security Information and Event Management*. Sistema de Gestión de Eventos e Información de Seguridad es un sistema que centraliza el almacenamiento y la interpretación de los datos relevantes de seguridad.
SLA	Siglas de *Service Level Agreement*. Contrato que describe el nivel de servicio que un cliente espera de su proveedor.
SMB	Siglas de *Server Message Block*. Protocolo cliente - servidor que gobierna el acceso a archivos y directorios completos, así como a otros recursos de red.
Sniffing	Técnica utilizada para escuchar todos los paquetes que circulan dentro de una red.
SOAR	Siglas de *Security Orchestration Automation and Response*. Plataforma de operaciones y generación de informes de seguridad que utiliza datos extraídos de distintas fuentes para proporcionar capacidades de gestión, análisis y generación de informes en apoyo a los equipos analistas en un SOC.
SOC	Siglas de *Security Operations Center*. Plataforma que permite la supervisión y administración de la seguridad del sistema de información por herramientas de recogida, correlación de eventos e intervención remota.
Sprint	Período breve de tiempo fijo en el que un equipo *Agile* trabaja para completar una cantidad de trabajo establecida.
SQL	Siglas de *Structured Query Language*. Lenguaje gestor para el manejo de la información en las bases de datos relacionales.
SSL	Siglas de *Secure Sockets Layer*. Protocolo que permite la comunicación cifrada entre un sitio web y un navegador web, securizando la transmisión de información confidencial.
SSO	Siglas de *Single Sign-On*. Procedimiento de autenticación que habilita a un usuario determinado para acceder a varios sistemas con una sola instancia de identificación.
Startup	Empresa de nueva creación que comercializa productos y/o servicios a través del uso intensivo de las tecnologías de la información y la comunicación (TIC's), con un modelo de negocio escalable el cual le permite un crecimiento rápido y sostenido en el tiempo.
Subscripción	En el argot de Azure, es una unidad lógica de servicios de Azure que se vincula a una cuenta.
Tenant	En el argot de Azure, es una instancia dedicada y confiable de Azure AD que se crea automáticamente cuando una organización se suscribe a un servicio en la nube de Azure.
TLS	Siglas de *Transport Layer Security*. Versión actualizada y mejorada de SSL.
VPN	Siglas de *Virtual Private Network*. Tecnología de red de ordenadores que permite una extensión segura de la red de área local sobre una red pública o no controlada como Internet.
WAF	Siglas de *Web Application Firewall*. Tipo de cortafuegos que protege de múltiples ataques al servidor de aplicaciones web en el *backend*.
WAN	Siglas de *Wide Area Network*. Gran red que conecta grupos de dispositivos a grandes distancias.

8. Referencias bibliográficas

[1] Flexera. (2022) “State of the Cloud Report”. [Web]: https://info.flexera.com/CM-REPORT-State-of-the-Cloud [Último acceso: Noviembre 2022].

[2] C. Roca. (Octubre 2021) ThePower, “Los 7 Proveedores de Cloud Computing TOP del mercado”. [Web]: https://www.thepowermba.com/es/blog/proveedores-cloud-computing [Último acceso: Octubre 2022].

[3] L. Sujay V. (Agosto 2022). Statista, “Cloud infrastructure service s vendor market share worldwide from 4th quarter 2017 to 1st quarter 2022”. [Web]: https://www.statista.com/statistics/967365/worldwide-cloud-infrastructure-services-market-share-vendor/#:~:text=In%20the%20first%20quarter%20of,with%20eight%20percent%20market%20share [Último acceso: Octubre 2022].

[4] Microsoft (Septiembre 2022). “¿Qué es Azure Blueprint?”. [Web]: https://learn.microsoft.com/es-es/azure/governance/blueprints/overview [Último acceso: Octubre 2022].

[5] Microsoft (Septiembre 2022). “¿Qué es Azure Rights Management?”. [Web]: https://learn.microsoft.com/es-es/azure/information-protection/what-is-azure-rms [Último acceso: Octubre 2022].

[6] Microsoft (Junio 2020). “Cloud Security Posture Management (CSPM) with Azure Security Center”. [Web]: https://learn.microsoft.com/en-us/shows/azure-friday/cloud-security-posture-management-cspm-with-azure-security-center [Último acceso: Octubre 2022].

[7] Microsoft (Septiembre 2022). “Funcionalidades técnicas de seguridad de Azure”. [Web]: https://learn.microsoft.com/es-es/azure/security/fundamentals/technical-capabilities [Último acceso: Octubre 2022].

[8] UNDP (2015). “What are the Sustainable Development Goals?”. [Web]: https://www.undp.org/sustainable-development-goals [Último acceso: Octubre 2022].

[9] UOC (Septiembre 2022). “¿Cómo incorporar la competencia "Comportamiento ético y global" al Trabajo Final (TF)?”. [Web]: https://drive.google.com/file/d/1sJWP6UzG8a5gPbV0-NLSNAj0JE5NjVBJ/view [Último acceso: Octubre 2022].

[10] T. Mufti, P. Mittal y B. Gupta (Febrero 2020). EUDL. “A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services”.

[Web]: https://eudl.eu/pdf/10.4108/eai.27-2-2020.2303255 [Último acceso: Octubre 2022].

[11] A. Yevge, P. Ghag, C. Solanki y A. Mishra (Febrero 2022). EasyChair Preprint. "Review Paper on Cloud Service Provider – AWS, Azure, GCP". [Web]: https://mail.easychair.org/publications/preprint_download/HsX3 [Último acceso: Octubre 2022].

[12] Karl Ots (2021). Springer. "Azure Security Handbook". [Web]: https://link.springer.com/content/pdf/10.1007/978-1-4842-7292-3.pdf [Último acceso: Octubre 2022].

[13] P. Tender, D. Rendon y S. Erskine (2019). "Pro Azure Governance and Security". [Web]: https://link.springer.com/content/pdf/10.1007/978-1-4842-4910-9.pdf [Último acceso: Octubre 2022].

[14] Md K, Beesetty Y, Pramod B, Vineet K (Abril 2022). "Infrastructure As A Service (IaaS) Market Research, 2030". [Web]: https://www.alliedmarketresearch.com/infrastructure-as-a-service-iaas-market [Último acceso: Octubre 2022].

[15] Azure. "High-level modeling". [Web]: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/strategy/monitoring-strategy#high-level-modeling [Último acceso: Octubre 2022].

[16] CloudFlare. "What is the cloud". [Web]: https://www.cloudflare.com/es-es/learning/cloud/what-is-the-cloud/ [Último acceso: Octubre 2022].

[17] CloudFlare. "What is cloud migration". [Web]: https://www.cloudflare.com/es-es/learning/cloud/what-is-cloud-migration/ [Último acceso: Octubre 2022].

[18] Azure (Enero 2022). "SLA summary for Azure services". [Web]: https://azure.microsoft.com/en-us/support/legal/sla/summary/ [Último acceso: Octubre 2022].

[19] Eu Commission (Enero 2021). "European Alliance on Industrial Data, Edge and Cloud". [Web]: https://digital-strategy.ec.europa.eu/en/policies/cloud-computing [Último acceso: Octubre 2022].

[20] CSA (Junio 2022). "Top Threats to Cloud Computing Pandemic Eleven". [Web]: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-pandemic-eleven/ [Último acceso: Octubre 2022].

[21] S. Menashe, O. Peles, O. Hollander (Diciembre 2021). JFrog "Log4j Log4Shell 0-Day Vulnerability: All You Need To Know". [Web]: https://jfrog.com/blog/log4shell-0-day-vulnerability-all-you-need-to-know/ [Último acceso: Octubre 2022].

[22] Alexander Culafi (Octubre 2022). TechTarget. “Microsoft confirms data leak caused by misconfiguration”. [Web]: https://www.techtarget.com/searchsecurity/news/252526344/Microsoft-confirms-data-leak-caused-by-misconfiguration [Último acceso: Octubre 2022].

[23] AEPD (Mayo 2020). “Protección de Datos: Guía para el Ciudadano”. [Web]: https://www.aepd.es/sites/default/files/2020-05/guia-ciudadano.pdf [Último acceso: Octubre 2022].

[24] Microsoft (Setiembre 2022). “Funciones de seguridad en la nube”. [Web]: https://learn.microsoft.com/es-es/azure/cloud-adoption-framework/organize/cloud-security [Último acceso: Noviembre 2022].

[25] I. Arce (Abril 2003). IEEE Security & Privacy . “The weakest link revisited [information security]”. [Último acceso: Noviembre 2022].

[26] Microsoft (Octubre 2022). “End-to-end security in Azure”. [Web]: https://learn.microsoft.com/en-us/azure/security/fundamentals/end-to-end [Último acceso: Noviembre 2022].

[27] Michael Vizard (Octubre 2022). SecurityBoulevard: “DDoS Attacks Exceeded Six Million in First Half of 2022”. [Web]: https://securityboulevard.com/2022/10/ddos-attacks-exceeded-six-million-in-first-half-of-2022/ [Último acceso: Noviembre 2022].

[28] Radware (Marzo 2021). Radware Blog: “How WAFs Can Mitigate The OWASP Top 10”. [Web]: https://blog.radware.com/security/applicationsecurity/2021/03/how-wafs-can-mitigate-the-owasp-top-10/ [Último acceso: Noviembre 2022].

[29] CincoDías (Octubre 2022). El País: “Situación actual de las startups en España”. [Web]: https://cincodias.elpais.com/cincodias/2022/10/19/emprendedores/1666175939_015416.html#:~:text=Las%20startups%20tienen%20una%20supervivencia%20superior&text=Entre%20el%20a%C3%B1o%202016%20y,4%2C1%25%20son%20startups. [Último acceso: Noviembre 2022].

[30] Microsoft (Octubre 2021). “Identity decision guide”. [Web]: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/decision-guides/identity/ [Último acceso: Noviembre 2022].

[31] SOCRadar (Octubre 2022). “Sensitive Data of 65,000+ Entities in 111 Countries Leaked due to a Single Misconfigured Data Bucket”. [Web]: https://socradar.io/sensitive-data-of-65000-entities-in-111-countries-leaked-due-to-a-single-misconfigured-data-bucket/ [Último acceso: Diciembre 2022].

9. Anexos

9.1. Vulnerabilidad en los privilegios de lectura del Azure Active Directory

Durante la realización de este proyecto de trabajo de final de grado, se ha encontrado una vulnerabilidad –dado que objetivamente, existen unos privilegios de lectura innecesarios– en la cuenta de estudiantes, directorio: Universitat Oberta de Catalunya (uoc0.onmicrosoft.com).

A pesar de que un estudiante no tiene acceso al Azure Active Directory, ni por lo tanto, a la sección de usuarios del directorio de la organización uoc0 como se comprueba en la ilustración 54.

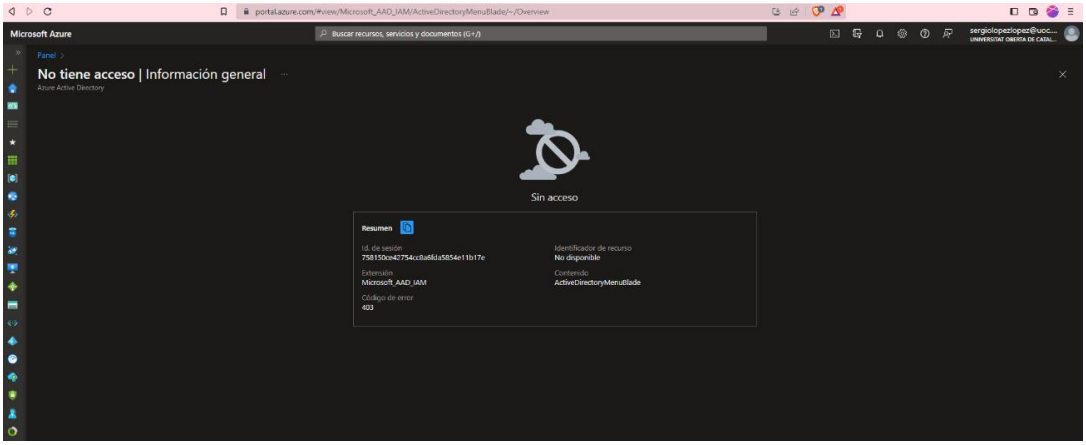


Ilustración 54: Comprobación AzureAD

Durante la asignación de roles en el control de acceso de un recurso, con el panel lateral IAM, muestra una lista acotada (de los 50 primeros miembros) de los miembros del AAD, mostrado en la imagen 55, y existiendo la opción de aplicar filtros de texto para exponer búsquedas más precisas.

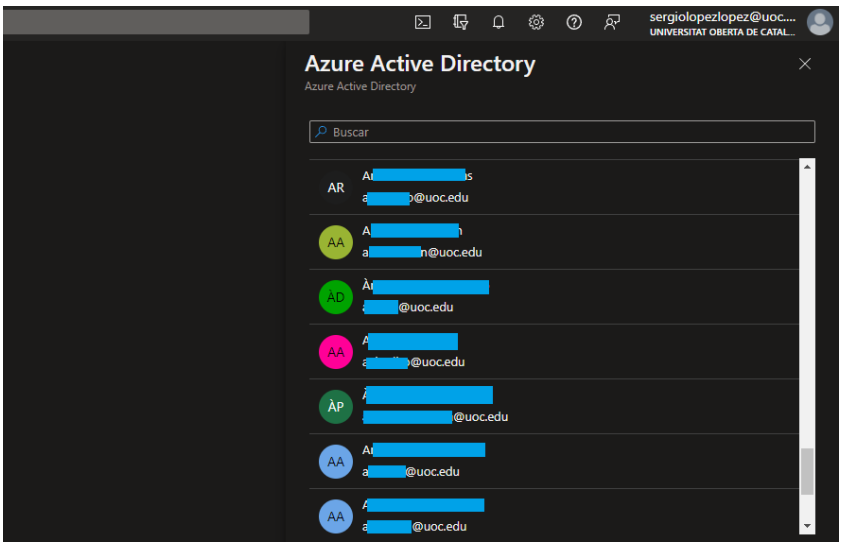


Ilustración 55: Lista de usuarios AzureAD UOC

Con las herramientas de desarrollador del navegador, accesibles con la tecla F12, en la pestaña de red, se pueden observar las llamadas hacía la API de Azure que se realizan de forma transparente al usuario del portal. En ella se puede encontrar una consulta hacía el *Graph* de Windows (ilustración 56), donde se administran los usuarios de Microsoft en formato grafo y las relaciones con aristas a los vértices de objetos (aplicaciones, usuarios, acciones, etc.).

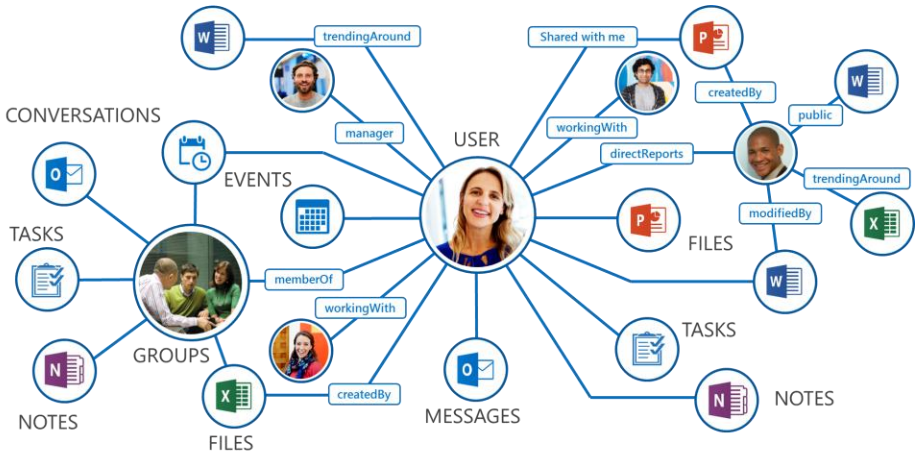


Ilustración 56: Grafo Microsoft Graph (Microsoft, 2019)

De tal manera, la llamada, tipo GET, que realiza el navegador es la siguiente: https://graph.windows.net/myorganization/users?api-version=1.6-internal&$top=51&$select=displayName,userPrincipalName,objectId,alternativeSecurityIds,dirSyncEnabled,lastDirSyncTime,userType,mail,otherMails, donde se observan las *queries* que se lanzan sobre la entidad "users", filtro por top (primeros 51 resultados) y una serie de campos en la selección.

Conjuntamente, la llamada incluye las cabeceras que contiene la autorización con un token OAuth obtenido en el inicio de sesión, con una caducidad de entre 30min y una hora según se ha comprobado en la investigación, pero renovados automáticamente debido al token de refresco.

Se adjunta una captura de, por un lado, la llamada que realiza, seguidamente, de la URI explicada y finalmente del token de las cabeceras necesarias para validar la petición, ilustración 57.

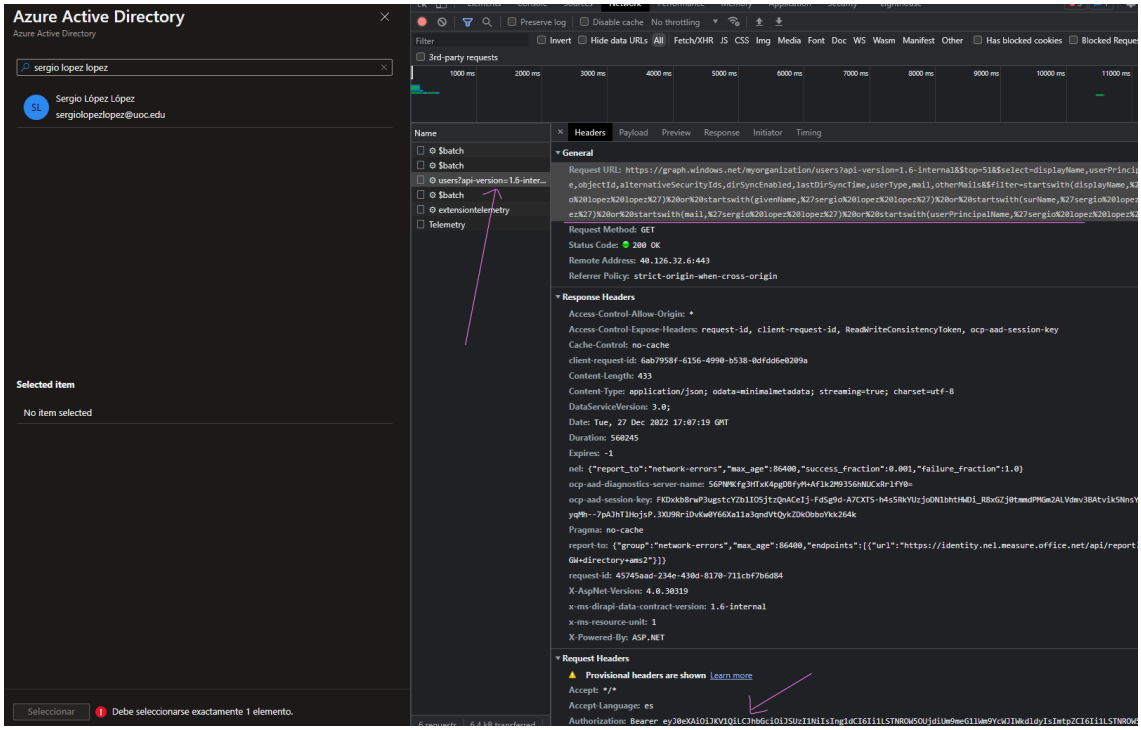


Ilustración 57: Llamada del portal a la API

Modificando la llamada, eliminando el filtro de "$top=51" la respuesta devuelve 100 resultados, y, por último, forzando con otro valor "$top=100000", la respuesta como máximo devuelve un resultado de 999 usuarios. El siguiente paso es automatizar dicha búsqueda, para iterar cada llamada utilizando la siguiente propiedad de la API, "*skiptoken*" que corresponde a la propiedad de la respuesta si existen más resultados paginados, "*odata.nextlink*", pudiéndose extraer al completo todos los usuarios de la organización, a pesar de la limitación por rol de usuario estudiante.

El script realizado –como requisito, se le debe introducir el token de autorización a mano, pudiéndose automatizar en una siguiente versión– para esta función realiza la siguiente lógica, realiza una primera llamada con el máximo de resultados y los almacena como primeros valores en una variable, y el enlace a la siguiente página de *skiptoken* en otra variable.

La siguiente llamada, al existir contenido en la variable *skiptoken*, realiza otro tipo de llamada añadiendo el filtro por paginación. De tal manera que el resultado de esta siguiente llamada se añade en el almacén anterior. Realizándose de forma continua hasta llegar al final, cuando el primer valor coincida con el resultado de la llamada de esa iteración.

El resultado se devuelve de forma interactiva, por cada iteración del bucle por pantalla como se muestra en la ilustración 58. Además, se añade una redirección final hacía un fichero tipo Excel, exportando todos los resultados para su futura consulta. Se añade el código empleado seguidamente.

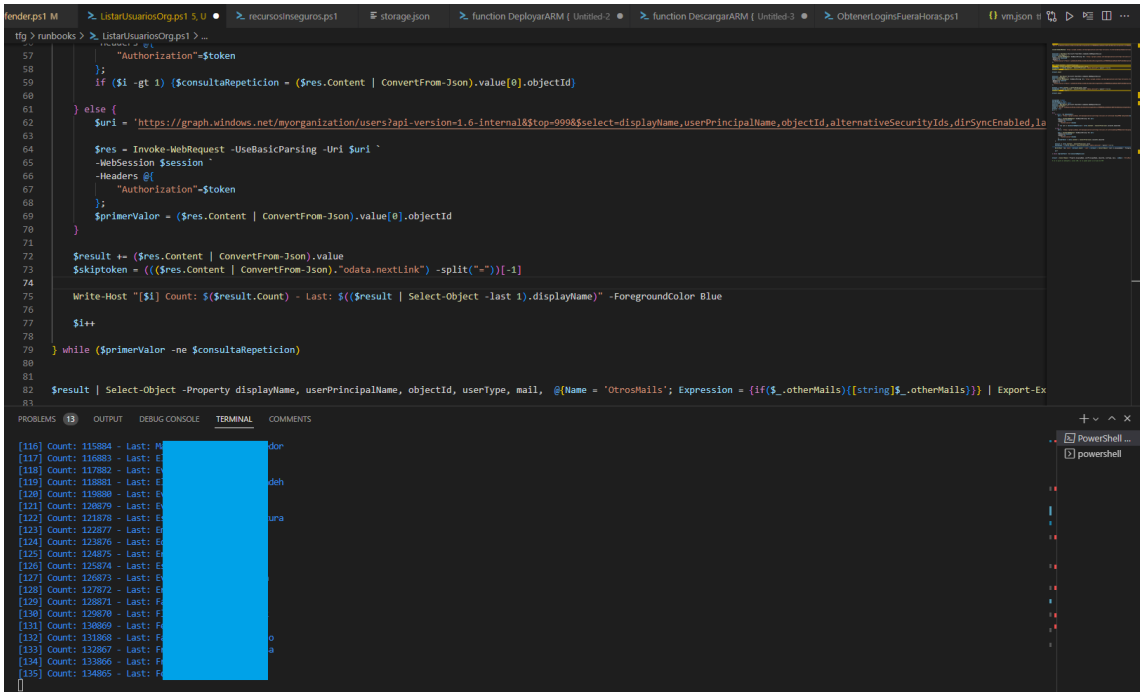


Ilustración 58: Resultado del script de listar todos los usuarios de la organización

Código del script ListarUsuariosOrg.ps1

```
$skiptoken = $null #Partimos de no tener skiptoken
$primerValor = $null #Partimos sin valores resultantes
$consultaRepeticion = $null #Partimos sin repeticiones
$session = New-Object Microsoft.PowerShell.Commands.WebRequestSession #Se crea un objeto tipo
WebRequestSession
$session.UserAgent = "" #Se asigna un UserAgent vacío, se evita fingerprint
$token = "Bearer eyJ0eXA..." # <--- Se debe añadir un token válido
$result = @() #Resultado vacío, se inicializa la array de result
$i = 1 #Se indica el indice del bucle a 1

do {
    if ($null -ne $skiptoken){
        $uri = "https://graph.windows.net/myorganization/users?api-version=1.6-internal" +
"&`$top=999&`$skiptoken=$skiptoken&`$select=displayName,userPrincipalName,objectId," +
        "alternativeSecurityIds,dirSyncEnabled,lastDirSyncTime,userType,mail,otherMails"

        #Llamada constante a la API paginadas con el skiptoken del previo resultado
        $res = Invoke-WebRequest -UseBasicParsing -Uri $uri `
        -WebSession $session `
        -Headers @{
            "Authorization"=$token
        };
        if ($i -gt 1) {$consultaRepeticion = ($res.Content | ConvertFrom-Json).value[0].objectId}

    } else {
        $uri = 'https://graph.windows.net/myorganization/users?api-version=1.6-internal' +
```

```
        '&$top=999&$select=displayName,userPrincipalName,objectId,alternativeSecurityIds,' +
        'dirSyncEnabled,lastDirSyncTime,userType,mail,otherMails'

        #Llamada inicial a la API
        $res = Invoke-WebRequest -UseBasicParsing -Uri $uri `
        -WebSession $session `
        -Headers @{
            "Authorization"=$token
        };
        $primerValor = ($res.Content | ConvertFrom-Json).value[0].objectId
    }

    $result += ($res.Content | ConvertFrom-Json).value #Incrementa el resultado de la llamada
    $skiptoken = ((($res.Content | ConvertFrom-Json)."odata.nextLink") -split("="))[-1] #Asigna
dinamicamente un nuevo skiptoken

    Write-Host "[$i] Count: $($result.Count) - Last: $(($result | Select-Object -last
1).displayName)" -ForegroundColor Blue

    $i++ #Incrementa el indice

} while ($primerValor -ne $consultaRepeticion)

#Devuelve el resultado y lo exporta a un Excel
$result | Select-Object -Property displayName, userPrincipalName, objectId, userType, mail, `
        @{Name = 'OtrosMails'; Expression = {if($_.otherMails){[string]$_.otherMails}}} `
        | Export-Excel -Path .\users.xlsx -WorksheetName "usuariosUOC"
```

9.2. Organigrama de la startup

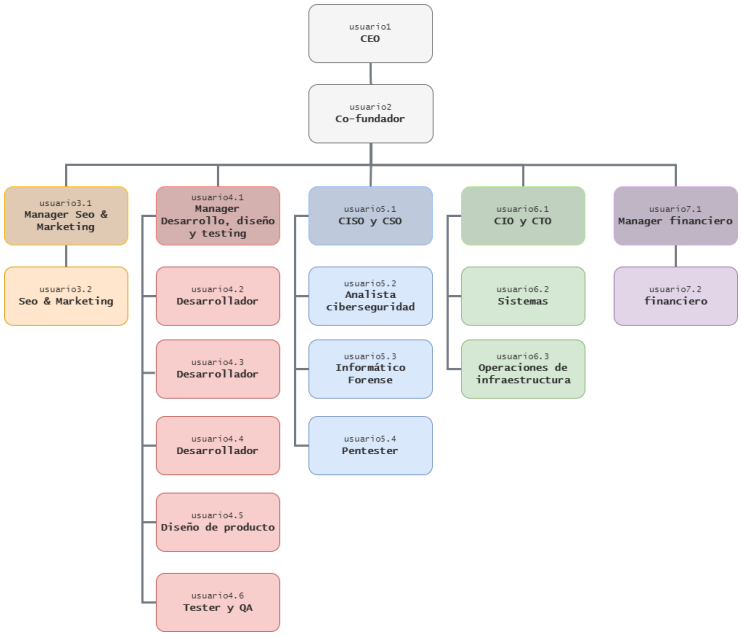


Ilustración 59: Organigrama Startup

Como se puede ver en el organigrama ilustrado en la imagen 59, la startup que se utiliza durante las pruebas de securización de la infraestructura dispone de un tamaño pequeño de organización, por ende, la forma en la que se debe proteger estará enfocado a esta situación.

Los usuarios que se mencionan durante la realización del proyecto son, todo el equipo de desarrollo (usuarios 4.x), el usuario inicial de la startup, el CEO y por último, los empleados de operaciones de infraestructura y sistemas de forma conjunta con el CISO.

De forma específica, el equipo de desarrollo será el cliente principal de los servicios y recursos desplegados en Azure, y, por otro lado, el equipo de operaciones junto con sistemas administrará la infraestructura que el CISO de seguridad auditará. Los roles que se llevarán a cabo de forma compartida son los ya mencionados: gestionar el estándar y directiva de seguridad, el cumplimiento normativo, la inteligencia sobre amenazas, la arquitectura y la seguridad de la aplicación y los datos, la administración de identidades y la preparación de informes en respuesta a incidentes.

9.3. Servicios y recursos de Azure

En este apartado, se indicarán algunas de las configuraciones interesantes –fuera del *scope* central del hilo del proyecto– que se han llevado a cabo durante el estudio de la creación de la infraestructura de la startup.

9.3.1. Blueprints

Este servicio permite a los desarrolladores compilar y crear nuevos entornos siguiendo unos estándares prefijados por el equipo de operaciones, con los que se cumplen los criterios de seguridad de la organización.

Esto se realiza mediante una serie de plantillas. Los pasos que se deben seguir para configurar una de las plantillas existentes, son los siguientes:

- Crear directiva y aplicarla sobre una o varias subscripciones:

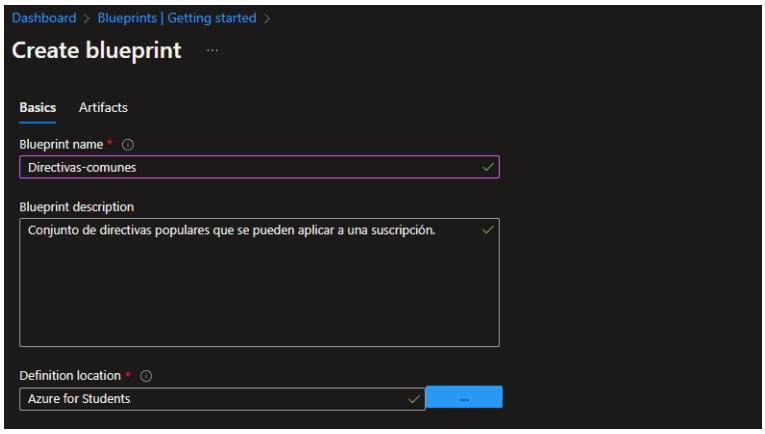


Ilustración 60: Paso 1 Blueprints

- Asignar artefactos a diferentes niveles

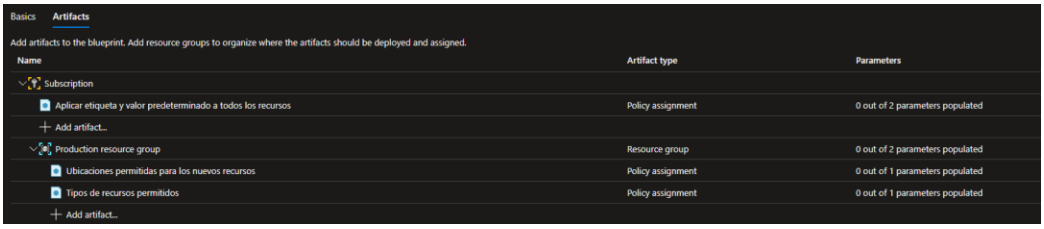


Ilustración 61: Paso 2 Blueprints

En el ejemplo, se puede ver como se indica a nivel de subscripción la regla de aplicar un tag predeterminada a todos los recursos (para su seguimiento) o a nivel de grupo de recursos, se indica que ubicaciones son las permitidas (por cuestiones legales por ejemplo) y que tipo de recursos se pueden desplegar.

- Una vez creadas se deben publicar con una versión (para tener un control de versiones) y asignar los valores necesarios, ilustración 62.

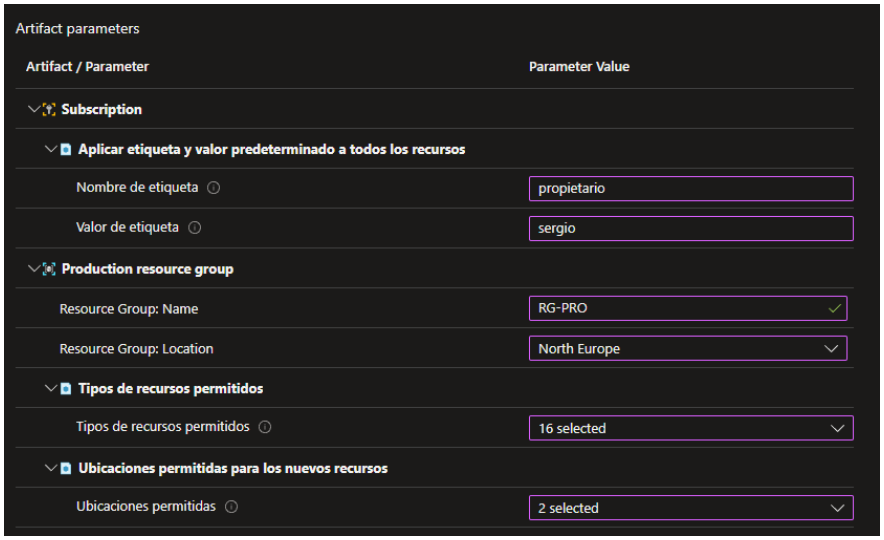


Ilustración 62: Paso 3 Blueprints

- Comprobación al crear un recurso

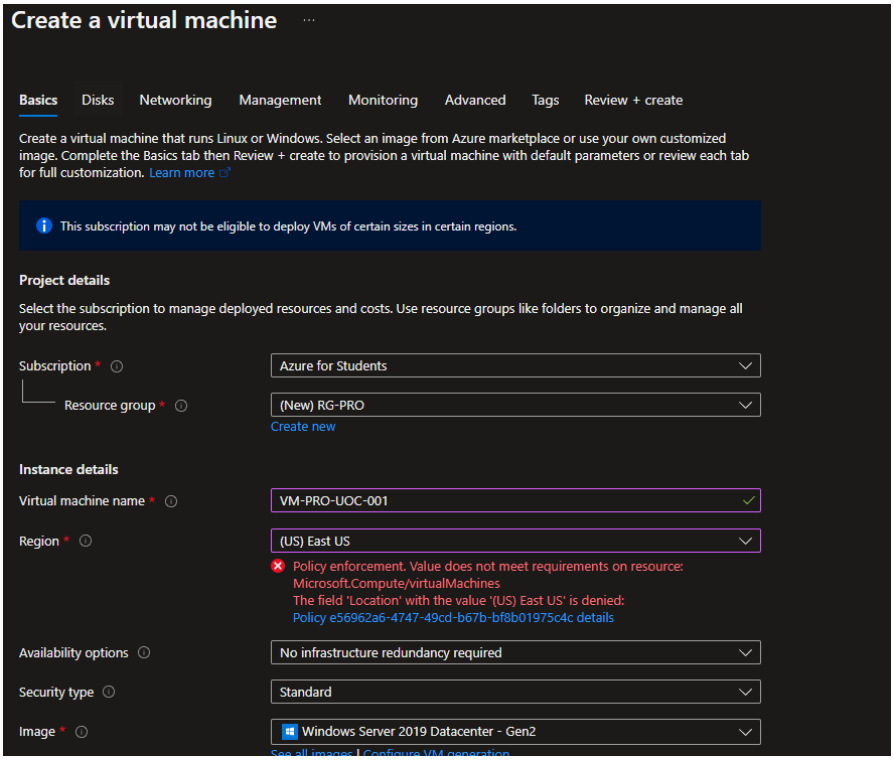


Ilustración 63: Test nombre Blueprints

Se puede observar en la imagen 63 que no permite la creación de la maquina virtual en esa zona y en la imagen 64, como no se puede acceder desde otra localización (con una VPN externa, se ha seleccionado RUSIA) debido a las politicas creadas en Azure Blueprints.

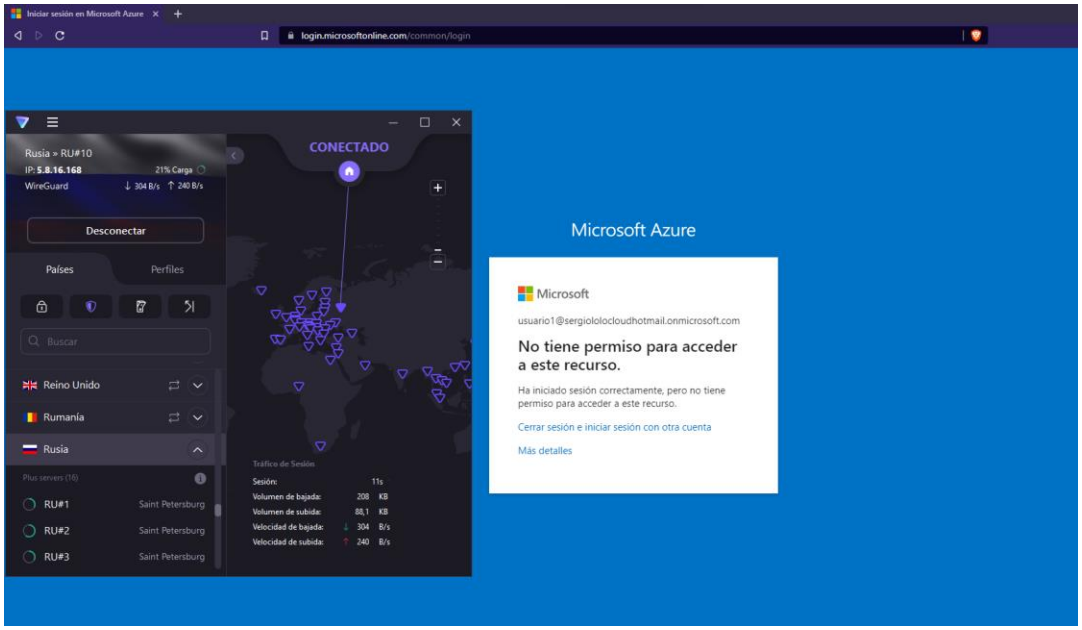


Ilustración 64: Test ubicación Blueprints

9.3.2. VPN

Una de las ventajas que ofrece el servicio de VPN de Azure es la captura de paquetes en el gateway de la VPN, de tal manera que el administrador puede controlar el tráfico y encontrar alteraciones a investigar, ilustración 65.

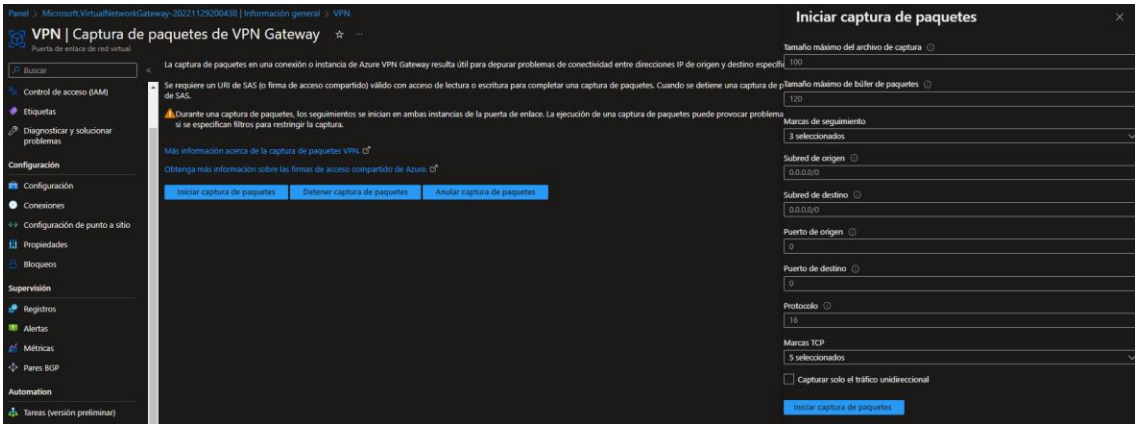


Ilustración 65: Captura de paquetes en el gateway VPN

9.3.3. Firewall

Seguidamente, se muestra una serie de capturas sobre el firewall creado para la organización (ilustraciones 66-72).

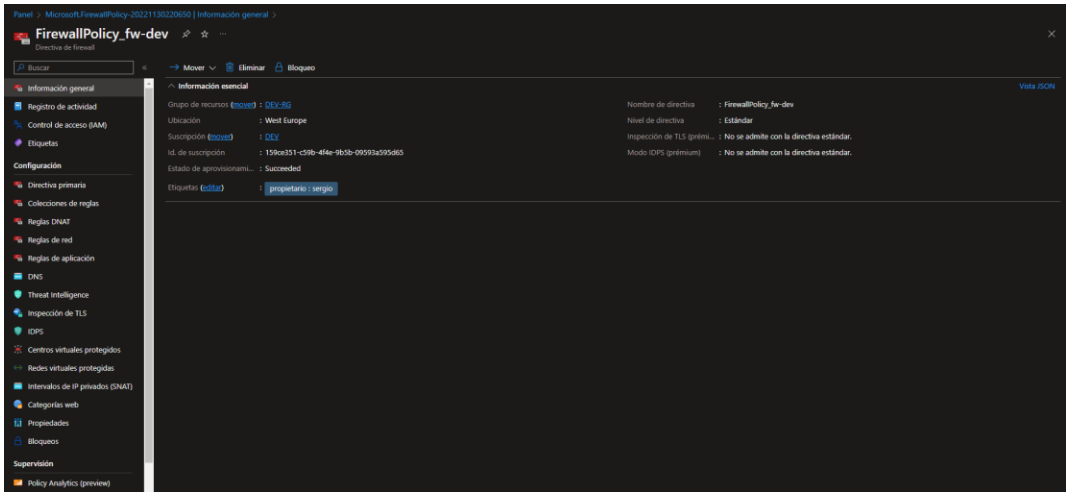


Ilustración 66: Directiva de firewall creado para la red de desarrollo

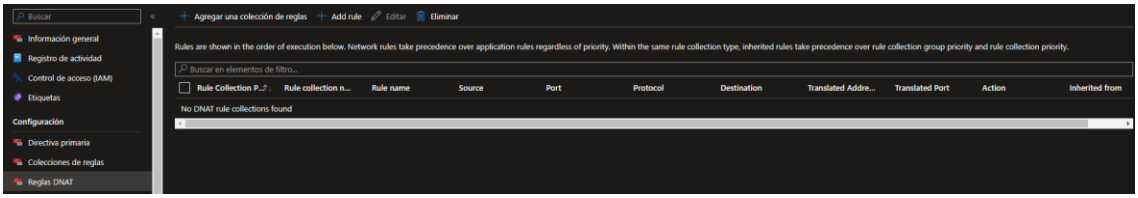


Ilustración 67: Apartado de reglas DNAT

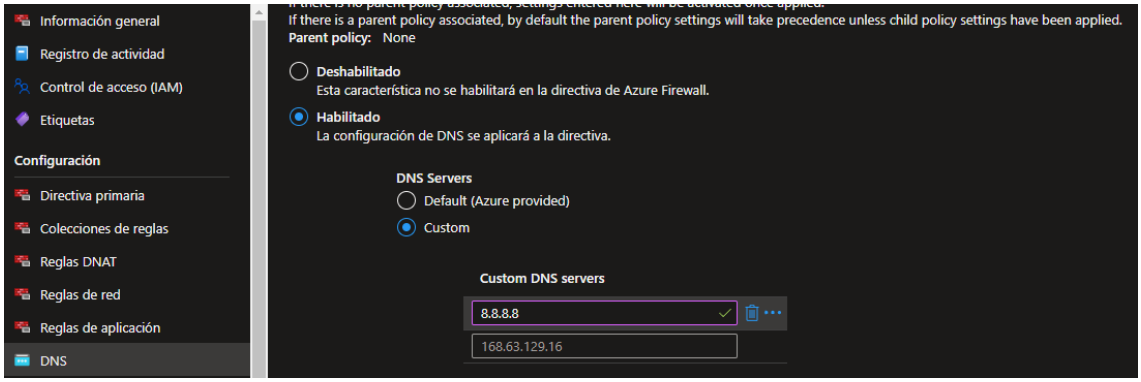


Ilustración 68: Apartado de resolución de nombres en el firewall

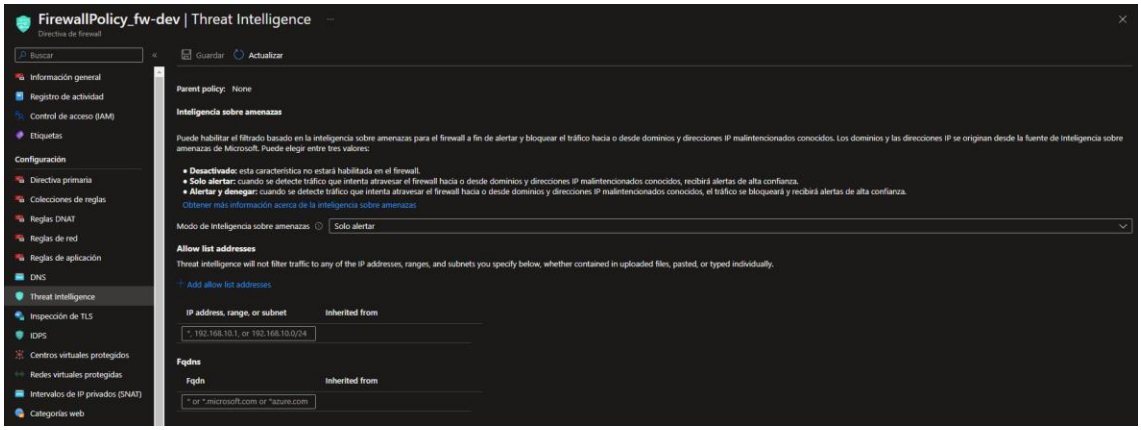


Ilustración 69: Aplicación de Threat Intelligence del firewall

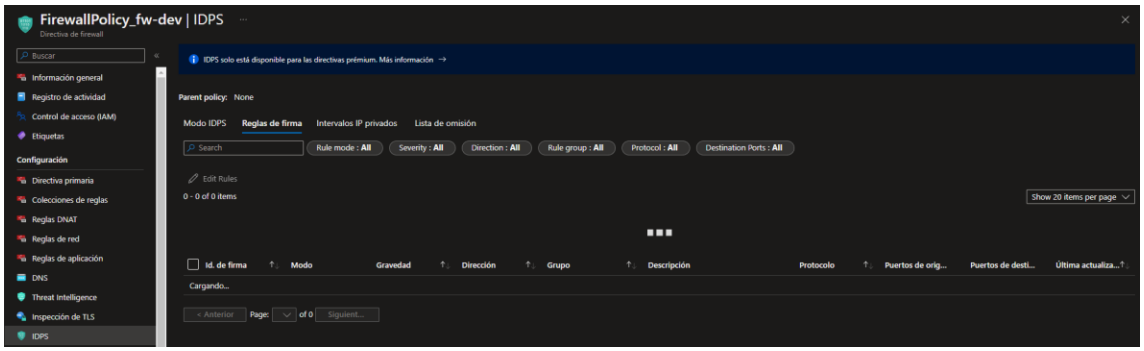


Ilustración 70: Capacidad IDPS

Ilustración 71: Capacidad de que el IDPS inspeccione los paquetes cifrados TLS

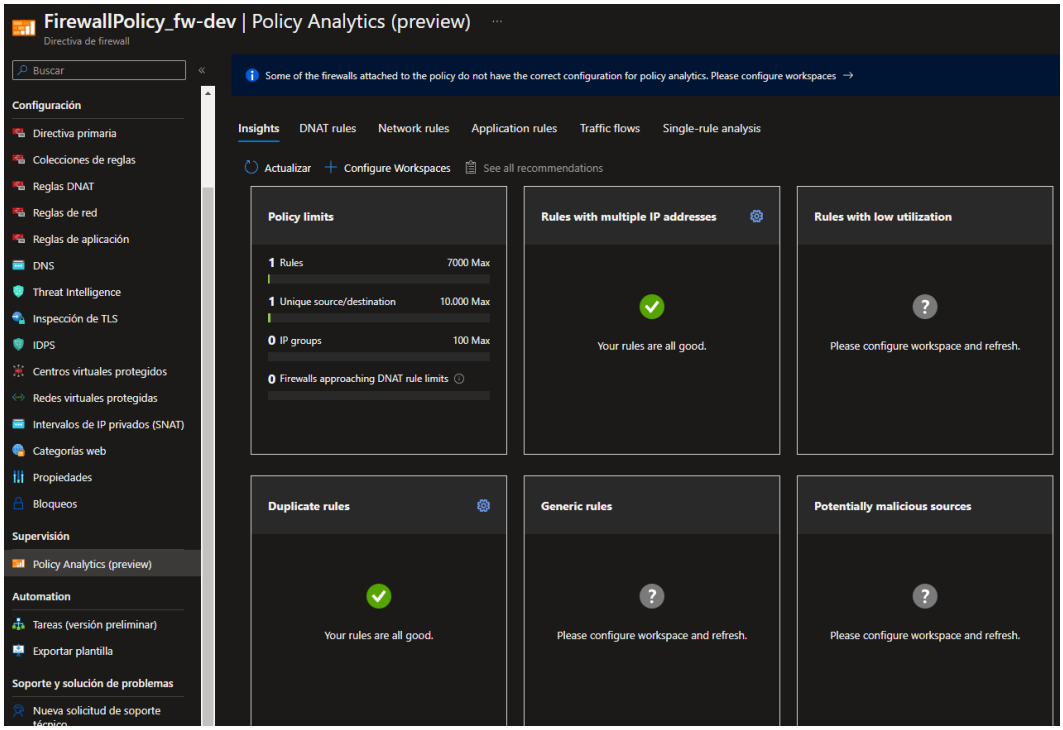


Ilustración 72: Analisis de politicas de firewall

9.3.4. Gateway de aplicación

Como se puede comprobar en la siguiente imagen 73 durante la creación de un *application Gateway*, este puede trabajar como WAF.

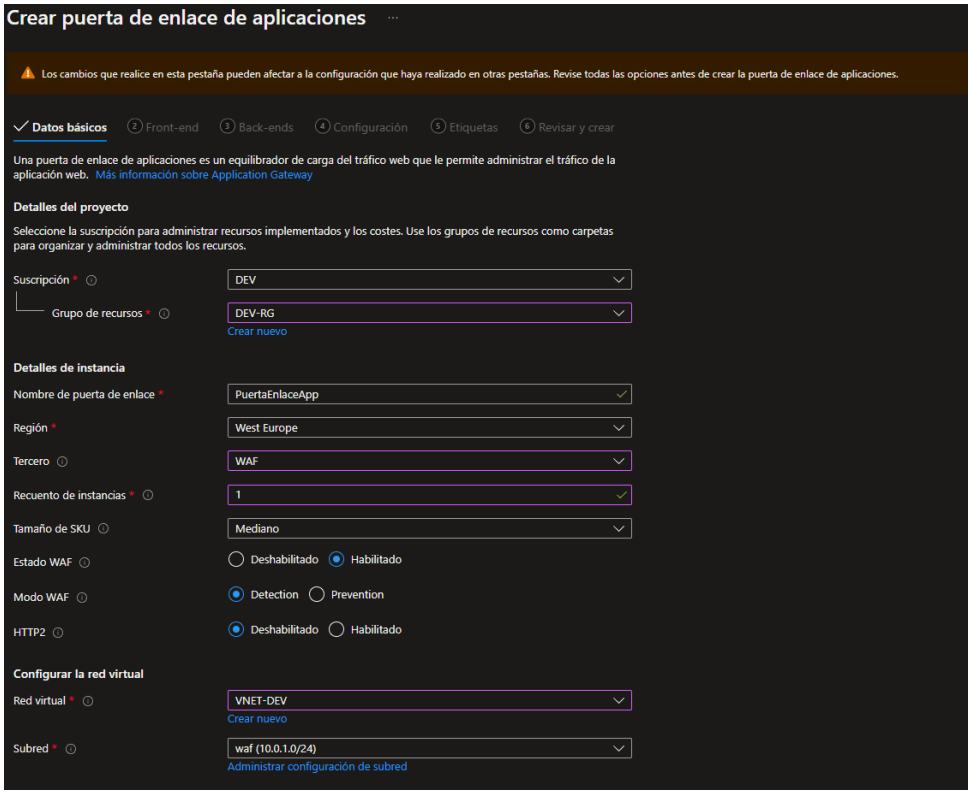


Ilustración 73: Creación con WAF del AppGW

9.3.5. Bastion

Durante la creación de la red virtual, se puede seleccionar el direccionamiento (segmentado) de la red que utilizarán las maquinas con Bastion, ilustración 74, de esta manera, toda la comunicación se hace de manera aislada.

Ilustración 74: Direccionamiento subred Bastion

9.3.6. Cuenta de almacenamiento

Captura de configuración de acceso para la organización a un recurso compartido en la unidad de red alojada en la cuenta de almacenamiento, accesible con Azure AD DS.

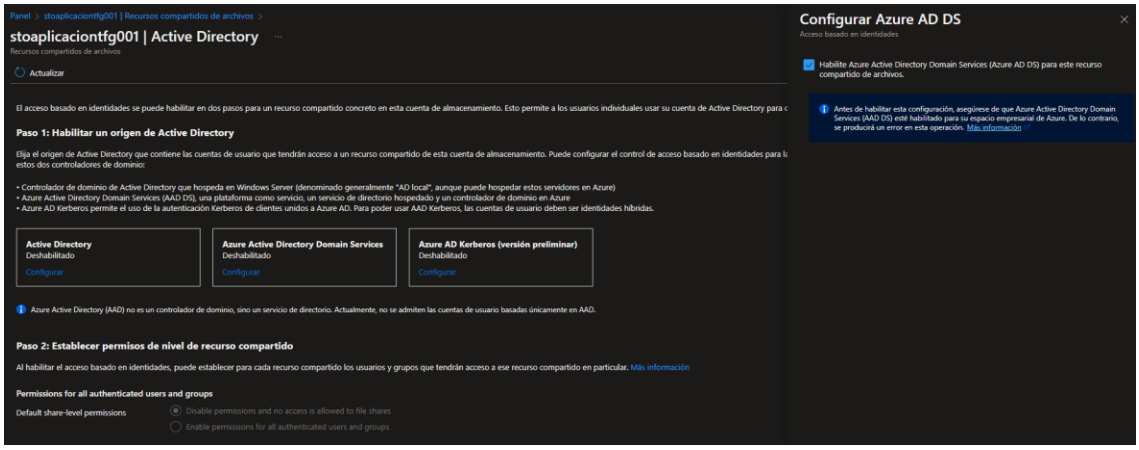


Ilustración 75: Sincronización Cuenta de almacenamiento con AzureAD

9.4. Herramientas externas de seguridad:

Resultado tras el uso de alguna de las herramientas existentes utilizadas para auditar términos en Seguridad de Azure:

- Azure Resource Inventory. Web: https://github.com/microsoft/ARI

Conjunto de scripts de *Powershell* que generan un reporte de todos los recursos disponibles en Azure, además muestra gráficas asociadas a diferentes categorías: recomendaciones de coste, seguridad, etc.

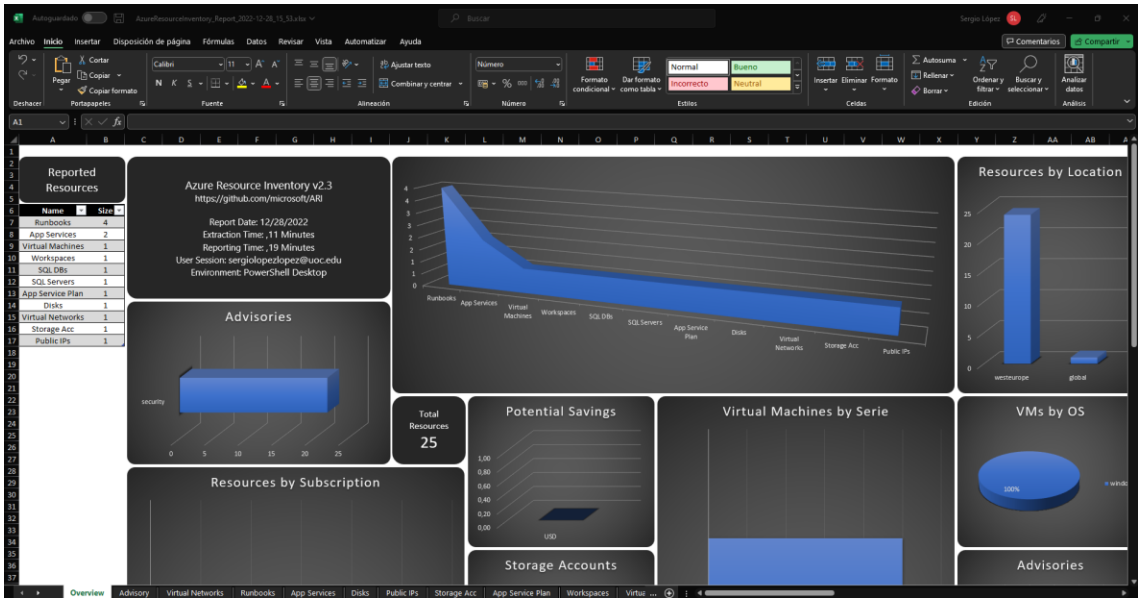


Ilustración 76: Reporte ARI

- PowerZure. Web: https://powerzure.readthedocs.io/en/latest/index.html

Conjunto de scripts escritos en *Powershell* que realizan comprobaciones y explotan los recursos con vulnerabilidades encontradas. De esta manera, provee una serie de comandos para diversas situaciones. En la imagen 77 se puede observar las siguientes casuísticas:

- `Get-AzureSQLDB -All`: Se obtienen todas bases de datos y servidores de la organización.
- `Show-AzureStorageContent -All`: Se obtienen todas cuentas de almacenamiento de la organización.
- `Get-AzureADDeviceOwener`: Muestra los dispositivos conectados con el AzureAD, así como el sistema operativo, el propietario, etc.
- `Get-AzureADAppOwner`: Muestra las aplicaciones del AzureAD y su creador o dueño.

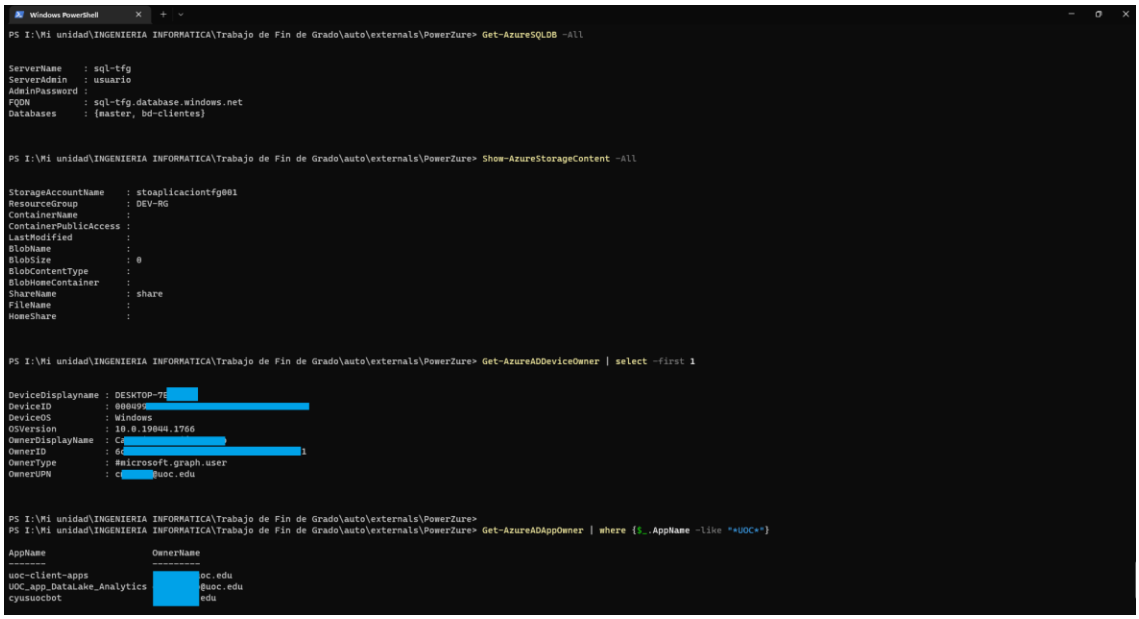


Ilustración 77: Reporte PowerZure

- **TerraGoat**. Web: https://github.com/bridgecrewio/terragoat

Utilizando *Terraform* –software de infraestructura como código desarrollado por *HashiCorp*– TerraGoat aloja un repositorio de infraestructura vulnerable de diferentes proveedores *cloud* (como Azure), es decir, permite desplegar infraestructura como código con *terraform* con malas configuraciones a reparar. De tal manera, ayuda a entrenar a sistemas de seguridad y los propios equipos responsables.

En la siguiente imagen se observa el funcionamiento de dicho repositorio usando los comandos básicos para la creación de la infraestructura en *Terraform*, “init” para validar y preparar las plantillas, seguido de “apply” para aplicar los cambios. Además, permite la eliminación rápida y segura de los recursos creados por TerraGoat con el comando “destroy”.

```
PS I:\Mi unidad\INGENIERIA INFORMATICA\Trabajo de Fin de Grado\auto\externals\terragoat\terraform\azure> ..\terraform.exe init -reconfigure -backend-config="resource_group_name=$TERRAGOAT_RESOURCE_GROUP" `
>>     -backend-config "storage_account_name=$TERRAGOAT_STATE_STORAGE_ACCOUNT" `
>>     -backend-config="container_name=$TERRAGOAT_STATE_CONTAINER" `
>>     -backend-config "key=$TF_VAR_environment.terraform.tfstate"

Initializing the backend...

Successfully configured the backend "azurerm"! Terraform will automatically
use this backend unless the backend configuration changes.

Initializing provider plugins...
- Finding hashicorp/azurerm versions matching ">= 2.0.0"...
- Finding latest version of hashicorp/random...
- Installing hashicorp/azurerm v3.37.0...
- Installed hashicorp/azurerm v3.37.0 (signed by HashiCorp)
- Installing hashicorp/random v3.4.3...
- Installed hashicorp/random v3.4.3 (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: Version constraints inside provider configuration blocks are deprecated

    on provider.tf line 3, in provider "azurerm":
     3:   version = ">= 2.0.0"

  Terraform 0.13 and earlier allowed provider version constraints inside the provider configuration block, but that is now deprecated and will be removed in a future version of Terraform. To silence this
  warning, move the provider version constraint into the required_providers block.


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.
PS I:\Mi unidad\INGENIERIA INFORMATICA\Trabajo de Fin de Grado\auto\externals\terragoat\terraform\azure>
```

Ilustración 78: Creación infra vulnerable TerraGoat

9.5. Esquema de flujo de los automatismos

Los diagramas de cada script siguen la leyenda mostrada en la imagen 79, y corresponden con las siguientes partes:

- **Fases**: separación lógica de las distintas partes del script, recopilación de información, transformación y gestión de datos, entre otras.
- **Acciones principales**: son abstracciones a gran escala de los comandos que se realizan en el script, es decir, indica la acción final aislada que deriva del resultado objetivo.
- **Acciones secundarias**: son abstracciones a gran escala de comandos o acciones no necesariamente implícitas en el código, que no son protagonistas del resultado objetivo.
- **Condicionales**: acciones con distintos resultados según un criterio a valorar, a más bajo nivel, declaración IF, IF-ELSE, SWITCH, FOR, WHILE, etc.

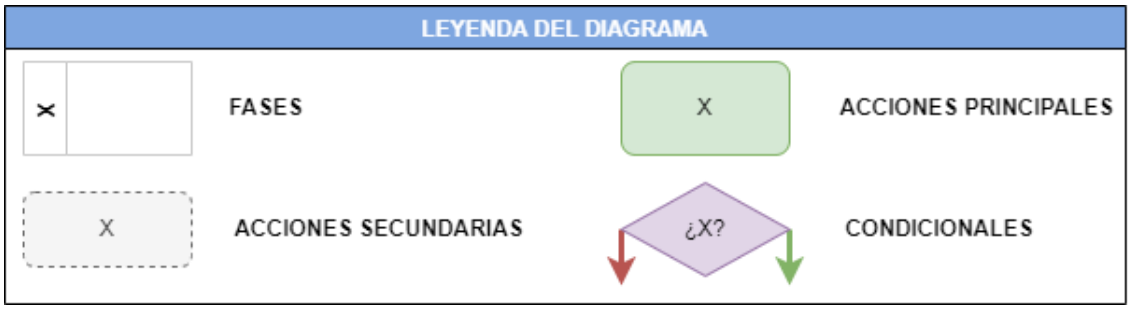
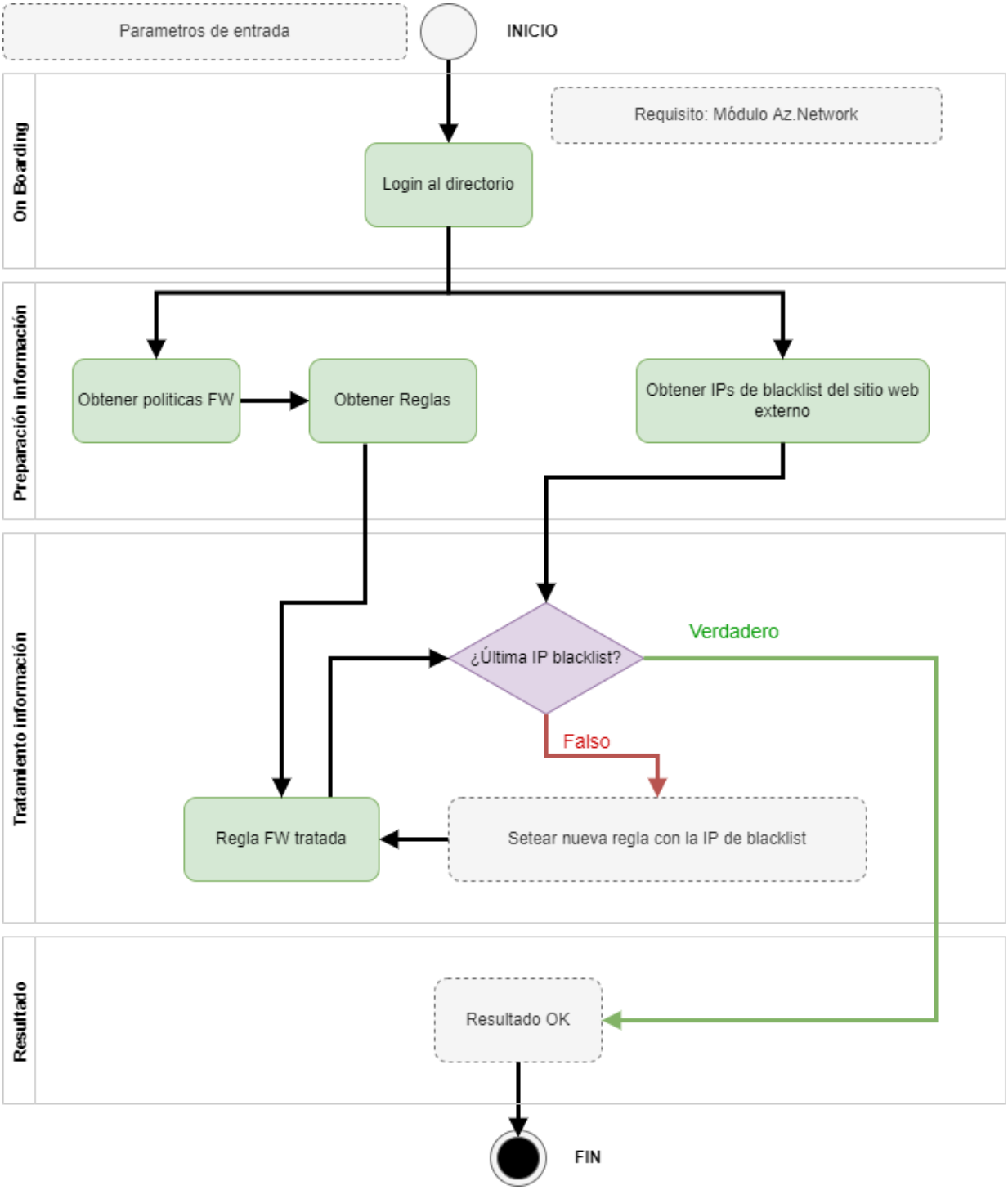
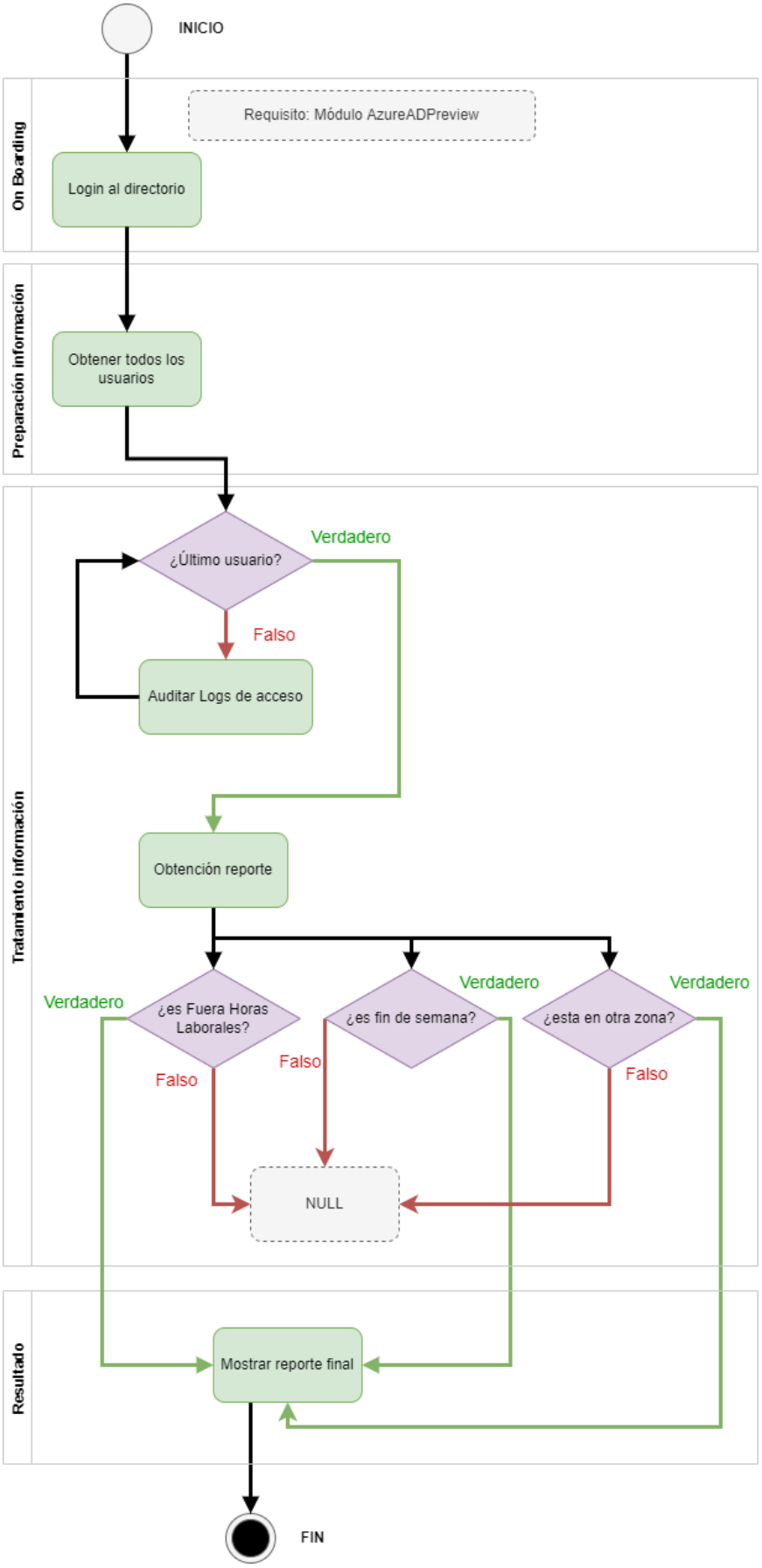


Ilustración 79: Leyenda diagramas

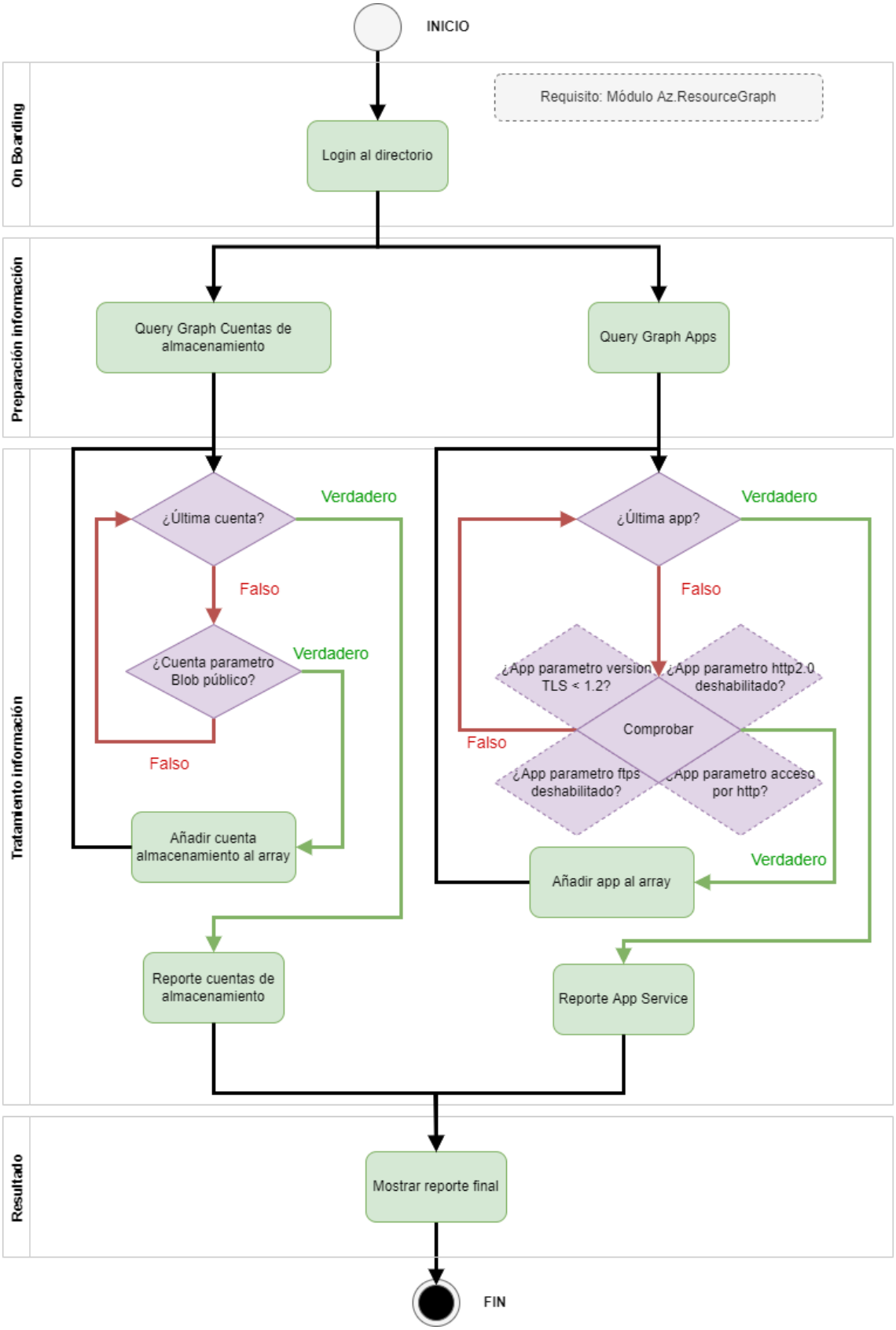
9.5.1. ActualizarPoliticasFireWall



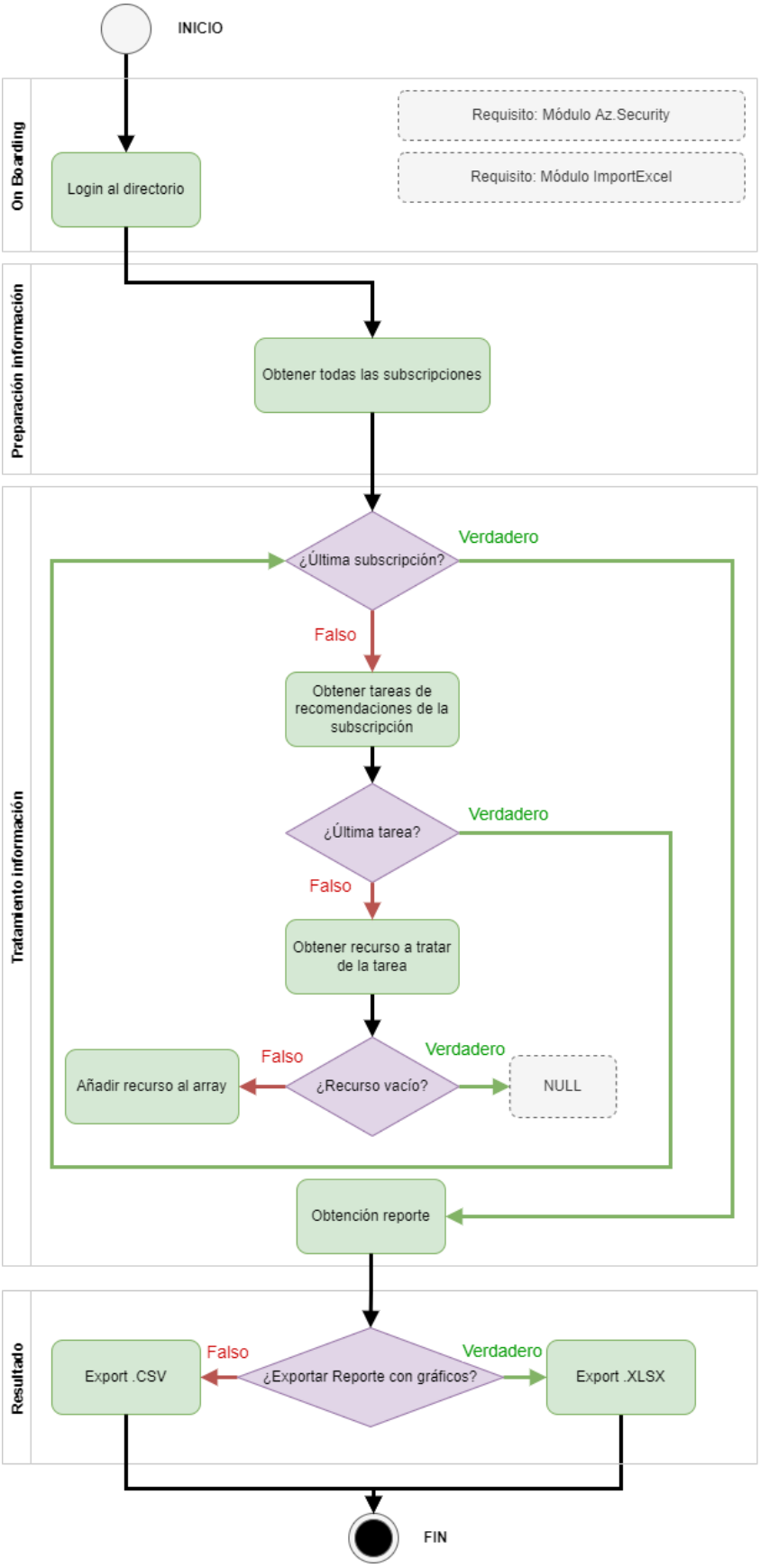
9.5.2. ObtenerLoginsFueraHoras



9.5.3. RecursosInseguros



9.5.4. ReportMicrosoftDefender



9.6. Output de la Herramienta Az Infra Sec

Se muestran las funciones que permite la herramienta por orden ascente del menú principal siguiente, ilustración 76.

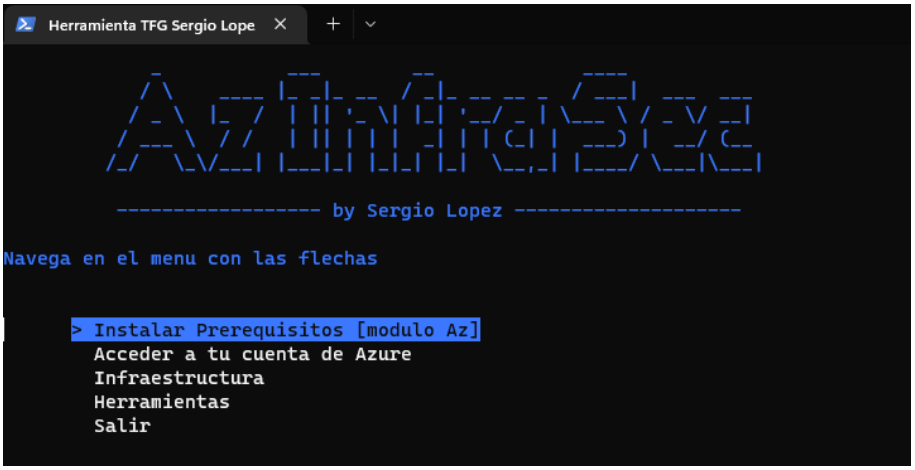


Ilustración 80: AzInfraSec Menu principal

- Instalar Prerequisitos [modulo Az]

En este primer punto, la herramienta instala los módulos necesarios para su funcionamiento, módulo Az (Az.Network, Az.Security, Az.ResourceGraph), AzureADPreview y ImportExcel. Una vez instalados, se importan en la sesión y esta vuelve al menú principal.

Ilustración 81: AzInfraSec - Menu instalación

- Acceso a la cuenta de Azure

El siguiente requisito para el funcionamiento de la herramienta es introducir el login con Microsoft Azure, además selecciona la subscripción objetivo (a pesar que en alguno de los automatismos recorrerán todas ellas). En la ilustración 82 se muestra el panel de terceros, en este caso la UOC, para autentificar la cuenta con Azure.

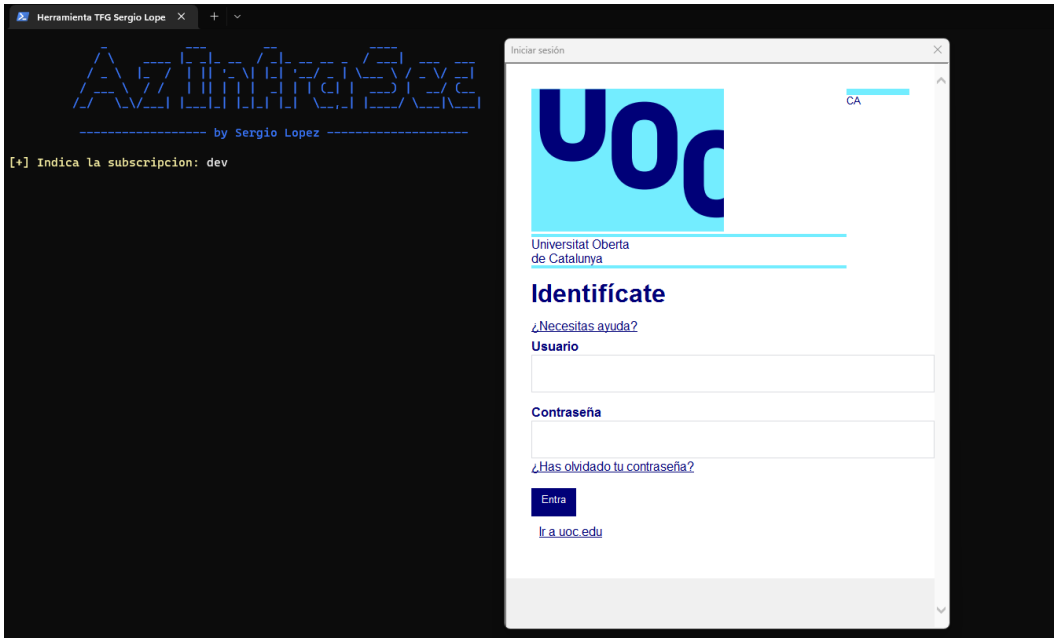


Ilustración 82: AzInfraSec Menu Login

- Menú Infraestructura

Ilustración 83: AzInfraSec Submenu infraestructura

> Máquina virtual

Dentro de la selección de máquina virtual se puede observar en la imagen 84 los inputs y en la imagen 85 el resultado en el portal de Azure de los recursos aprovisionados.

```
Herramienta TFG Sergio Lope        Windows PowerShell

------------------ by Sergio Lopez --------------------

Listado de recursos para crear posibles

>       Crear Maquina virtual

Crear? [S/N]: s
Nombre distintivo de la maquina (ej: backend) [*]: backend
Entorno o subscripcion (por defecto: dev): dev
Nombre usuario [*]: usuario5
Password usuario [*]: **************
Version de Sistema Operativo Windows (por defecto: 2019-datacenter-core-g2):
Tamano de la maquina (por defecto: Standard_B2s):
Sobre Red virtual existente? [nueva/existente [*]]: existente
Grupo de recursos de la red virtual (ej: DEV-RG) [*]: DEV-RG
Nombre de la subnet (ej: aplicacion1) [*]: aplicacion1
[v] Aprovisionamiento correcto

Presiona cualquier tecla para salir:
```

Ilustración 84: AzInfraSec herramienta creando VM

VM-BACKEND-DEV	Máquina virtual
VM-BACKEND-DEV-NIC	Interfaz de red
VM-BACKEND-DEV-NSG	Grupo de seguridad de red
VM-BACKEND-DEV_disk1_92ec8646a4244ae69ad2d02a37a85c48	Disco

Ilustración 85: Resultado creación mediante herramienta de VM

> App Service

Dentro de la selección de *App Service* se puede observar en la imagen 86 los inputs y en la imagen 87-88 el resultado en el portal de Azure de los recursos aprovisionados.

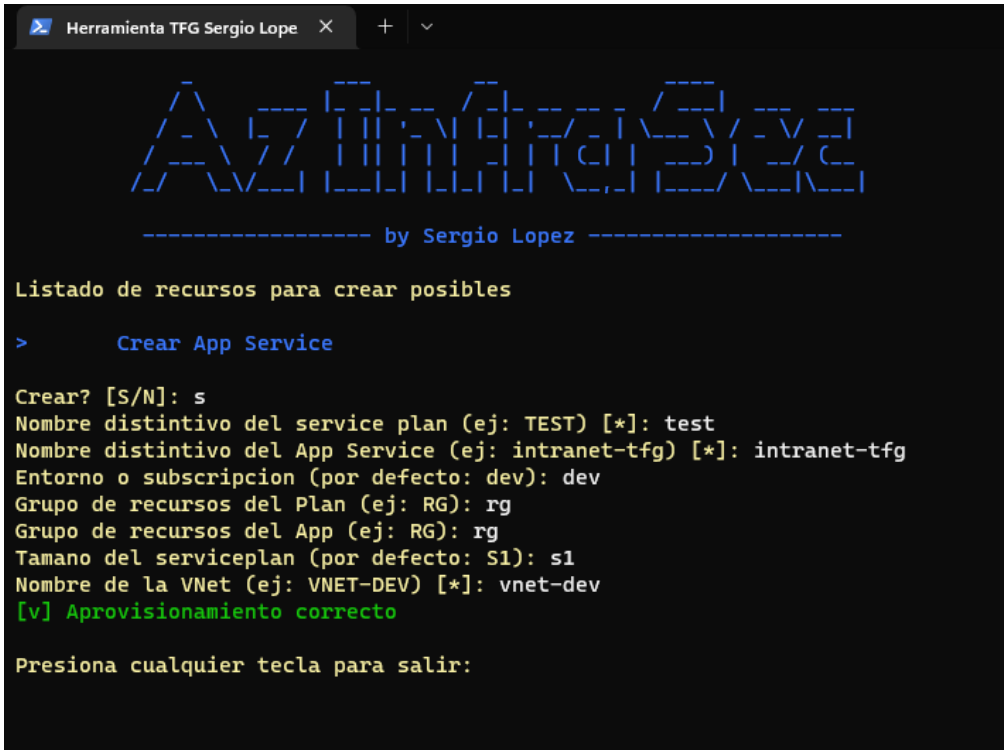


Ilustración 86: AzInfraSec herramienta creando App

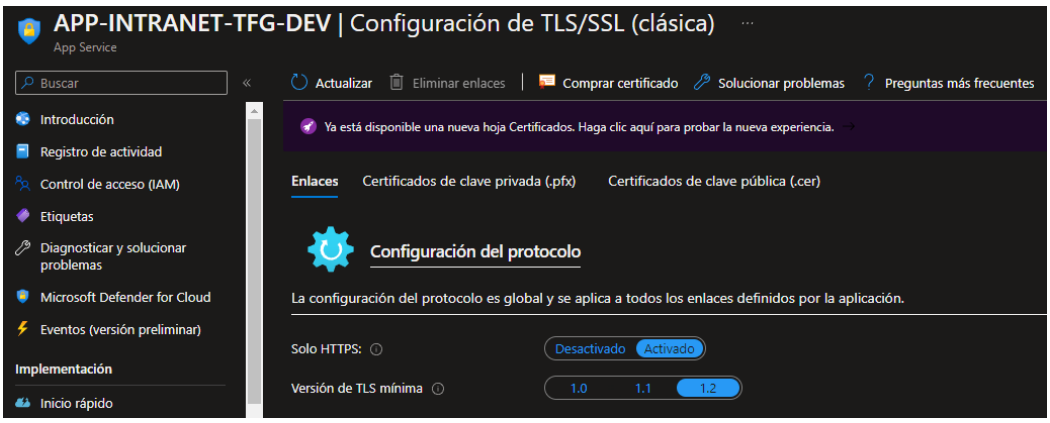


Ilustración 87: Resultado creación mediante herramienta de App

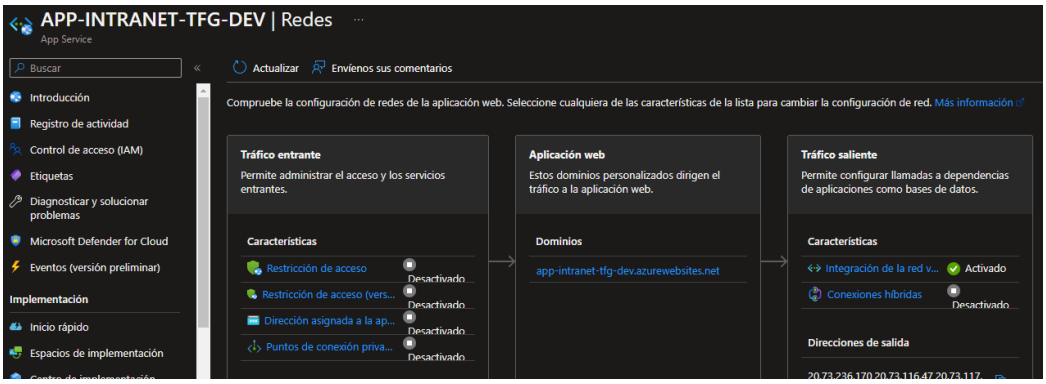


Ilustración 88: Resultado configuración con herramienta de App

> Cuenta de almacenamiento

Y, por último, la creación de la cuenta de almacenamiento se puede observar en la imagen 89 los inputs.

Ilustración 89: AzInfraSec herramienta creando Storage

- Menú de herramientas en scripts

Ilustración 90: AzInfraSec Submenu scripts

> Politicas de Azure Firewall

La ejecución de la inserción de nuevas políticas de Azure Firewall en función de las IPs de la lista negra de la web capturada en la imagen 91, tiene el mismo comportamiento que el mencionado en el apartado de automatización (punto 5), en la imagen 92 se muestran los inputs.

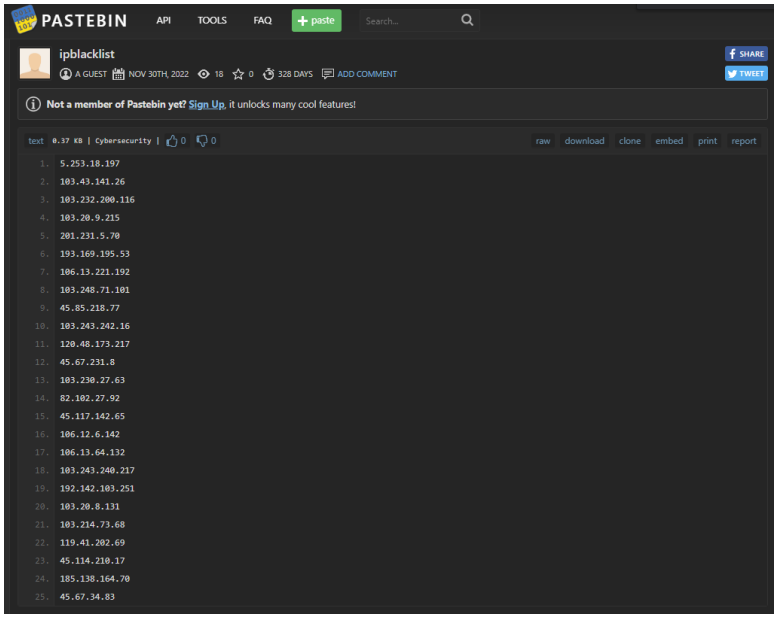


Ilustración 91: Lista negra IPs

Ilustración 92: AzInfraSec AFW

> Auditoria de acceso

Dentro de la selección del script de auditoría de accesos, se puede observar en la imagen 89 el mensaje de información y en la imagen 90 el resultado del reporte.

```
Herramienta TFG Sergio Lope    Windows PowerShell

 _          ___        __            ____
/ \    ____ |_ _|_ __  / _|_ __ __ _  / ___|  ___  ___
/ _ \  |_  /  | || '_ \| |_| '__/ _ | \___ \ / _ \/ __|
/ ___ \  / /   | || | | |  _| | | (_| |  ___) |  __/ (__
/_/   \_\/___| |___|_| |_|_| |_|  \__,_| |____/ \___|\___|

------------------ by Sergio Lopez --------------------

Listado de herramientas para securizar tu cuenta de Azure

>       Obtener informe de logins que tengan las siguientes caracteristicas:

        - Login fuera de horario laboral
        - Login en fin de semana
        - Login fuera de la localizacion

Ejecutar? [S/N]:
```

Ilustración 93: AzInfraSec info logins

```
Herramienta TFG Sergio Lope    Windows PowerShell

        - Login en fin de semana
        - Login fuera de la localizacion

Ejecutar? [S/N]: s

Nombre   Email                                          FechaAcceso           Dia IP              SistemaOperativo Buscador       Localidad
------   -----                                          -----------           --- --              ---------------- --------       ---------
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:48:59 Sunday 195.181.167.194 Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:49:55 Sunday 195.181.167.194 Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:49:58 Sunday 195.181.167.194 Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:53:22 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:57:10 Sunday 5.8.16.168      Windows 10       Chrome 108.0.0 Sankt-Peterburg - RU
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:57:25 Sunday 5.8.16.168      Windows 10       Chrome 108.0.0 Sankt-Peterburg - RU
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:57:50 Sunday 5.8.16.168      Windows 10       Chrome 108.0.0 Sankt-Peterburg - RU
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:58:22 Sunday 156.146.50.1    Windows 10       Chrome 108.0.0 Kyiv Misto - UA
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 12:58:56 Sunday 195.181.167.198 Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 13:00:03 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 13:00:16 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 13:01:19 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 13:01:21 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 13:01:23 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 16:57:08 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 16:57:11 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 18/12/2022 16:57:13 Sunday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 19/12/2022 0:09:55  Monday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 19/12/2022 0:09:58  Monday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES
usuario1 usuario1@sergiololocloudhotmail.onmicrosoft.com 19/12/2022 0:09:59  Monday 86.127.229.77   Windows 10       Chrome 108.0.0 Madrid - ES

Presiona cualquier tecla para salir:
```

Ilustración 94: AzInfraSec resultado logins

> Recursos inseguros

En el apartado de obtención de recursos inseguros, se comprueban las cuentas de almacenamiento con acceso público y las apps con TLS inseguros, el diseño del código es similar al mencionado, devolviendo el output ilustrado en la imagen 91.

```
Herramienta TFG Sergio Lope

 _          ___        __          ____
 / \   ____ |_ _|_ __  / _|_ __ __ _  / ___|  ___  ___
 / _ \  |_  /  | || '_ \| |_| '__/ _ | \___ \ / _ \/ __|
 / ___ \  / /   | || | | |  _| | | (_| |  ___) |  __/ (__
 /_/   \_\/___| |___|_| |_|_| |_|  \__,_| |____/ \___|\___|

 ----------------- by Sergio Lopez -------------------

Listado de herramientas para securizar tu cuenta de Azure

>       Obtener informe de los siguientes recursos inseguros:

        - Cuentas de almacenamiento publicas
        - App Services con TLS desprotegidos

Ejecutar? [S/N]: s
Revisando: stoaplicaciontfg01 (dev-rg)

Subscripcion ResourceGroup NombreStorageAccount BlobAccesoPublico NombreContainer ContainerAccesoPublico
------------ ------------- -------------------- ----------------- --------------- ----------------------
DEV          dev-rg        stoaplicaciontfg01   ▲ True            contenedor                   Container
DEV          dev-rg        stoaplicaciontfg01   ▲ True            blob                              Blob


Revisando: APP-INTRANET-TFG-DEV (dev-rg)
Revisando: appmintls01 (dev-rg)
Revisando: appmintls03 (dev-rg)
Revisando: apponlyhttps01 (dev-rg)

Subscripcion ResourceGroup NombreAppService MinTlsVersion Http20Enabled FtpsState HttpsOnly
------------ ------------- ---------------- ------------- ------------- --------- ---------
DEV          dev-rg        appmintls01      ▲ 1.0         ▲ False       FtpsOnly      True
DEV          dev-rg        appmintls03      ▲ 1.1         ▲ False       FtpsOnly      True
DEV          dev-rg        apponlyhttps01   1.2           ▲ False       FtpsOnly   ▲ False


Presiona cualquier tecla para salir:
```

Ilustración 95: AzInfraSec reporte recursos inseguros

En el que se pueden comprobar la veracidad de los resultados en las imagenes siguientes:

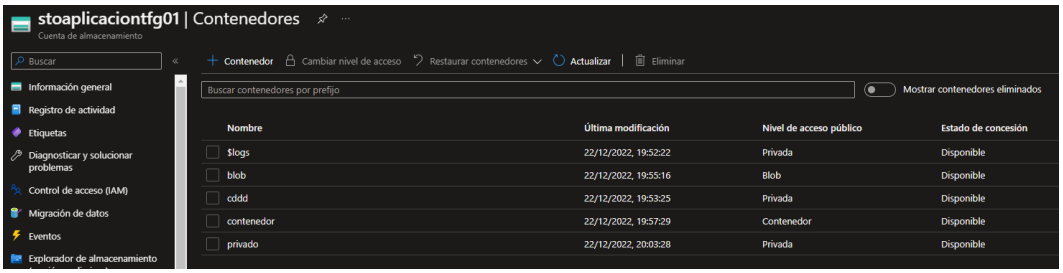


Ilustración 96: Contenedores con acceso público tipo contenedor y blob

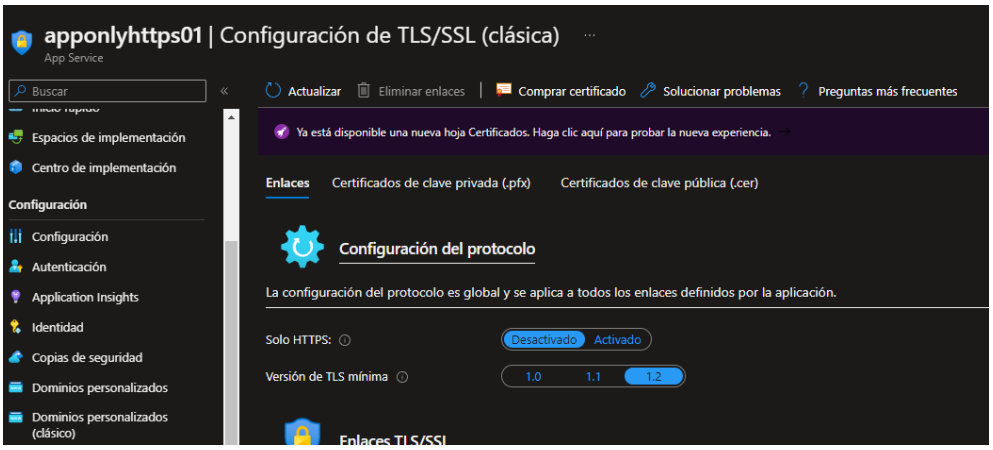


Ilustración 97: App con solo HTTPs desactivado

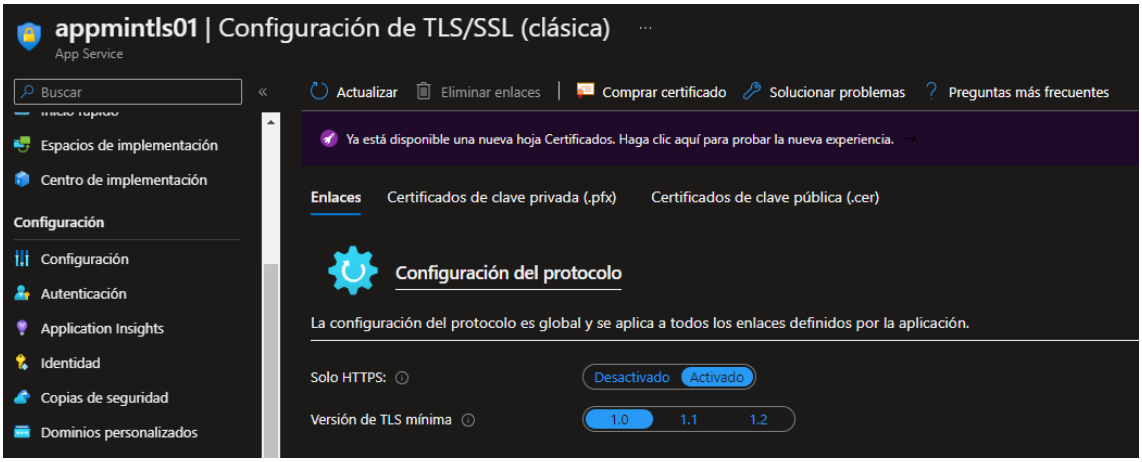


Ilustración 98: App con versión mínima TLS vulnerable

> Reporte Microsoft Defender for cloud

La última herramienta corresponde con la obtención del reporte a partir de las recomendaciones de Seguridad de *Microsoft Defender for Cloud*, la información del apartado se muestra en la imagen 95. Como ventaja diferencial frente a las demás herramientas cuya actividad es similar en la ejecución desde el automatismo como en el ejecutable AzInfraSec, es su salida, que desde la herramienta es únicamente a modo informativo, se muestra también en la imagen 95, sino también el Excel que va rellenando en cada ejecución.

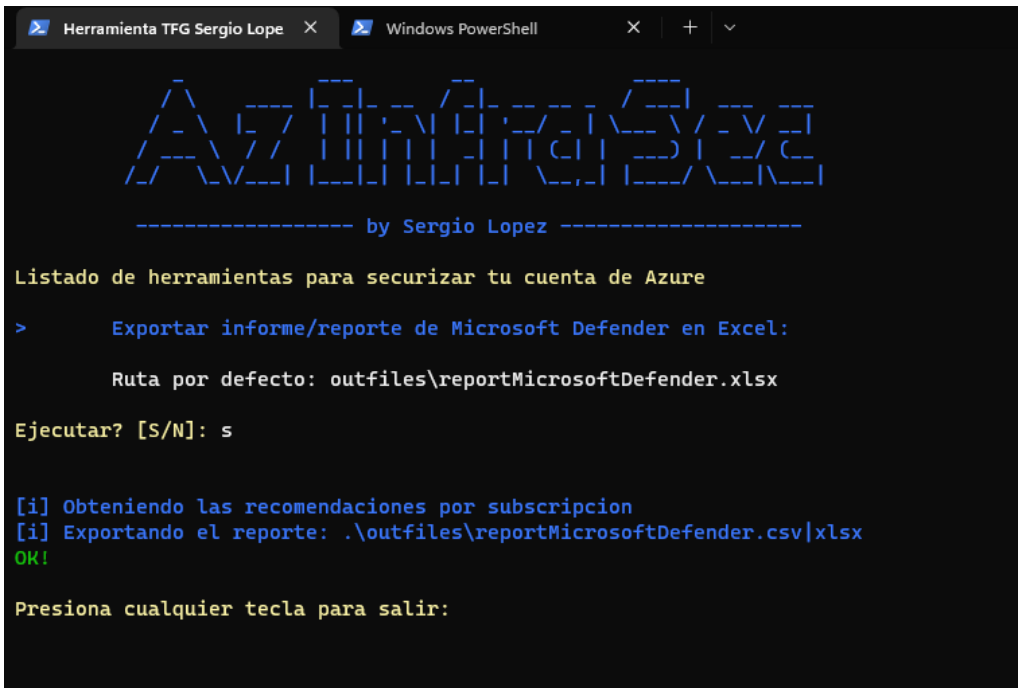
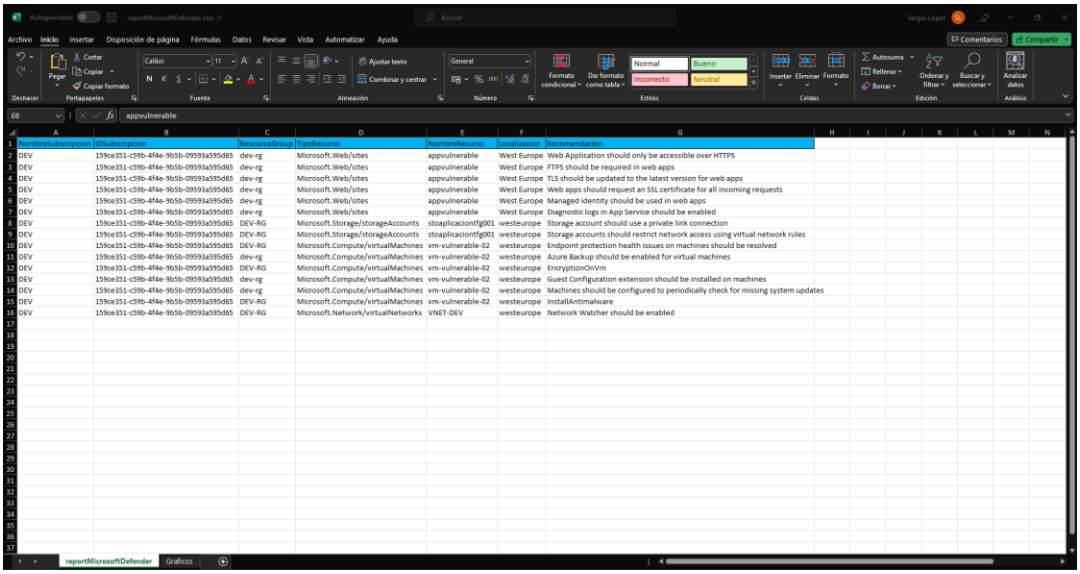


Ilustración 99: AzInfraSec reporte Microsoft Defender



NombreSubscripcion	IDSubscripcion	ResourceGroup	TipoRecurso	NombreRecurso	Localizacion	Recomendacion
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Web/sites	appvulnerable	West Europe	Web Application should only be accessible over HTTPS
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Web/sites	appvulnerable	West Europe	FTPS should be required in web apps
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Web/sites	appvulnerable	West Europe	TLS should be updated to the latest version for web apps
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Web/sites	appvulnerable	West Europe	Web apps should request an SSL certificate for all incoming requests
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Web/sites	appvulnerable	West Europe	Managed identity should be used in web apps
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Web/sites	appvulnerable	West Europe	Diagnostic logs in App Service should be enabled
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	DEV-RG	Microsoft.Storage/storageAccounts	stoaplicacionttfg001	westeurope	Storage account should use a private link connection
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	DEV-RG	Microsoft.Storage/storageAccounts	stoaplicacionttfg001	westeurope	Storage accounts should restrict network access using virtual network rules
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	DEV-RG	Microsoft.Compute/virtualMachines	vm-vulnerable-02	westeurope	Endpoint protection health issues on machines should be resolved
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Compute/virtualMachines	vm-vulnerable-02	westeurope	Azure Backup should be enabled for virtual machines
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	DEV-RG	Microsoft.Compute/virtualMachines	vm-vulnerable-02	westeurope	EncryptionOnVm
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Compute/virtualMachines	vm-vulnerable-02	westeurope	Guest Configuration extension should be installed on machines
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	dev-rg	Microsoft.Compute/virtualMachines	vm-vulnerable-02	westeurope	Machines should be configured to periodically check for missing system updates
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	DEV-RG	Microsoft.Compute/virtualMachines	vm-vulnerable-02	westeurope	InstallAntimalware
DEV	159ce351-c59b-4f4e-9b5b-09593a595d65	DEV-RG	Microsoft.Network/virtualNetworks	VNET-DEV	westeurope	Network Watcher should be enabled

Ilustración 100: Resultado herramienta Microsoft Defender

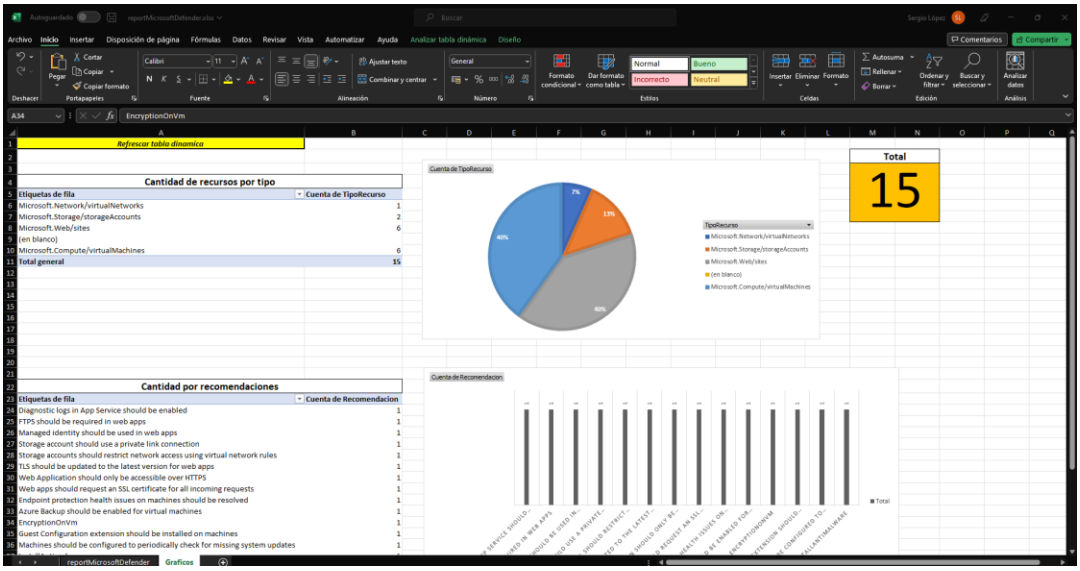


Ilustración 101: Resultado herramienta Microsoft Defender con gráficas

9.7. Código de plantillas y de scripts

- Plantilla vm.json

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "nombre": {
      "type": "string",
```

```
      "defaultValue": "xxx",
      "metadata": {
        "description": "Nombre"
      }
    },
    "entorno": {
      "type": "string",
      "defaultValue": "dev",
      "metadata": {
        "description": "entorno o subscripción"
      }
    },
    "nombreusuarioAdmin": {
      "type": "string",
      "metadata": {
        "description": "Nombre de usuario de la maquina virtual"
      }
    },
    "passwordAdmin": {
      "type": "secureString",
      "minLength": 12,
      "metadata": {
        "description": "Password de >12 caracteres de la maquina virtual"
      }
    },
    "VersionSO": {
      "type": "string",
      "defaultValue": "2019-datacenter-core-g2",
      "allowedValues": [
        "2008-R2-SP1",
        "2008-R2-SP1-smalldisk",
        "2012-Datacenter",
        "2012-datacenter-gensecond",
        "2012-Datacenter-smalldisk",
        "2012-datacenter-smalldisk-g2",
        "2012-Datacenter-zhcn",
        "2012-datacenter-zhcn-g2",
        "2012-R2-Datacenter",
        "2012-r2-datacenter-gensecond",
        "2012-R2-Datacenter-smalldisk",
        "2012-r2-datacenter-smalldisk-g2",
        "2012-R2-Datacenter-zhcn",
        "2012-r2-datacenter-zhcn-g2",
        "2016-Datacenter",
        "2016-datacenter-gensecond",
        "2016-datacenter-gs",
        "2016-Datacenter-Server-Core",
        "2016-datacenter-server-core-g2",
        "2016-Datacenter-Server-Core-smalldisk",
```

```
        "2016-datacenter-server-core-smalldisk-g2",
        "2016-Datacenter-smalldisk",
        "2016-datacenter-smalldisk-g2",
        "2016-Datacenter-with-Containers",
        "2016-datacenter-with-containers-g2",
        "2016-datacenter-with-containers-gs",
        "2016-Datacenter-zhcn",
        "2016-datacenter-zhcn-g2",
        "2019-Datacenter",
        "2019-Datacenter-Core",
        "2019-datacenter-core-g2",
        "2019-Datacenter-Core-smalldisk",
        "2019-datacenter-core-smalldisk-g2",
        "2019-Datacenter-Core-with-Containers",
        "2019-datacenter-core-with-containers-g2",
        "2019-Datacenter-Core-with-Containers-smalldisk",
        "2019-datacenter-core-with-containers-smalldisk-g2",
        "2019-datacenter-gensecond",
        "2019-datacenter-gs",
        "2019-Datacenter-smalldisk",
        "2019-datacenter-smalldisk-g2",
        "2019-Datacenter-with-Containers",
        "2019-datacenter-with-containers-g2",
        "2019-datacenter-with-containers-gs",
        "2019-Datacenter-with-Containers-smalldisk",
        "2019-datacenter-with-containers-smalldisk-g2",
        "2019-Datacenter-zhcn",
        "2019-datacenter-zhcn-g2",
        "2022-datacenter",
        "2022-datacenter-azure-edition",
        "2022-datacenter-azure-edition-core",
        "2022-datacenter-azure-edition-core-smalldisk",
        "2022-datacenter-azure-edition-smalldisk",
        "2022-datacenter-core",
        "2022-datacenter-core-g2",
        "2022-datacenter-core-smalldisk",
        "2022-datacenter-core-smalldisk-g2",
        "2022-datacenter-g2",
        "2022-datacenter-smalldisk",
        "2022-datacenter-smalldisk-g2"
      ],
      "metadata": {
        "description": "La versión de windows del sistema operativo de la maquina virtual"
      }
    },
    "tamano": {
      "type": "string",
      "defaultValue": "Standard_B2s",
      "metadata": {
```

```
        "description": "tamano de la maquina virtual"
      }
    },
    "localizacion": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "localizacion de los recursos"
      }
    },
    "VNet-Nueva-o-Existente": {
      "type": "string",
      "metadata": {
        "description": "Boolean que indica si es nueva (true) o existe (false) la vnet"
      }
    },
    "NombreVNet": {
      "type": "string",
      "defaultValue": "VNET-<env>",
      "metadata": {
        "description": "red virtual"
      }
    },
    "DireccionamientoVNet": {
      "type": "string",
      "defaultValue": "10.0.0.0/16",
      "metadata": {
        "description": "Prefijo de direccionamiento de la VNet"
      }
    },
    "DireccionamientoSubNet": {
      "type": "string",
      "defaultValue": "10.0.0.0/24",
      "metadata": {
        "description": "Prefijo de direccionamiento de la Subnet"
      }
    },
    "RGVNet": {
      "type": "string",
      "metadata": {
        "description": "grupo de recursos de la red virtual"
      }
    },
    "NombreSubnet": {
      "type": "string",
      "metadata": {
        "description": "SubRed"
      }
    }
```

```
  },
  "variables": {
    "NombreCuentaAlmacenamiento": "[format('bootdiags{0}',
uniqueString(resourceGroup().id))]",
    "NombreVM-1rtransformada": "[replace('VM-<nombre>-<env>', '<env>',
parameters('entorno'))]",
    "NombreVM-transformada": "[toUpper(replace(variables('NombreVM-1rtransformada'),
'<nombre>', parameters('nombre')))]",
    "NombreVNet-transformada": "[toUpper(replace(parameters('NombreVNet'), '<env>',
parameters('entorno')))]",
    "IdVNet": {
      "nueva": "[resourceId('Microsoft.Network/virtualNetworks',variables('NombreVNet-
transformada'))]",
      "existente":
"[resourceId(parameters('RGVNet'),'Microsoft.Network/virtualNetworks',variables('NombreVNet
-transformada'))]"
    },
    "IdSubnet": "[concat(variables('IdVNet')[parameters('VNet-Nueva-o-
Existente')],'/subnets/',parameters('NombreSubnet'))]",
    "NombreAdaptadorRed": "[concat(variables('NombreVM-transformada'),'-NIC')]",
    "NombreNSG": "[concat(variables('NombreVM-transformada'),'-NSG')]",
    "IdNSG":
"[resourceId(parameters('RGVNet'),'Microsoft.Network/networkSecurityGroups',variables('Nomb
reNSG'))]"
  },
  "resources": [
    {
      "condition": "[equals(parameters('VNet-Nueva-o-Existente'),'nueva')]",
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2021-02-01",
      "name": "[variables('NombreVNet-transformada')]",
      "location": "[parameters('localizacion')]",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[parameters('DireccionamientoVNet')]"
          ]
        },
        "subnets": [
          {
            "name": "[parameters('NombreSubnet')]",
            "properties": {
              "addressPrefix": "[parameters('DireccionamientoSubNet')]"
            }
          }
        ]
      }
    },
    {
```

```
      "type": "Microsoft.Network/networkSecurityGroups",
      "apiVersion": "2021-02-01",
      "name": "[variables('NombreNSG')]",
      "location": "[parameters('localizacion')]",
      "properties": {
        "securityRules": []
      }
    },
    {
      "type": "Microsoft.Network/networkInterfaces",
      "apiVersion": "2021-02-01",
      "name": "[variables('NombreAdaptadorRed')]",
      "location": "[parameters('localizacion')]",
      "properties": {
        "ipConfigurations": [
          {
            "name": "ipconfig1",
            "properties": {
              "subnet": {
                "id": "[variables('IdSubnet')]"
              }
            }
          }
        ],
        "enableAcceleratedNetworking": false,
        "enableIPForwarding": false,
        "networkSecurityGroup": {
            "id": "[variables('IdNSG')]"
        },
        "nicType": "Standard"

      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/virtualNetworks', variables('NombreVNet-
transformada'))]",
        "[resourceId('Microsoft.Network/networkSecurityGroups', variables('NombreNSG'))]"
      ]
    },
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2021-03-01",
      "name": "[variables('nombreVM-transformada')]",
      "location": "[parameters('localizacion')]",
      "properties": {
        "hardwareProfile": {
          "vmSize": "[parameters('tamano')]"
        },
        "osProfile": {
          "computerName": "[variables('nombreVM-transformada')]",
```

```
          "adminUsername": "[parameters('nombreusuarioAdmin')]",
          "adminPassword": "[parameters('passwordAdmin')]"
        },
        "storageProfile": {
          "imageReference": {
            "publisher": "MicrosoftWindowsServer",
            "offer": "WindowsServer",
            "sku": "[parameters('VersionSO')]",
            "version": "latest"
          },
          "osDisk": {
            "createOption": "FromImage",
            "managedDisk": {
              "storageAccountType": "[if(equals(parameters('entorno'),'dev'),
'StandardSSD_LRS', 'Premium_ZRS')]"
            }
          }
        },
        "networkProfile": {
          "networkInterfaces": [
            {
              "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('NombreAdaptadorRed'))]"
            }
          ]
        },
        "diagnosticsProfile": {
          "bootDiagnostics": {
            "enabled": false
          }
        }
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkInterfaces',
variables('NombreAdaptadorRed'))]"
      ]
    }
  ]
}
```

- Plantilla app.json

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "entorno": {
```

```
            "type": "string",
            "defaultValue": "dev",
            "metadata": {
                "description": "entorno o subscripción"
            }
        },
        "nombrePlan": {
            "defaultValue": "ASP-<entorno>",
            "type": "String",
            "metadata": {
                "description": "nombre del service plan"
            }
        },
        "tamano": {
            "type": "string",
            "defaultValue": "S1",
            "metadata": {
                "description": "tamano del service plan"
            }
        },
        "nombreApp": {
            "defaultValue": "aplicacion-<entorno>-tfg",
            "type": "String",
            "metadata": {
                "description": "nombre de la app service"
            }
        },
        "RGPlan": {
            "defaultValue": "<entorno>-RG",
            "type": "String",
            "metadata": {
                "description": "grupo de recursos del service plan"
            }
        },
        "RGVNet": {
            "defaultValue": "<env>-RG",
            "type": "String",
            "metadata": {
                "description": "grupo de recursos de la vnet"
            }
        },
        "nombreVNet": {
            "defaultValue": "VNET-<env>",
            "type": "String",
            "metadata": {
                "description": "nombre de la vnet"
            }
        },
        "localizacion": {
```

```
            "type": "string",
            "defaultValue": "[resourceGroup().location]",
            "metadata": {
                "description": "localizacion de los recursos"
            }
        }
    },
    "variables": {
        "NombrePlan-1rtransformada": "[replace('ASP-<nombre>-<env>', '<env>',
parameters('entorno'))]",
        "NombrePlan-transformada": "[toUpper(replace(variables('NombrePlan-
1rtransformada'), '<nombre>', parameters('nombrePlan')))]",
        "NombreApp-1rtransformada": "[replace('APP-<nombre>-<env>', '<env>',
parameters('entorno'))]",
        "NombreApp-transformada": "[toUpper(replace(variables('NombreApp-1rtransformada'),
'<nombre>', parameters('nombreApp')))]",
        "NombreVNet-transformada": "[toUpper(replace(parameters('nombreVNet'), '<env>',
parameters('entorno')))]",
        "NombreRGVNet-transformada": "[toUpper(replace(parameters('RGVNet'), '<env>',
parameters('entorno')))]"
    },
    "resources": [
        {
            "type": "Microsoft.Web/serverfarms",
            "apiVersion": "2022-03-01",
            "name": "[variables('NombrePlan-transformada')]",
            "location": "[parameters('localizacion')]",
            "sku": {
                "name": "[parameters('tamano')]",
                "tier": "Standard",
                "size": "[parameters('tamano')]",
                "family": "S",
                "capacity": 1
            },
            "kind": "app",
            "properties": {
                "perSiteScaling": false,
                "elasticScaleEnabled": false,
                "isSpot": false,
                "reserved": false
            }
        },
        {
            "type": "Microsoft.Web/sites",
            "apiVersion": "2022-03-01",
            "name": "[variables('NombreApp-transformada')]",
            "location": "[parameters('localizacion')]",
            "kind": "app",
            "properties": {
```

```
                "httpsOnly": true,
                "serverFarmId":
"[resourceId('Microsoft.Web/serverfarms',variables('NombrePlan-transformada'))]",
                "vnetRouteAllEnabled": true,
                "vnetImagePullEnabled": false,
                "vnetContentShareEnabled": false,
                "siteConfig": {
                    "numberOfWorkers": 1,
                    "acrUseManagedIdentityCreds": false,
                    "alwaysOn": false,
                    "http20Enabled": true,
                    "minTlsVersion": "1.2",
                    "ftpsState": "FtpsOnly"
                },
                "identity": {
                    "type": "SystemAssigned"
                },
                "virtualNetworkSubnetId":
"[concat(resourceId('Microsoft.Network/virtualNetworks',variables('NombreVNet-
transformada')), '/subnets/aplicacion')]",
                "keyVaultReferenceIdentity": "SystemAssigned"
            },
            "dependsOn": [
                "[resourceId('Microsoft.Web/serverfarms', variables('NombrePlan-
transformada'))]"
            ]
        },
        {
            "type": "Microsoft.Web/sites/virtualNetworkConnections",
            "apiVersion": "2022-03-01",
            "name": "[concat(variables('NombreApp-transformada'), '/ca1453ae-bb9a-4057-
b02b-6cf78c3355e8_aplicacion')]",
            "location": "[parameters('localizacion')]",
            "dependsOn": [
                "[resourceId('Microsoft.Web/sites', variables('NombreApp-transformada'))]"
            ],
            "properties": {
                "vnetResourceId": "[concat(resourceId(variables('NombreRGVNet-
transformada'),'Microsoft.Network/virtualNetworks',variables('NombreVNet-transformada')),
'/subnets/aplicacion')]",
                "isSwift": true
            }
        }
    ]
}
```

- Plantilla storage.json

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "entorno": {
            "type": "string",
            "defaultValue": "dev",
            "metadata": {
                "description": "entorno o subscripción"
            }
        },
        "nombre": {
            "defaultValue": "sto<entorno>",
            "type": "String",
            "metadata": {
                "description": "nombre del service plan"
            }
        },
        "localizacion": {
            "type": "string",
            "defaultValue": "[resourceGroup().location]",
            "metadata": {
                "description": "localizacion de los recursos"
            }
        }
    },
    "variables": {
        "NombreStorage-1rtransformada": "[replace('sto<nombre><env>', '<nombre>',
parameters('nombre'))]",
        "NombreStorage-transformada": "[toLower(replace(variables('NombreStorage-
1rtransformada'), '<env>', parameters('entorno')))]",
        "NombreVNet-transformada": "[toUpper(replace('VNET-<env>', '<env>',
parameters('entorno')))]"
    },
    "resources": [
        {
            "type": "Microsoft.Storage/storageAccounts",
            "apiVersion": "2022-05-01",
            "name": "[variables('NombreStorage-transformada')]",
            "location": "[parameters('localizacion')]",
            "sku": {
                "name": "Standard_LRS",
                "tier": "Standard"
            },
            "kind": "StorageV2",
            "properties": {
                "dnsEndpointType": "Standard",
                "publicNetworkAccess": "Enabled",
```

```
                "minimumTlsVersion": "TLS1_2",
                "allowBlobPublicAccess": false,
                "allowSharedKeyAccess": true,
                "networkAcls": {
                    "bypass": "AzureServices",
                    "virtualNetworkRules": [
                        {
                            "id":
"[concat(resourceId('Microsoft.Network/virtualNetworks',variables('NombreVNet-
transformada')), '/subnets/aplicacion')]",
                            "action": "Allow",
                            "state": "Succeeded"
                        }
                    ],
                    "ipRules": [],
                    "defaultAction": "Deny"
                },
                "supportsHttpsTrafficOnly": true,
                "encryption": {
                    "requireInfrastructureEncryption": false,
                    "services": {
                        "file": {
                            "keyType": "Account",
                            "enabled": true
                        },
                        "table": {
                            "keyType": "Account",
                            "enabled": true
                        },
                        "queue": {
                            "keyType": "Account",
                            "enabled": true
                        },
                        "blob": {
                            "keyType": "Account",
                            "enabled": true
                        }
                    },
                    "keySource": "Microsoft.Storage"
                },
                "accessTier": "Hot"
            }
        }
    ]
}
```

- Script ActualizarPoliticasFireWall.ps1

```
param(
    $politicaFW,
    $rg,
    $tipoPoliticaFW,
    $nombre,
    $nombreRegla,
    $protocolo,
    $prioridad,
    $puertoDestino
)

#Obtener lista negra
$IPblacklist = (Invoke-WebRequest -Uri https://pastebin.com/raw/RVArjEKJ).Content

#Función utilizada para conectarse con Azure
function Conectar {
    param(
        $subscripcion
    )
    Connect-AzAccount -Identity
    Set-AzContext -Subscription $subscripcion
}

#Función obtención de politica del FireWall
function ObtenerPoliticaFW {
    param(
        $politicaFW,
        $rg,
        $tipoPoliticaFW,
        $nombre
    )

    $politica = Get-AzFirewallPolicy -Name $politicaFW -ResourceGroupName $rg #Obtención
politicas del fireWall
    $ReglaObtenida = Get-AzFirewallPolicyRuleCollectionGroup -Name $tipoPoliticaFW -
ResourceGroupName $rg -AzureFirewallPolicyName $politicaFW #Obtención colección de reglas
    $ReglaMatcheada = $ReglaObtenida.Properties.RuleCollection | Where-Object {$_.Name -eq
$nombre} #Búsqueda de la regla a partir del nombre indicado

    #Retorno de un objeto con los campos necesarios
    return [PSCustomObject]@{
        politica = $politica
        ReglaObtenida = $reglaObtenida
        ReglaMatcheada = $ReglaMatcheada
    }
}
```

```
#Función para setear la politica configurada en el FireWall
function AnyadirPoliticaFW {
    param(
        $i,
        [PSObject]$out,
        $nombreRegla,
        $protocolo,
        $ipOrigen,
        $ipDestino,
        $puertoDestino,
        $prioridad,
        $tipoPoliticaFW
    )

    $params = @{
        Name                = "$($nombreRegla)_$i"
        Protocol            = $protocolo
        SourceAddress       = $ipOrigen
        DestinationAddress  = $ipDestino
        DestinationPort     = $puertoDestino
    }
    $nuevaRegla = New-AzFirewallPolicyNetworkRule @params #Creación de nueva regla de red

    $out.ReglaMatcheada.Rules.Add($nuevaRegla) #Asignación de la regla obtenida con la
nueva configuración

    #En caso de no indicar prioridad, se consultará la última para aplicarla al final
    if (!($prioridad)) {
        $prioridad = $out.ReglaMatcheada.Priority + ($out.ReglaMatcheada.RulesText |
ConvertFrom-Json).count
    }

    $params = @{
        Name                    = $tipoPoliticaFW
        FirewallPolicyObject    = $out.politica
        Priority                = $prioridad
        RuleCollection          = $out.ReglaObtenida.Properties.rulecollection
    }
    Set-AzFirewallPolicyRuleCollectionGroup @params #Seteo de la regla configurada
}

#Conexión subscripción de DEV (hardcoded, se puede cambiar)
Conectar "DEV"

#Obtención
$out = ObtenerPoliticaFW -politicaFW $politicaFW -rg $rg -tipoPoliticaFW $tipoPoliticaFW -
nombre $nombre

#Por cada IP de la lista negra, se asigna la nueva regla
```

```
foreach ($ip in ($IPblacklist -split "`n")) {
    AnyadirPoliticaFW -i (([DateTimeOffset]::Now.ToUnixTimeSeconds())+(Get-Random)) -
nombreRegla $nombreRegla -out $out -protocolo $protocolo -ipOrigen $ip -ipDestino $ip -
prioridad $prioridad -puertoDestino $puertoDestino -tipoPoliticaFW $tipoPoliticaFW
}
```

- ObtenerLoginsFueraHoras.ps1

```
#Horario de logins: de momento, no existe la limitación, pero es posible auditar los logins
y de esta manera controlar el acceso de
#los usuarios en lapsos de tiempo no permitidos, automatizable.

#Modulos requisitos
#Install-Module -Name AzureADPreview -Scope CurrentUser -AllowClobber
#Import-Module -Name AzureADPreview

#Conexión al módulo AzureAD
$mail = "x@uoc.edu" #Cuenta de la uoc no dispone de AzureAD disponible para estudiantes,
realizado con una cuenta de prueba externa
$tenantId = "28cf3eef-xxxxx-8af7ae2c7c90"
Connect-AzureAD -AccountId $mail -TenantId $tenantId

$Users = Get-AzureADUser #Obtención de usuarios

#Reporte por cada usuario obtenido
$report = foreach ($User in $Users) {
    Get-AzureADAuditSignInLogs -Filter "UserPrincipalName eq '$($User.UserPrincipalName)'"
| `
    Select-Object @{Name = 'Nombre'; Expression = {$_.UserDisplayName}}, `
                  @{Name = 'Email'; Expression = {$_.UserPrincipalName}}, `
                  @{Name = 'FechaAcceso'; Expression = {[DateTime]$_.CreatedDateTime}}, `
                  @{Name = 'Dia'; Expression = {([DateTime]$_.CreatedDateTime).DayOfWeek}},
`
                  @{Name = 'IP'; Expression = {$_.IpAddress}}, `
                  @{Name = 'SistemaOperativo'; Expression =
{$_.DeviceDetail.OperatingSystem}}, `
                  @{Name = 'Buscador'; Expression = {$_.DeviceDetail.Browser}}, `
                  @{Name = 'Localidad'; Expression = {"$($_.Location.State) -
$($_.Location.CountryOrRegion)"}}
}

#Funcion que controla si la fecha es fin de semana
function esFinDeSemana{
    param($fecha)

    $d = ([DateTime]$fecha).DayOfWeek.value__

    if(($d -eq 6) -or ($d -eq 0)){
```

```
        return $true
    }

    return $false
}

#Funcion que controla si la fecha esta dentro de horario laboral
function esFueraHorasLaborales{
    param($fecha)

    $h = ([DateTime]$fecha).Hour

    if(($h -ge 19) -or ($h -le 7)){
        return $true
    }

    return $false
}

#Funcion que controla la localización del usuario
function estaFueraZona{
    param($localizacion, $email)

    #Consulta con la BD de RRHH - Fuera del Scope
    if($false){
    if(moduloConsultaSQL($localizacion, $email)){
        return $true
    }
    }
    return $false
}

#Se obtiene reporte en funcion de las validaciones anteriores
$report | ForEach-Object {
    if( (esFinDeSemana($_.FechaAcceso)) -or (esFueraHorasLaborales($_.FechaAcceso)) -or
(estaFueraZona($_.Location.State, $_.UserPrincipalName)) ) {
        $_
    }
} | Sort-Object {$_.FechaAcceso} | Format-Table -AutoSize -Property *
```

- RecursosInseguros.ps1

```
#Obtener todas las subscripciones del tenant
$subs = Get-AzSubscription -TenantId '28cf3eef-a47f-48e5-981b-8af7ae2c7c90'

# Cuentas de almacenamiento públicas
function revisarStorageAccounts {
    #Query hacia el R.Graph para extraer nivel de StoAccount
    $queryStorageAccount =
```

```
@'
    resources
    | where type =~ 'Microsoft.Storage/storageAccounts'
    | extend allowBlobPublicAccess = parse_json(properties).allowBlobPublicAccess
    | where allowBlobPublicAccess == true
    | project subscriptionId, resourceGroup, name, allowBlobPublicAccess
'@

    #Consulta de la query x subscripción
    $reportStorageAccount = foreach ($sub in $subs){
        Search-AzGraph -Query $queryStorageAccount -Subscription $sub| ForEach-Object {
            $_
        }
    }

    #Reporte por cada storage ON -> Bucle por cada container
    $reportStorageAccount | ForEach-Object {
        $sub = (Get-AzContext).Subscription
        if ($sub.id -ne $_.subscriptionId){
            $null = Select-Azsubscription -SubscriptionId $_.subscriptionId
        }

        $storageAccount = Get-AzStorageAccount -ResourceGroupName $_.resourceGroup -Name
$_.name
        $ctx = $storageAccount.Context
        $container = Get-AzStorageContainerAcl -Context $ctx -ErrorAction SilentlyContinue

        Write-Host "Revisando: $($_.name) ($($_.resourceGroup))"

        $containers = $container | Where-Object -Property PublicAccess -ne "Off"

        foreach ($c in $containers) {
            [PSCustomObject]@{
                Subscripcion = $sub.Name
                ResourceGroup  = $_.resourceGroup
                NombreStorageAccount = $_.name
                BlobAccesoPublico = "⚠  $($_.allowBlobPublicAccess)"
                NombreContainer = $c.Name
                ContainerAccesoPublico = $c.PublicAccess
            }
        }
    } | Sort-Object {$_.NombreStorageAccount} | Format-Table -AutoSize -Property *
}

# App Service con TLS inferior a 1.2
function revisarAppService {
    $queryAppService =
@'
    resources
```

```
    | where type =~ 'Microsoft.Web/sites'
    | project subscriptionId, resourceGroup, name
'@

    #Consulta de la query x subscripción
    $AppServices = foreach ($sub in $subs){
        Search-AzGraph -Query $queryAppService -Subscription $sub| ForEach-Object {
            $_
        }
    }

    #Reporte por cada storage ON -> Bucle por cada container
    $AppServices | ForEach-Object {
        $sub = (Get-AzContext).Subscription
        if ($sub.id -ne $_.subscriptionId){
            $null = Select-Azsubscription -SubscriptionId $_.subscriptionId
        }

        $web = (Get-AzWebApp -ResourceGroupName $_.resourceGroup -Name $_.name)

        Write-Host "Revisando: $($_.name) ($($_.resourceGroup))"

        $riskTls = $false
        $riskHttp2 = $false
        $riskFtps = $false
        $riskHttps = $false

        if (($web.SiteConfig.MinTlsVersion) -ne "1.2") { $riskTls = $true}
        if (!([bool]($web.SiteConfig.Http20Enabled))) {$riskHttp2 = $true}
        if (($web.SiteConfig.FtpsState) -ne "FtpsOnly") {$riskFtps = $true}
        if (!([bool]($web.HttpsOnly))) {$riskHttps = $true}

        if ($riskTls -or $riskHttp2 -or $riskFtps -or $riskHttps){
            [PSCustomObject]@{
                Subscripcion = $sub.Name
                ResourceGroup  = $_.resourceGroup
                NombreAppService = $_.name
                MinTlsVersion = if ($riskTls) {"⚠  $($web.SiteConfig.MinTlsVersion)"}
else {$web.SiteConfig.MinTlsVersion}
                Http20Enabled = if ($riskHttp2) {"⚠  $($web.SiteConfig.Http20Enabled)"}
else {$web.SiteConfig.Http20Enabled}
                FtpsState = if ($riskFtps) {"⚠  $($web.SiteConfig.FtpsState)"} else
{$web.SiteConfig.FtpsState}
                HttpsOnly = if ($riskHttps) {"⚠  $($web.HttpsOnly)"} else
{$web.HttpsOnly}
            }
        }

    } | Sort-Object {$_.NombreAppService} | Format-Table -AutoSize -Property *
```

```
}

#Reporte Cuentas de almacenamiento
revisarStorageAccounts

#Reporte App Service
revisarAppService
```

- ReportMicrosoftDefender.ps1

```
$ErrorActionPreference = 'SilentlyContinue' #En caso de error, continua silenciosamente
$rutaReport = ".\outfiles" #Ruta por defecto del fichero
$nombreReport = "reportMicrosoftDefender" #Nombre del fichero excel (o csv)

$excel = $true #Si se precisa un report con graficas, es necesario indicarlo con $true

#Obtener todas las subscripciones del tenant
$subs = Get-AzSubscription #-TenantId '28cf3eef-a47f-48e5-981b-8af7ae2c7c90'

Write-Host "[i] Obteniendo las recomendaciones por subscripcion" -ForegroundColor Blue
$recomendaciones = @()
$recomendaciones = foreach($sub in $subs) {

    $null = Select-AzSubscription -SubscriptionId $sub.Id

    try {
        # Obtener todas las recomendaciones
        $tareas = Get-AzSecurityTask

        foreach($tarea in $tareas) {
            $recurso = (Get-AzResource -ResourceId $tarea.ResourceId)

            # Comprobar que el campo de recurso no está vacio y evitar falsos positivos
            if(!([string]::IsNullOrEmpty($recurso.Name))) {
                [PSCustomObject]@{
                    Recomendacion = $tarea.RecommendationType
                    NombreRecurso = $recurso.Name
                    NombreSubscripcion = $sub.Name
                    IDSubscripcion = ($tarea.ResourceId.Split("/")[2])
                    ResourceGroup = $recurso.ResourceGroupName
                    Localizacion = $recurso.Location
                    TipoRecurso = $recurso.ResourceType
                }
            }
        }
    }
    catch {
        Write-Host "No se han encontrado recomendaciones para la subscripcion: " $sub.Name
```

```
-ForegroundColor Red
    }
}

#Probar si es posible la exportación del reporte en Excel o CSV
try {
    Write-Host "[i] Exportando el reporte: $($rutaReport)\$($nombreReport).csv|xlsx" -
ForegroundColor Blue

    if (!($excel)){
        $recomendaciones | Select-Object "NombreSubscripcion", "IDSubscripcion",
"ResourceGroup", "TipoRecurso", "NombreRecurso", "Localizacion", "Recomendacion" `
        | Sort-Object -Property NombreRecurso | Export-Csv -Path ($rutaReport + "\" +
$nombreReport + ".csv") -Force -NoTypeInformation
    } else {
        $recomendaciones | Select-Object "NombreSubscripcion", "IDSubscripcion",
"ResourceGroup", "TipoRecurso", "NombreRecurso", "Localizacion", "Recomendacion" `
        | Sort-Object -Property NombreRecurso | Export-Excel -Path ($rutaReport + "\" +
$nombreReport + ".xlsx") -WorksheetName "reportMicrosoftDefender"
    }

    Write-Host "OK! `r`n" -ForegroundColor Green
}
catch {
    Write-Host "KO! Revisa la ruta, fichero o los permisos" -ForeGroundColor Red
}
```

- Herramienta de seguridad: Az Infra Sec

La herramienta, dada su complejidad, la extensión del script es demasiado elevada para adjuntarla en el documento, se aloja públicamente en el siguiente repositorio: https://dev.azure.com/sergiolopezlopez/tfg/_git/tfg?path=/Herramienta%20AzInfraSec/herramienta-seguridad.ps1