

**Estandares Cloud Azure**

V1.0

LIMA – PERÚ

Índice de contenido

[1.1 Histórico de versiones 5](#_Toc203734832)

[2 Introducción 6](#_Toc203734833)

[3 Marco Fundacional de Gobernanza y Operaciones en la Nube 6](#_Toc203734834)

[3.1 Principios del Modelo Operativo en la Nube: Adopción del Azure Well-Architected Framework 7](#_Toc203734835)

[3.2 Estrategia de Gobernanza y Cumplimiento 7](#_Toc203734836)

[3.2.1 Organización de Recursos 8](#_Toc203734837)

[3.2.2 Control de Acceso Basado en Roles (RBAC) 8](#_Toc203734838)

[3.2.3 Imposición de Estándares con Azure Policy 8](#_Toc203734839)

[3.3 Principios de FinOps y Gestión de Costos en Azure 9](#_Toc203734840)

[3.4 Modelo de Madurez en la Nube 9](#_Toc203734841)

[4 Estándares de Nomenclatura y Etiquetado de Recursos 10](#_Toc203734842)

[4.1 Convención de Nomenclatura de Recursos 10](#_Toc203734843)

[4.1.1 Estructura Formalizada 11](#_Toc203734844)

[4.1.2 Desglose de Componentes 11](#_Toc203734845)

[4.1.3 Manejo de Excepciones y Restricciones 11](#_Toc203734846)

[4.1.4 Tabla Maestra de Acrónimos de Servicios de Azure 11](#_Toc203734847)

[4.2 Estrategia de Etiquetado (Tagging) para Gobernanza 14](#_Toc203734848)

[4.2.1 Etiquetas Mandatorias 14](#_Toc203734849)

[4.2.2 Etiquetas Opcionales (Recomendadas) 14](#_Toc203734850)

[5 Automatización de Infraestructura con Terraform y Azure DevOps 14](#_Toc203734851)

[5.1 Adopción de un Enfoque de Infraestructura como Código (IaC) 15](#_Toc203734852)

[5.2 Estructura de Proyectos y Módulos en Terraform 15](#_Toc203734853)

[5.2.1 Estructura de Repositorio y Directorios 15](#_Toc203734854)

[5.2.2 Uso de Módulos Reutilizables 16](#_Toc203734855)

[5.3 Gestión del Estado Remoto de Terraform (Remote State) 16](#_Toc203734856)

[5.4 Arquitectura de Pipelines CI/CD para Infraestructura en Azure DevOps 16](#_Toc203734857)

[5.5 Gestión Segura de Secretos y Credenciales en el Pipeline 17](#_Toc203734858)

[6 Planos Arquitectónicos y Guías de Servicio 18](#_Toc203734859)

[6.1 Arquitectura de Red de Referencia: Topología Hub-Spoke 18](#_Toc203734860)

[6.2 Guía de Buenas Prácticas para Servicios de Datos y PaaS 18](#_Toc203734861)

[6.2.1 Aislamiento de Red con Private Endpoints 18](#_Toc203734862)

[6.2.2 Gestión de Secretos con Azure Key Vault 19](#_Toc203734863)

[6.2.3 Integración con Microsoft Purview 19](#_Toc203734864)

[7 Conclusiones y Próximos Pasos 19](#_Toc203734865)

[8 Conclusiones 20](#_Toc203734866)

[9 REFERENCIAS 20](#_Toc203734867)

Índice de tablas

[Tabla 1: Tabla maestra acronimos 14](#_Toc203734927)

Índice de Ilustraciones

[Ilustración 1: Pilares gobernanza 6](#_Toc203734923)

[Ilustración 2: Modelo de Madurez 10](#_Toc203734924)

Histórico de versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha** | **Estado** | **Responsable** | **Descripción del Cambio** |
| 1.0 | 18/07/2024 | Borrador | E2E -Miguelangel Garcia | Primera entrega |

Marco de Referencia de Buenas Prácticas y Estándares en la Nube para Microsoft Azure

# Introducción

Este documento establece el marco de referencia oficial para la adopción, gobernanza y operación de Microsoft Azure en la organización. Su propósito es estandarizar las prácticas en torno a la arquitectura, seguridad, automatización y gestión de costos para garantizar que todas las soluciones en la nube sean seguras, escalables, resilientes y costo-eficientes. Las directrices aquí contenidas son de obligado cumplimiento para todos los nuevos proyectos, comenzando con la implementación del proyecto "Wiener APIM" , y servirán como la base para la migración y modernización de cargas de trabajo existentes.

El marco se fundamenta en cuatro pilares clave:

Ilustración 1: Pilares gobernanza

La adhesión a estos principios no es opcional; es un requisito para asegurar el éxito a largo plazo y la sostenibilidad de las operaciones en la nube.

# Marco Fundacional de Gobernanza y Operaciones en la Nube

Esta sección establece los principios no negociables que rigen la gestión del entorno de Azure, asegurando el control, la seguridad y la eficiencia financiera desde el inicio. La transición de un modelo de TI reactivo a uno de gobernanza proactiva y centralizada es fundamental para la creación de un Cloud Center of Excellence (CCoE), y este documento constituye su manifiesto fundacional.

## Principios del Modelo Operativo en la Nube: Adopción del Azure Well-Architected Framework

Todas las decisiones de diseño, implementación y operación en Azure se regirán por los principios del Microsoft Azure Well-Architected Framework (WAF). Este marco proporciona un conjunto de principios rectores que se utilizarán para evaluar y optimizar cada carga de trabajo. La aplicación de estos principios es un requisito continuo durante todo el ciclo de vida de la aplicación.

Los cinco pilares del WAF y su aplicación práctica en la organización son:

## Estrategia de Gobernanza y Cumplimiento

Una estructura de gobernanza sólida es la base para mantener el control, la seguridad y el cumplimiento normativo en el entorno de Azure.

### Organización de Recursos

Se establece una jerarquía estricta y estandarizada para la organización de todos los recursos de Azure, utilizando Grupos de Administración, Suscripciones y Grupos de Recursos. Esta estructura es fundamental para la aplicación de políticas, la gestión del acceso y la segmentación de costos.

* **Grupos de Administración (Management Groups):** Se utilizarán para agrupar suscripciones que comparten requisitos comunes de políticas y control de acceso. Se creará una jerarquía que separe los entornos de producción de los de no producción, y que pueda segmentar por unidades de negocio o "Landing Zones".
* **Suscripciones (Subscriptions):** Actuarán como una unidad de gestión, facturación y escala. Se utilizarán suscripciones dedicadas para servicios de plataforma compartidos (como la red Hub) y suscripciones separadas para cada aplicación o línea de negocio principal.
* **Grupos de Recursos (Resource Groups):** Serán el contenedor para todos los recursos de Azure que comparten el mismo ciclo de vida. Todos los recursos de una aplicación específica, como el proyecto Wiener APIM, residirán en el mismo grupo de recursos, alineado con la convención de nomenclatura de la aplicación.

### Control de Acceso Basado en Roles (RBAC)

El principio de mínimo privilegio es mandatorio y se aplicará en todos los niveles. El acceso a los recursos de Azure se otorgará únicamente a través de roles RBAC que proporcionen los permisos estrictamente necesarios para realizar una función específica.

* **Se prohíbe el uso de roles genéricos y de altos privilegios** como "Propietario" (Owner) o "Colaborador" (Contributor) a nivel de suscripción o grupo de administración para usuarios o principales de servicio de automatización.
* **Se crearán y asignarán roles personalizados para tareas específicas**. Por ejemplo, el principal de servicio utilizado por los pipelines de Terraform tendrá un rol personalizado que le permita gestionar recursos, pero no asignar roles a otros principales, limitando así el radio de impacto en caso de una posible brecha de seguridad.

### Imposición de Estándares con Azure Policy

Azure Policy será la herramienta principal para la gobernanza proactiva y automatizada. Se utilizará no solo para auditar el cumplimiento, sino fundamentalmente para *imponer* los estándares definidos en este documento.

* Se implementarán políticas con el efecto modify para añadir automáticamente etiquetas mandatorias a los recursos durante su creación si estas faltan.
* Se crearán políticas con el efecto deny para bloquear el despliegue de cualquier recurso que no cumpla con las convenciones de nomenclatura o que carezca de etiquetas críticas, asegurando que ningún recurso no conforme pueda ser creado en el entorno.

## Principios de FinOps y Gestión de Costos en Azure

La gestión proactiva de costos es un pilar fundamental de las operaciones en la nube. La falta de una estrategia formal de FinOps representa un riesgo financiero significativo. La política de etiquetado, impuesta por Azure Policy, es la herramienta técnica que habilita la visibilidad, la atribución y, en última instancia, la responsabilidad financiera.

* **Visibilidad y Atribución de Costos:** La estrategia de etiquetado es la piedra angular para la visibilidad de costos. Etiquetas como PROJECT\_CODE (que se mapeará a un centro de costos), BUSINESS\_OWNER y APID permitirán una atribución precisa de los costos a través de los informes y análisis de Azure Cost Management. Esto permite responder a la pregunta fundamental: **¿quién está gastando qué y por qué?.**
* **Estrategias de Optimización:**
* **Monitorización Proactiva:** Se utilizará Azure Advisor de forma regular para identificar recursos infrautilizados, como máquinas virtuales o bases de datos sobredimensionadas, y aplicar recomendaciones de "right-sizing" para alinear el costo con el uso real.
* **Compromisos de Gasto:** Para cargas de trabajo con un uso predecible y constante, como las máquinas virtuales del proyecto Wiener o el nivel de servicio de APIM, se evaluará proactivamente el uso de Instancias Reservadas de Azure y Planes de Ahorro para obtener descuentos significativos en comparación con el modelo de pago por uso.
* **Creación de Presupuestos y Alertas:** Se configurarán presupuestos en Azure Cost Management para cada suscripción y grupo de recursos crítico. Se establecerán alertas automáticas que notificarán a los responsables designados en las etiquetas (PROJECT\_MANAGER y BUSINESS\_OWNER) cuando el gasto se acerque a los umbrales predefinidos, permitiendo una intervención temprana antes de que se produzcan desviaciones presupuestarias.

## Modelo de Madurez en la Nube

Para guiar la evolución continua de las capacidades en la nube de la organización, se adopta un Modelo de Madurez en la Nube (Cloud Maturity Model - CMM). Este modelo proporciona un marco para evaluar el estado actual de las prácticas en la nube e identificar los pasos necesarios para avanzar hacia niveles superiores de eficiencia, innovación y gobernanza.

Se definen los siguientes niveles de madurez:

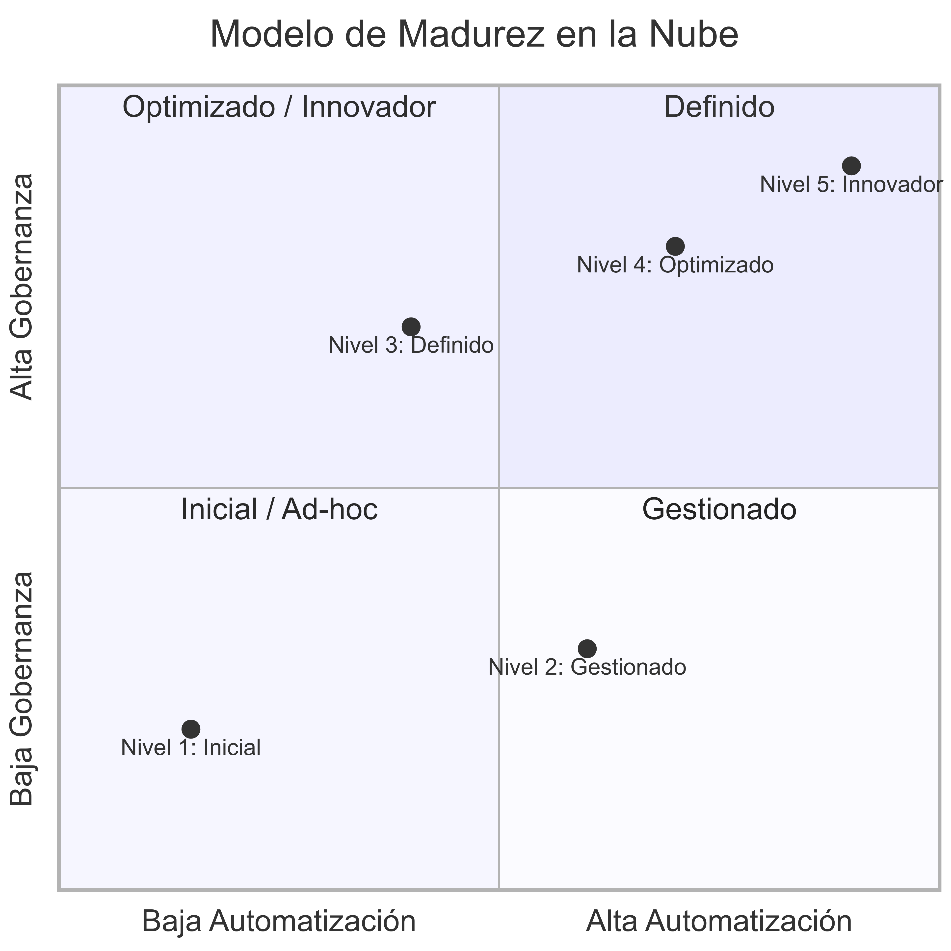
1. **Nivel 1 (Inicial/Ad-hoc):** Despliegues manuales, falta de estándares, gobernanza reactiva. La organización se encuentra actualmente en esta fase, como lo demuestra la necesidad de este documento.
2. **Nivel 2 (Gestionado):** Se establecen estándares básicos de nomenclatura y etiquetado. Se introduce la automatización a través de IaC. Este documento marca el paso fundamental a este nivel.
3. **Nivel 3 (Definido):** Las prácticas de IaC y CI/CD están estandarizadas y son obligatorias. La gobernanza es proactiva a través de Azure Policy. Los principios de FinOps se aplican de forma consistente.
4. **Nivel 4 (Optimizado):** Los procesos están altamente automatizados, incluyendo la remediación de problemas de seguridad y cumplimiento. Se utilizan análisis avanzados para optimizar continuamente los costos y el rendimiento.
5. **Nivel 5 (Innovador):** La plataforma en la nube actúa como un habilitador de la innovación empresarial. Se adoptan prácticas de Platform Engineering para proporcionar a los desarrolladores plataformas de autoservicio seguras y eficientes.
6. 

Ilustración 2: Modelo de Madurez

Este documento es el catalizador para la transición del Nivel 1 al Nivel 3, sentando las bases para futuras optimizaciones.

# Estándares de Nomenclatura y Etiquetado de Recursos

Esta sección formaliza las convenciones exactas que deben seguirse para nombrar y etiquetar cada recurso en Azure. La estandarización rigurosa no es una cuestión de estética, sino un requisito funcional que habilita la automatización, la gobernanza y la gestión a escala. Un nombre de recurso bien estructurado es una forma de metadato que puede ser fácilmente analizado por scripts y políticas, simplificando la gestión programática del entorno.

## Convención de Nomenclatura de Recursos

### Estructura Formalizada

Basándose en el análisis de los requerimientos para el proyecto Wiener , se establece la siguiente estructura de nomenclatura como el estándar obligatorio para todos los recursos de Azure:

**{nube}-{empresa}-{aplicacion}-{ambiente}-{tipo\_servicio}-{descriptor}**

### Desglose de Componentes

* **{nube}:** Prefijo de 2 letras para el proveedor de nube. Será az para Microsoft Azure.
* **{empresa}:** Código de 3 letras para la entidad de negocio o unidad organizativa. Para el proyecto inicial, será uwr.
* **{aplicacion}:** Código de hasta 6 letras que identifica de forma única la aplicación, proyecto o carga de trabajo. Para el proyecto inicial, es apim.
* **{ambiente}:** Código de 3 letras para el entorno de despliegue. Los valores estandarizados son dev (Desarrollo), qas (Calidad), prd (Producción) y snb (Sandbox).
* **{tipo\_servicio}:** Acrónimo oficial del tipo de recurso de Azure. Este componente es crítico y debe obtenerse de la Tabla Maestra de Acrónimos de Servicios de Azure (ver más abajo).
* **{descriptor}:** Un nombre descriptivo o un número secuencial de dos dígitos (ej. principal, dwh, 01) que diferencia instancias del mismo tipo dentro del mismo contexto.

**Ejemplo de aplicación:** Una Virtual Network para la aplicación APIM de la empresa UWR en el entorno de desarrollo se nombraría: az-uwr-apim-dev-vnet-principal.

### Manejo de Excepciones y Restricciones

Un estándar robusto debe reconocer y adaptarse a las limitaciones técnicas de la plataforma. Ciertos servicios de Azure tienen restricciones de nomenclatura específicas que impiden la aplicación directa del estándar anterior. Para estos casos, se utilizarán las siguientes convenciones adaptadas, que serán validadas por políticas de Azure específicas para cada tipo de recurso.

* **Azure Key Vault (kv):** Máximo 24 caracteres, alfanuméricos, sin guiones. La convención se adapta a un formato concatenado y en minúsculas.
* Ejemplo: azuwrapimdevkv01
* **Azure Storage Account / Data Lake Storage (st, dls):** 3 a 24 caracteres, solo letras minúsculas y números. La convención se adapta a un formato concatenado.
* Ejemplo: azuwrapimdevdls01
* **Azure Synapse Workspace (synw):** No permite guiones ni mayúsculas.
* Ejemplo: az-uwr-apim-dev-synw-dwh (el estándar ya cumple con la restricción de mayúsculas).

### Tabla Maestra de Acrónimos de Servicios de Azure

La siguiente tabla establece la "fuente de la verdad" para el componente {tipo\_servicio}. Su uso es obligatorio para garantizar la consistencia en toda la organización. Esta tabla se basa en la lista inicial y ha sido estandarizada y curada.

| Categoría | Nombre del Servicio | Acrónimo | Notas de Nomenclatura |
| --- | --- | --- | --- |
| Compute | Virtual Machine | vmch |  |
| Compute | Virtual Machine Scale Set | vmss |  |
| Compute | App Service (Web App) | aswa |  |
| Compute | App Service Plan | aspl |  |
| Compute | Azure Functions | func |  |
| Compute | Azure Spring Apps | aspa |  |
| Compute | Azure Batch | btch |  |
| Compute | Azure Dedicated Host | adho |  |
| Compute | Azure VMware Solution | avms |  |
| Compute | Azure CycleCloud | cycl |  |
| Containers | Azure Kubernetes Service (AKS) | aksv |  |
| Containers | Azure Container Apps | acap |  |
| Containers | Azure Container Instances | acin |  |
| Containers | Azure Container Registry | acrg |  |
| Containers | Service Fabric Cluster | sfcl |  |
| Networking | Virtual Network | vnet |  |
| Networking | Subnet | snet |  |
| Networking | Network Interface | nint |  |
| Networking | Public IP Address | pipa |  |
| Networking | Network Security Group | nsgp |  |
| Networking | Application Security Group | asgp |  |
| Networking | Azure Firewall | afwl |  |
| Networking | Azure Firewall Policy | afwp |  |
| Networking | Load Balancer | ldbl |  |
| Networking | Application Gateway | apgw |  |
| Networking | VPN Gateway | vpng |  |
| Networking | Local Network Gateway | lngw |  |
| Networking | Virtual Network Gateway | vngw |  |
| Networking | Route Table | rtbl |  |
| Networking | Azure DNS (Public Zone) | dnsz |  |
| Networking | Azure DNS (Private Zone) | pdns |  |
| Networking | Private Endpoint | pept |  |
| Networking | Private Link Service | plks |  |
| Networking | Azure Bastion | bast |  |
| Networking | Web Application Firewall (WAF) | wafa |  |
| Networking | Azure Front Door | afdo |  |
| Networking | Traffic Manager Profile | trmg |  |
| Networking | ExpressRoute Circuit | exrc |  |
| Networking | Network Watcher | netw |  |
| Networking | Virtual WAN | vwan |  |
| Storage | Storage Account | stor | 3-24 caracteres, solo letras minúsculas y números. |
| Storage | Data Lake Storage Gen2 | dlsg | 3-24 caracteres, solo letras minúsculas y números. |
| Storage | Azure Files | afls |  |
| Storage | Azure Blob Storage | blst | Parte de una cuenta de almacenamiento. |
| Storage | Azure NetApp Files | anfs |  |
| Storage | Azure HPC Cache | hpcc |  |
| Storage | Azure StorSimple | stsp |  |
| Databases | Azure SQL Database | sqld |  |
| Databases | Azure SQL Server (Logical) | sqls |  |
| Databases | Azure SQL Managed Instance | sqlm |  |
| Databases | Azure Database for PostgreSQL | psql |  |
| Databases | Azure Database for MySQL | msql |  |
| Databases | Azure Database for MariaDB | madb |  |
| Databases | Azure Cosmos DB Account | csdb |  |
| Databases | Azure Cache for Redis | reds |  |
| Databases | SQL Server on Virtual Machines | sqlv |  |
| Databases | Azure Database Migration Service | dmsv |  |
| Analytics | Azure Synapse Analytics | syna | No permite guiones ni mayúsculas. |
| Analytics | Azure Databricks | dbrk |  |
| Analytics | Azure Data Factory | adfa |  |
| Analytics | Azure Stream Analytics | stra |  |
| Analytics | Azure Data Explorer | adex |  |
| Analytics | Power BI | pwbi |  |
| Analytics | Azure Analysis Services | anas |  |
| Analytics | HDInsight | hdin |  |
| Analytics | Data Lake Analytics | dlan |  |
| AI + ML | Azure Machine Learning | amls |  |
| AI + ML | Azure AI Services | aiss | Anteriormente Cognitive Services. |
| AI + ML | Azure OpenAI Service | aoai |  |
| AI + ML | Azure AI Search | aisr | Anteriormente Cognitive Search. |
| AI + ML | Azure Bot Service | bots |  |
| AI + ML | Azure AI Vision | visn |  |
| AI + ML | Azure AI Speech | spch |  |
| AI + ML | Azure AI Language | lang |  |
| AI + ML | Azure Document Intelligence | doci |  |
| IoT | IoT Hub | ioth |  |
| IoT | IoT Central | iotc |  |
| IoT | Azure Digital Twins | adtw |  |
| IoT | IoT Edge | iote |  |
| IoT | Time Series Insights | tsin |  |
| Management | Resource Group | rsgp |  |
| Management | Azure Monitor | moni |  |
| Management | Log Analytics Workspace | logs |  |
| Management | Azure Policy | plcy |  |
| Management | Azure Blueprints | blpr |  |
| Management | Azure Automation Account | auto |  |
| Management | Azure Advisor | advi |  |
| Management | Azure Lighthouse | lths |  |
| Management | Azure Resource Mover | rsmv |  |
| Management | Azure Service Health | shlt |  |
| Management | Azure Cost Management | cstm |  |
| Security | Microsoft Defender for Cloud | defc |  |
| Security | Microsoft Sentinel | sent |  |
| Security | Key Vault | kvlt | 3-24 caracteres, alfanuméricos, sin guiones. |
| Security | Azure Active Directory | aads |  |
| Security | Managed Identity | mgid |  |
| Security | Azure DDoS Protection Plan | ddos |  |
| Security | Azure Information Protection | aipr |  |
| Security | Azure Dedicated HSM | dhsm |  |
| Integration | API Management | apim |  |
| Integration | Logic Apps | lapp |  |
| Integration | Service Bus | svcb |  |
| Integration | Event Grid | evgd |  |
| Integration | Event Hub Namespace | evhn |  |
| DevOps | Azure DevOps | devs |  |
| DevOps | Azure DevTest Labs | dtlb |  |
| DevOps | Azure Load Testing | ldts |  |
| Migration | Azure Migrate | migr |  |
| Migration | Azure Site Recovery | asrc |  |
| Migration | Azure Database Migration Service | dmsv |  |
| Mixed Reality | Azure Spatial Anchors | span |  |
| Mixed Reality | Azure Remote Rendering | rren |  |
| Communication | Azure Communication Services | coms |  |
| Web | Azure SignalR Service | sigr |  |
| Web | Azure Web PubSub | wpsb |  |
| Web | Azure Static Web Apps | swpa |  |

Tabla 1: Tabla maestra acronimos

En caso de no existir el servicio se debe notificar al área de infraestructura para su generación y actualización en el estándar.

## Estrategia de Etiquetado (Tagging) para Gobernanza

El etiquetado es el mecanismo principal para añadir metadatos contextuales a los recursos de Azure. Esta política, formalizada a partir de los estándares del proyecto, es fundamental para la gestión de costos, la automatización y la gobernanza.

### Etiquetas Mandatorias

Las siguientes etiquetas deben estar presentes en todos los recursos que las soporten. Su ausencia resultará en el bloqueo del despliegue a través de Azure Policy.

* **coid** (Código de Empresa): Identificador de la entidad legal (ej. WNR).
* **apid** (ID de Aplicación): Identificador único de la aplicación o carga de trabajo (ej. APIMWIENER).
* **env** (Entorno): El ambiente del recurso (dev, qas, prd, snb).
* **business\_owner** (Dueño de Negocio): El nombre del area responsable funcional o de negocio del recurso (ej. Tecnologias de la Información).
* **project\_code** (Código de Proyecto): El código de proyecto utilizado para la atribución de costos y la facturación interna (ej. PY-2507-01).

### Etiquetas Opcionales (Recomendadas)

Aunque no son obligatorias, se recomienda encarecidamente el uso de estas etiquetas para enriquecer el contexto de los recursos.

* **project\_manager** (Gerente de Proyecto): El nombre del responsable de la ejecución del proyecto (ej. Luciana).
* **country** (País): El país donde reside la empresa o se despliega el recurso (ej. Peru).

# Automatización de Infraestructura con Terraform y Azure DevOps

Esta sección establece el marco técnico para el aprovisionamiento y la gestión de la infraestructura en Azure. Se decreta que **Terraform** como herramienta de Infraestructura como Código (IaC) y **Azure DevOps** como plataforma de CI/CD son la única vía soportada para el despliegue de recursos. Queda prohibido el aprovisionamiento manual de recursos en entornos de QAS y Producción a través del portal de Azure. Salvo excepciones técnicamente fundamentadas.

Para despliegues manuales se debe presentar un informe técnico fundamentando la imposibilidad del uso de IaC.

## Adopción de un Enfoque de Infraestructura como Código (IaC)

Toda la infraestructura de Azure se definirá en archivos de código declarativo (lenguaje HCL de Terraform). Este código será almacenado y versionado en un repositorio Git, y los cambios se aplicarán exclusivamente a través de pipelines de despliegue automatizados. Este enfoque garantiza que la infraestructura sea repetible, auditable, consistente y que se pueda someter a los mismos procesos de revisión de código y pruebas que el software de aplicación, reduciendo drásticamente el error humano y la deriva de configuración.

## Estructura de Proyectos y Módulos en Terraform

Una estructura de proyecto bien definida es crucial para gestionar la complejidad y promover la reutilización.

### Estructura de Repositorio y Directorios

Se recomienda una estructura de monorepo por aplicación o unidad de negocio. Dentro de cada repositorio, se utilizará una estructura de directorios estandarizada para aislar los entornos y centralizar los módulos reutilizables :

/terraform  
├── modules/  
│   ├── networking/      # Módulo para VNet, subnets, NSGs  
│   ├── apim/            # Módulo para una instancia de APIM estandarizada  
│   └──...  
├── environments/  
│   ├── dev/  
│   │   ├── main.tf  
│   │   ├── variables.tf  
│   │   └── dev.tfvars  
│   ├── qas/  
│   │   ├── main.tf  
│   │   ├── variables.tf  
│   │   └── qas.tfvars  
│   └── prd/  
│       ├── main.tf  
│       ├── variables.tf  
│       └── prd.tfvars  
└── main.tf              # Configuración global (providers, backend)

Esta estructura es una estrategia fundamental para gestionar el "radio de impacto" (blast radius). Al separar los entornos en directorios, cada uno con su propio estado, un error en el entorno de desarrollo (dev) no puede afectar de ninguna manera al entorno de producción (prd), ya que sus estados están completamente aislados.

### Uso de Módulos Reutilizables

Se mandata la creación y el uso de módulos de Terraform para componentes de infraestructura comunes (ej. una VNet con subredes y NSGs preconfigurados, una instancia de APIM con políticas de seguridad base, un Key Vault con la configuración de seguridad estándar). Los módulos encapsulan la complejidad, reducen la duplicación de código y aseguran que todos los recursos se desplieguen siguiendo las mejores prácticas y los estándares de la organización.

## Gestión del Estado Remoto de Terraform (Remote State)

El archivo de estado de Terraform (terraform.tfstate), que mapea los recursos del código a la infraestructura real, es un componente crítico y sensible.

* **Backend en Azure Blob Storage:** Se prohíbe el uso de archivos de estado locales. El estado debe almacenarse de forma remota y centralizada en un contenedor de Azure Blob Storage. Este contenedor residirá en una cuenta de almacenamiento dedicada y securizada, creada específicamente para este propósito.
* **Aislamiento de Estado por Entorno:** Cada entorno (dev, qas, prd) debe tener su propio archivo de estado (.tfstate) dentro del contenedor de Blob Storage. Esto se logra parametrizando la clave (key) en la configuración del backend.
* **Seguridad del Estado:**
* **Bloqueo de Estado (State Locking):** La cuenta de almacenamiento se configurará para utilizar el mecanismo de bloqueo nativo de Azure Blob Storage, que previene operaciones concurrentes que podrían corromper el archivo de estado.
* **Cifrado en Reposo:** El cifrado del lado del servicio (SSE) con claves gestionadas por Microsoft estará habilitado por defecto en la cuenta de almacenamiento, protegiendo el estado en reposo.
* **Acceso Restringido:** El acceso público a la cuenta de almacenamiento estará deshabilitado. El acceso se restringirá mediante RBAC, otorgando permisos únicamente al principal de servicio utilizado por Azure DevOps.

## Arquitectura de Pipelines CI/CD para Infraestructura en Azure DevOps

Se define una plantilla de pipeline YAML multi-etapa como el estándar para todos los proyectos de infraestructura. Este pipeline implementa un flujo de trabajo de "GitOps", donde las operaciones sobre la infraestructura son el resultado de acciones en el repositorio Git.

El pipeline constará de las siguientes etapas:

1. **Etapa de Validación (Disparador: Pull Request a la rama main):**
   * Ejecuta terraform init -backend=false para inicializar sin configurar el backend.
   * Ejecuta terraform validate para comprobar la sintaxis del código HCL.
   * Ejecuta terraform fmt -check para asegurar un formato de código consistente.
2. **Etapa de Planificación (Disparador: Merge a main):**
   * Esta etapa despliega automáticamente al entorno de Desarrollo (DEV).
   * Ejecuta terraform init para configurar el backend remoto del entorno de DEV.
   * Ejecuta terraform plan -out=tfplan para generar un plan de ejecución detallado. El plan muestra exactamente qué recursos se crearán, modificarán o eliminarán.
   * El plan se publica como un artefacto del pipeline y se requiere una aprobación manual de un revisor designado para proceder.
3. **Etapa de Aplicación (Depende de la aprobación del Plan):**
   * Descarga el artefacto tfplan.
   * Ejecuta terraform apply "tfplan" para aplicar los cambios en el entorno de DEV.
4. **Etapas de Promoción a QAS y PRD:**
   * Estas etapas se configuran con disparadores manuales. La promoción a entornos superiores no es automática.
   * Replican el flujo de Planificación y Aplicación, utilizando los archivos .tfvars y los estados remotos correspondientes a cada entorno.
   * Requieren aprobaciones formales por parte de los responsables del entorno de QAS y Producción, respectivamente, antes de que cualquier cambio pueda ser aplicado.

## Gestión Segura de Secretos y Credenciales en el Pipeline

La seguridad de las credenciales utilizadas para la automatización es de máxima prioridad.

* **Principal de Servicio de Mínimo Privilegio:** Se creará un Principal de Servicio (SPN) de Azure Active Directory para cada aplicación o entorno. A este SPN se le asignará un rol RBAC personalizado que le otorgue únicamente los permisos necesarios para gestionar los recursos definidos en el código de Terraform. No se le otorgarán permisos de Propietario (Owner).
* **Integración con Azure Key Vault:** Las credenciales del SPN (Client ID, Client Secret, Tenant ID) y cualquier otro secreto (ej. claves de API para el backend) **no deben almacenarse nunca como variables en texto plano o secretos en Azure DevOps**. En su lugar, se almacenarán de forma segura en un Azure Key Vault. El pipeline de Azure DevOps se configurará para acceder a estos secretos en tiempo de ejecución a través de un "Variable Group" vinculado al Key Vault, utilizando una conexión de servicio con permisos de mínimo privilegio sobre el mismo.

Este sistema completo, que combina un pipeline multi-etapa con validaciones, separación de estados, aprobaciones manuales y gestión de secretos just-in-time, constituye un modelo de "Despliegue de Confianza Cero" (Zero Trust Deployment). La confianza no se asume en ninguna etapa; se verifica a través de controles técnicos y de proceso, reduciendo drásticamente el riesgo de despliegues no autorizados, accidentales o maliciosos.

# Planos Arquitectónicos y Guías de Servicio

Esta sección traduce los principios y estándares en arquitecturas de referencia concretas y guías de implementación para los servicios clave utilizados en el proyecto Wiener y que serán comunes en futuros proyectos.

## Arquitectura de Red de Referencia: Topología Hub-Spoke

Se establece la topología de red Hub-Spoke como el estándar para toda la organización en Azure. Este modelo no es solo un patrón de red, sino un modelo de gobierno de seguridad que centraliza el control, la inspección y la visibilidad del tráfico.

* **Diseño Conceptual:**
* **Hub VNet:** Se creará una Virtual Network (VNet) central que alojará servicios compartidos de red y seguridad. Será el único punto de conectividad con redes on-premise (a través de VPN Gateway o ExpressRoute) y el punto de control para todo el tráfico de entrada y salida de internet.
* **Spoke VNets:** Se crearán VNets separadas para cada aplicación o carga de trabajo (como la VNet para el proyecto APIM). Las Spokes se conectarán al Hub mediante VNet Peering. El tráfico entre diferentes Spokes deberá transitar a través del Hub, prohibiendo el peering directo entre Spokes para mantener el control centralizado.
* **Resolución de Nombres con Zonas DNS Privadas:**
* Para la resolución de nombres de servicios PaaS accedidos a través de Private Endpoints (ej. storageaccount.privatelink.blob.core.windows.net), se crearán Zonas DNS Privadas en Azure.
* Estas zonas se vincularán a la Hub VNet. Las VNets de las Spokes se configurarán para utilizar los servidores DNS del Hub (o servidores DNS personalizados en el Hub que reenvíen a Azure DNS), asegurando una resolución de nombres consistente en toda la topología.

## Guía de Buenas Prácticas para Servicios de Datos y PaaS

La seguridad de los datos es primordial. Los siguientes estándares se aplican a todos los servicios PaaS que almacenan o procesan datos.

### Aislamiento de Red con Private Endpoints

Este es un cambio de paradigma fundamental en la seguridad en la nube, moviéndose hacia una arquitectura de "confianza cero" (Zero Trust).

* Se establece como política mandatoria que todos los servicios PaaS que lo soporten (incluyendo Azure SQL Database, Azure Data Lake Storage, Azure Key Vault, Azure Synapse Analytics, etc.) deben tener el **acceso desde redes públicas deshabilitado**.
* El acceso a estos servicios se realizará exclusivamente a través de **Private Endpoints** desplegados en las VNets de las aplicaciones (Spokes). Un Private Endpoint proyecta el servicio PaaS dentro de la red privada, asignándole una dirección IP del rango de la VNet. El tráfico hacia el servicio nunca abandona la red troncal de Microsoft, eliminando por completo la exposición a la internet pública.

### Gestión de Secretos con Azure Key Vault

* Azure Key Vault es el **único repositorio aprobado** para almacenar secretos de aplicación, como cadenas de conexión, claves de API y certificados. Los secretos no deben almacenarse en el código fuente, archivos de configuración o variables de pipeline.
* El acceso a los secretos en Key Vault se controlará mediante el modelo de permisos RBAC y Access Policies, aplicando el principio de mínimo privilegio. Las aplicaciones y servicios de Azure deben utilizar **Identidades Gestionadas (Managed Identities)** para autenticarse en Key Vault, eliminando la necesidad de gestionar credenciales.
* Se mandata la habilitación de **Microsoft Defender for Key Vault** para detectar patrones de acceso anómalos o intentos de explotación, proporcionando una capa adicional de inteligencia de amenazas.

### Integración con Microsoft Purview

Para proyectos con un alto componente de datos y requisitos de gobernanza, como el proyecto Wiener, se recomienda la integración con **Microsoft Purview**. Purview permite escanear, catalogar y clasificar los activos de datos en servicios como Data Lake Storage y Synapse Analytics. Esto proporciona capacidades críticas para el seguimiento del linaje de datos (data lineage), la creación de un glosario de negocio y la aplicación de políticas de gobierno de datos a escala.

# Conclusiones

Este documento establece un marco de referencia integral y prescriptivo para la adopción y operación de Microsoft Azure. Su implementación representa un paso decisivo en la madurez de la nube de la organización, pasando de un enfoque ad-hoc a un modelo gobernado, automatizado y seguro.

Las directrices aquí presentadas no son meras recomendaciones, sino estándares de obligado cumplimiento que serán aplicados y validados a través de herramientas de automatización y políticas de gobernanza. La adopción de Infraestructura como Código con Terraform y pipelines CI/CD en Azure DevOps es el pilar operativo que garantizará la consistencia y repetibilidad de los despliegues. Las arquitecturas de referencia, como la topología Hub-Spoke y el uso mandatorio de Private Endpoints, establecen una base de seguridad robusta alineada con los principios de Zero Trust.

El éxito de esta iniciativa depende del compromiso de todos los equipos técnicos y de negocio para adoptar y adherirse a estos estándares. El Cloud Center of Excellence (CCoE) será responsable de mantener y evolucionar este marco, así como de proporcionar la formación y el soporte necesarios para su implementación en todos los proyectos, comenzando de inmediato con el proyecto Wiener APIM.

# REFERENCIAS

1. Azure Well-Architected Framework - Learn Microsoft, fecha de acceso: julio 18, 2025, <https://learn.microsoft.com/en-us/azure/well-architected/>
2. well-architected/well-architected/service-guides/azure-api-management.md at main - GitHub, fecha de acceso: julio 18, 2025, <https://github.com/MicrosoftDocs/well-architected/blob/main/well-architected/service-guides/azure-api-management.md>
3. Azure Cost Management - Best Practices & Tools - Spacelift, fecha de acceso: julio 18, 2025, <https://spacelift.io/blog/azure-cost-management>
4. Intelligent FinOps in Azure - Mechanics Team - Medium, fecha de acceso: julio 18, 2025, <https://officegarageitpro.medium.com/intelligent-finops-in-azure-c603cf5e4bac>
5. Tutorial: Manage tag governance with Azure Policy - Learn Microsoft, fecha de acceso: julio 18, 2025, <https://learn.microsoft.com/en-us/azure/governance/policy/tutorials/govern-tags>
6. Enforce tag usage on Azure resources using Tag policies - Sander van de Velde, fecha de acceso: julio 18, 2025, <https://sandervandevelde.wordpress.com/2022/03/19/enforce-tag-usage-on-azure-resources-using-tag-policies/>
7. Define your naming convention - Cloud Adoption Framework - Learn Microsoft, fecha de acceso: julio 18, 2025, <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming>
8. How to use Azure Policy to enforce resource naming conventions in your DevOps pipelines, fecha de acceso: julio 18, 2025, <https://www.m365princess.com/blogs/devops-policy/>
9. Cloud Cost Optimization | Microsoft Azure, fecha de acceso: julio 18, 2025, <https://azure.microsoft.com/en-us/solutions/cost-optimization>
10. Terraform with Azure DevOps CI/CD Pipelines - Tutorial - Spacelift, fecha de acceso: julio 18, 2025, <https://spacelift.io/blog/terraform-azure-devops>
11. Terraform Best Practices For CI/CD Pipelines - Terrateam, fecha de acceso: julio 18, 2025, <https://terrateam.io/blog/terraform-best-practices-ci-cd>
12. Terraform Files and Folder Structure | Organizing Infrastructure-as-Code - Env0, fecha de acceso: julio 18, 2025, <https://www.env0.com/blog/terraform-files-and-folder-structure-organizing-infrastructure-as-code>
13. Hub-spoke network topology in Azure - Azure Architecture Center - Learn Microsoft, fecha de acceso: julio 18, 2025, <https://learn.microsoft.com/en-us/azure/architecture/networking/architecture/hub-spoke>