

SALTA, 10-MAR-2026

## **RESOLUCIÓN Nº 148**

### **UNIVERSIDAD PROVINCIAL DE LA ADMINISTRACIÓN, TECNOLOGÍA Y OFICIOS**

Expediente SICAD Nº 259/26

**VISTO** el Artículo 75, apartado 19, de la Constitución Nacional, la Ley nacional 24521 y la Ley 8312, modificatoria de su similar 7803, y

#### **CONSIDERANDO:**

Que, de conformidad con lo dispuesto por el Artículo 26 de la Ley 24521, la enseñanza superior universitaria estará a cargo de las Universidades Nacionales, de las Universidades Provinciales y Privadas, y de los Institutos Universitarios;

Que mediante Ley 8312, modificatoria de su similar 7803, se crea la Universidad Provincial de la Administración, Tecnología y Oficios (UPATecO) como persona jurídica pública, con autonomía institucional y académica, y autarquía financiera y administrativa;

Que, por su parte, el Artículo 27 de la Ley 7803, modificado por Ley 8312, establece que los títulos que emita la UPATecO tendrán validez en todo el territorio de la provincia de Salta y habilitarán a los egresados a ejercer su profesión u oficio, conforme la normativa vigente en la materia;

Que, en las actuaciones de referencia, la Secretaría Académica de esta Universidad eleva el Proyecto de Plan de Estudios correspondiente a la carrera de pregrado "Tecnatura Universitaria en Gestión de Infraestructura *Cloud* y *DevOps*";

Que dicha carrera se organiza en base al Modelo de Formación Modular y Certificación por Competencias, con módulos que conforman unidades académicas mínimas;

Que la carrera mencionada tiene como objetivos, entre otros, la formación de técnicos universitarios con una sólida base tecnológica, metodológica y práctica, capaces de diseñar, implementar, automatizar y administrar infraestructuras digitales en entornos *cloud*, aplicando metodologías de integración y entrega continua, garantizando disponibilidad, escalabilidad, seguridad y eficiencia operativa, y contribuyendo al desarrollo tecnológico y productivo en el marco de estándares de calidad, seguridad informática y ética profesional;

Que en autos queda claramente establecida la fundamentación de la necesidad de la carrera, duración, perfil del egresado, requisitos de ingreso, objetivos de la carrera, metodología, evaluación, organización curricular, contenidos mínimos de cada espacio curricular, entre otros ítems;

**.. // RESOLUCIÓN N° 148**

Expediente SICAD N° 259/26

Que la Dirección Jurídica de esta Universidad se expide favorablemente a través de Dictamen N° 159/26, por lo que corresponde el dictado del acto administrativo pertinente, en virtud de la Ley 8312, modificatoria de su similar 7803, y del Decreto N° 88/23;

Por ello;

**EL RECTOR DE LA UNIVERSIDAD PROVINCIAL  
DE LA ADMINISTRACIÓN, TECNOLOGÍA Y OFICIOS**

**R E S U E L V E :**

**ARTÍCULO 1º.-** Aprobar el Plan de Estudios de la carrera de pregrado “Tecnatura Universitaria en Gestión de Infraestructura *Cloud* y *DevOps*”, de la Universidad Provincial de la Administración, Tecnología y Oficios (UPATecO), que como Anexo forma parte de este instrumento legal.

**ARTÍCULO 2º.-** La presente resolución será refrendada por la señora Vicerrectora de esta Universidad.

**ARTÍCULO 3º.-** Comunicar y archivar.



Firmado digitalmente por  
Dra. MARÍA de los D. TALENS  
Vicerrectora UPATecO



Firmado digitalmente por  
Dr. CARLOS MORELLO  
Rector UPATecO

|    |
|----|
| RC |
|    |
| DD |



**UNIVERSIDAD PROVINCIAL DE ADMINISTRACIÓN,  
TECNOLOGÍA Y OFICIOS (UPATecO)**

**AUTORIDADES**

**RECTOR  
Dr. Carlos Morello**

**VICERRECTORA  
Dr. María de los D. Talens**





## PLAN DE ESTUDIO 2026

### IDENTIFICACIÓN DE LA CARRERA

#### 1.- CARACTERÍSTICAS DE LA CARRERA

|                                    |   |
|------------------------------------|---|
| <b>Denominación de la Carrera:</b> | <b>Tecnicatura Universitaria en Gestión de Infraestructura Cloud y DevOps</b> |
| <b>Nivel Académico:</b>            | <b>Tecnicatura Universitaria</b>  |
| <b>Modalidad:</b>                  | <b>Híbrida</b>  |
| <b>Título que Otorga</b>           | <b>Técnico Universitario en Gestión de Infraestructura Cloud y DevOps</b>     |
| <b>Duración de la carrera:</b>     | <b>2 años y medio</b>   |
| <b>Régimen de cursado:</b>         | <b>Cuatrimestral</b>  |
| <b>Carga Horaria Total:</b>        | <b>1616 horas reloj</b>   |





## **2.- FUNDAMENTACIÓN:**

### **2.1.- Contexto y pertinencia de la propuesta**

La acelerada transformación digital de los sectores productivos, financieros, industriales y de servicios ha generado una profunda reconfiguración de los modelos tecnológicos tradicionales. La adopción masiva de infraestructuras basadas en computación en la nube (Cloud Computing), arquitecturas distribuidas, microservicios y metodologías DevOps constituye actualmente un eje estratégico para la competitividad organizacional.

En Argentina y particularmente en las economías regionales, el crecimiento de empresas tecnológicas, fintech, comercio electrónico, servicios digitales, organismos públicos digitalizados y procesos de modernización del Estado ha incrementado la demanda de perfiles técnicos especializados en administración de infraestructuras cloud, automatización de despliegues, contenedorización y gestión de entornos productivos de alta disponibilidad.

La infraestructura tecnológica se ha convertido en un componente crítico para la continuidad operativa, la seguridad de la información y la escalabilidad de los servicios digitales. La implementación de prácticas DevOps permite integrar desarrollo y operaciones, reduciendo tiempos de entrega, mejorando la calidad del software y optimizando recursos.

Sin embargo, se observa una brecha entre la creciente demanda de técnicos con competencias específicas en infraestructura cloud y automatización, y la oferta formativa sistemática en el ámbito universitario técnico. La mayoría de las capacitaciones disponibles se encuentran fragmentadas en certificaciones aisladas, sin una estructura académica integral que articule fundamentos, prácticas y responsabilidad profesional.

La presente tecnicatura se propone atender esta necesidad estructural, formando técnicos universitarios capaces de intervenir en entornos tecnológicos complejos, garantizando eficiencia operativa, seguridad y calidad en la gestión de infraestructuras digitales.

### **2.2.- Vinculación con la misión institucional**

La Tecnicatura Universitaria en Gestión de Infraestructura Cloud y DevOps se inscribe en la misión institucional de formar profesionales técnicos con alta empleabilidad, orientados a sectores estratégicos de la economía del conocimiento y comprometidos con el desarrollo territorial.

La propuesta responde a la necesidad de fortalecer el capital humano en áreas tecnológicas críticas, promoviendo la formación de recursos locales especializados capaces de insertarse tempranamente en el mercado laboral y contribuir al crecimiento del ecosistema digital regional.





Asimismo, la carrera favorece la articulación entre el sistema universitario y el entramado productivo tecnológico, promoviendo la vinculación con empresas de software, proveedores de servicios cloud, startups, pymes tecnológicas y organismos públicos en proceso de modernización digital.

De este modo, la tecnicatura amplía la oferta académica en un campo estratégico, contribuyendo a la diversificación formativa y al desarrollo sostenible basado en el conocimiento y la innovación tecnológica.

### **2.3.- Fundamentación académica y formativa**

La estructura curricular de la Tecnicatura Universitaria en Gestión de Infraestructura Cloud y DevOps se organiza de manera progresiva y articulada, integrando tres campos de formación: fundamentos tecnológicos, formación específica y prácticas profesionalizantes.

El campo de fundamentos incluye contenidos de programación, sistemas operativos, redes, bases de datos, arquitectura de computadoras y seguridad informática, proporcionando la base conceptual necesaria para comprender el funcionamiento de infraestructuras tecnológicas complejas.

La formación específica aborda administración de sistemas Linux y Windows Server, virtualización, contenedores, orquestación, infraestructura como código, integración y entrega continua (CI/CD), monitoreo, observabilidad, automatización de procesos y gestión de servicios en la nube pública, privada e híbrida.

De manera articulada, las prácticas profesionalizantes permiten el desarrollo de competencias en entornos reales o simulados, promoviendo el aprendizaje basado en proyectos, la resolución de problemas técnicos concretos y el trabajo colaborativo bajo metodologías ágiles.

Asimismo, la inclusión de contenidos vinculados a seguridad informática, protección de datos, documentación técnica, ética profesional y responsabilidad social fortalece la formación integral del egresado, en consonancia con los estándares actuales de la industria tecnológica.

La propuesta curricular contempla la actualización permanente frente a la rápida evolución del sector, asegurando coherencia entre los contenidos académicos y las tecnologías vigentes en el ámbito productivo.

### **2.4.- Contribución al desarrollo regional**





La implementación de esta tecnicatura contribuirá al fortalecimiento del ecosistema tecnológico regional, promoviendo la generación de empleo calificado y la consolidación de capacidades locales en infraestructura digital.

Al formar técnicos universitarios especializados en gestión de infraestructura cloud y prácticas DevOps, la institución aporta al desarrollo de la economía del conocimiento, al agregado de valor tecnológico y a la modernización de organizaciones públicas y privadas.

La carrera favorece la radicación y crecimiento de empresas tecnológicas en la región, reduce la dependencia de servicios externos especializados y promueve la innovación mediante la disponibilidad de recursos humanos capacitados.

En este sentido, la tecnicatura se constituye como una propuesta estratégica para acompañar la transformación digital, fortaleciendo el desarrollo productivo sustentado en tecnología, conocimiento y trabajo calificado.

### **3.- OBJETIVOS DE LA CARRERA**

#### **Objetivo General**

Formar técnicos universitarios en Gestión de Infraestructura Cloud y DevOps con una sólida base tecnológica, metodológica y práctica, capaces de diseñar, implementar, automatizar y administrar infraestructuras digitales en entornos cloud, aplicando metodologías de integración y entrega continua, garantizando disponibilidad, escalabilidad, seguridad y eficiencia operativa, y contribuyendo al desarrollo tecnológico y productivo en el marco de estándares de calidad, seguridad informática y ética profesional.

#### **Objetivos Específicos**

La carrera tiene como objetivos específicos:

1. Proporcionar una formación técnica básica en programación, sistemas operativos, redes, bases de datos y arquitectura de sistemas que permita comprender los fundamentos del funcionamiento de infraestructuras tecnológicas modernas.
2. Desarrollar competencias para la implementación y administración de infraestructuras en la nube en entornos públicos, privados e híbridos, aplicando buenas prácticas de configuración, monitoreo y optimización de recursos.





3. Formar en el uso de herramientas de virtualización, contenedorización y orquestación que permitan gestionar entornos distribuidos de manera eficiente y escalable.
4. Formar en la aplicación de prácticas DevOps, integración continua (CI) y entrega continua (CD), promoviendo la automatización de procesos mediante infraestructura como código y pipelines de despliegue.
5. Desarrollar habilidades para el análisis, detección y resolución de incidentes en entornos productivos, implementando estrategias de observabilidad, monitoreo y gestión de eventos.
6. Favorecer el desarrollo de competencias en documentación técnica, comunicación profesional y trabajo colaborativo bajo metodologías ágiles.
7. Promover prácticas responsables en el uso de tecnologías digitales, incorporando principios de ética profesional, protección de la información y responsabilidad social en el manejo de infraestructuras críticas.
8. Preparar a los estudiantes para su inserción laboral mediante prácticas profesionalizantes progresivas y espacios de integración en entornos reales o simulados de infraestructura tecnológica.
9. Contribuir a la formación de talentos calificados que fortalezcan el ecosistema tecnológico regional y apoyen el desarrollo de la economía del conocimiento.

#### **4.- PERFIL DEL EGRESADO**

El egresado de la Tecnicatura Universitaria en Gestión de Infraestructura Cloud y DevOps será un técnico universitario capacitado para desempeñarse en actividades vinculadas a la implementación, administración y automatización de infraestructuras tecnológicas en entornos de computación en la nube, interviniendo en la configuración, despliegue, monitoreo y optimización de servicios digitales bajo estándares de calidad, seguridad informática y buenas prácticas operativas.

Poseerá una sólida formación tecnológica que le permitirá comprender los fundamentos de los sistemas operativos, redes, virtualización, contenedorización y arquitectura de infraestructura digital, aplicando herramientas de infraestructura como código y metodologías DevOps para la integración y entrega continua de aplicaciones. Estará capacitado para gestionar entornos cloud públicos, privados o híbridos, implementar procesos de automatización, administrar recursos





tecnológicos y aplicar criterios de monitoreo y observabilidad que garanticen disponibilidad, escalabilidad y eficiencia operativa.

El egresado estará preparado para integrarse a equipos técnicos interdisciplinarios de desarrollo y operaciones, colaborar en la implementación de soluciones tecnológicas confiables y participar en procesos de transformación digital en organizaciones públicas y privadas. Asimismo, desempeñará su actividad con responsabilidad profesional, ética en el manejo de la información y compromiso con la seguridad de los sistemas y la protección de datos.

## **5.- COMPETENCIAS DEL EGRESADO**

El egresado estará capacitado para:

### **a) Competencias técnicas y operativas**

- Diseñar, implementar y administrar infraestructuras tecnológicas en entornos de computación en la nube (cloud pública, privada e híbrida).
- Configurar y gestionar servicios de virtualización, contenedores y orquestadores para el despliegue de aplicaciones.
- Automatizar procesos mediante herramientas de infraestructura como código y scripts de configuración.
- Implementar y administrar pipelines de integración continua (CI) y entrega continua (CD). Gestionar sistemas operativos y servicios en entornos Linux y/o Windows Server.
- Implementar soluciones de monitoreo, observabilidad y gestión de incidentes en infraestructuras digitales.
- Realizar tareas de despliegue, actualización, mantenimiento y optimización de recursos tecnológicos.

Registrar y documentar configuraciones, procedimientos y cambios en entornos productivos.

### **b) Competencias en calidad, seguridad y normativa**

- Aplicar buenas prácticas de seguridad informática en la administración de infraestructuras y servicios digitales.
- Implementar políticas básicas de protección de datos, control de accesos y respaldo de información.

Garantizar disponibilidad, escalabilidad y continuidad operativa de los sistemas.





- Actuar conforme a la normativa vigente en materia de protección de datos y seguridad de la información.
- Aplicar estándares de calidad y procedimientos técnicos en la gestión de servicios tecnológicos.
- Desempeñarse con ética profesional en el manejo de información crítica y recursos digitales.

### **c) Competencias transversales**

- Integrarse y trabajar en equipos técnicos interdisciplinarios bajo metodologías ágiles.
- Comunicar información técnica de manera clara y adecuada a distintos niveles organizacionales.
- Analizar y resolver problemas operativos en entornos tecnológicos dinámicos.
- Adaptarse a tecnologías emergentes y promover la actualización permanente de conocimientos.
- Desarrollar una actitud responsable, colaborativa y orientada a la mejora continua en el desempeño profesional.

## **6.- ALCANCES DEL TÍTULO**

El título de Técnico Universitario en Gestión de Infraestructura Cloud y DevOps habilita a los egresados a desempeñarse en ámbitos públicos y privados vinculados a la administración de infraestructuras tecnológicas, servicios digitales y procesos de automatización, en tareas de apoyo técnico, operativo y de gestión de entornos informáticos, bajo la supervisión de profesionales responsables cuando la complejidad del proyecto así lo requiera.

En particular, el egresado estará habilitado para:

- Implementar y administrar infraestructuras en la nube en entornos públicos, privados o híbridos.
- Configurar y gestionar servicios de virtualización, contenedores y herramientas de orquestación.
- Automatizar procesos de despliegue y configuración mediante herramientas de infraestructura como código.
- Implementar y mantener entornos de integración continua (CI) y entrega continua (CD). Administrar sistemas operativos y servicios en entornos Linux y/o Windows Server.





- Implementar soluciones de monitoreo, observabilidad y gestión de incidentes en infraestructuras digitales.
- Aplicar políticas básicas de seguridad informática, control de accesos y respaldo de información.
- Documentar configuraciones, procedimientos técnicos e informes operativos. Colaborar en proyectos de migración, modernización y transformación digital.
- Integrar equipos técnicos en empresas de tecnología, áreas de infraestructura de organizaciones públicas o privadas, proveedores de servicios cloud, startups y consultoras tecnológicas.

Todas estas actividades se desarrollarán sin perjuicio de las incumbencias propias de los profesionales universitarios de grado, tales como ingenieros en sistemas, ingenieros en informática, licenciados en informática u otros responsables técnicos que requieran intervención en diseño de arquitectura de alta complejidad, definición estratégica o certificaciones profesionales específicas.

## **7.- REQUISITOS DE INGRESO**

Podrán cursar esta tecnicatura universitaria aquellas personas que:

- Posean título secundario o equivalente completo, cualquiera sea su modalidad, emitidos por instituciones de gestión estatal o privada y consten con el debido reconocimiento ministerial, conforme lo establece el artículo 7º de la Ley de Educación Superior Nº 24521.
- Quienes no posean título secundario o equivalente, pero que sean mayores de 25 años y se encuentren en el marco de excepcionalidad establecido en la segunda parte del artículo 7º de la Ley de Educación Superior Nº 24521.
- Acrediten título secundario completo, emitido por otro país, pero debidamente reconocido por el Ministerio de Relaciones Exteriores y Culto y el Ministerio de Educación de la Nación.
- Registren su preinscripción en las fechas establecidas según resolución rectoral.





## 8.- PLAN DE ESTUDIOS Y ESTRUCTURA CURRICULAR

| COD                                  | MÓDULO   | RÉGIMEN       | CARGA HORARIA SEMANAL | CARGA HORARIA TOTAL | CORRELATIVAS PARA CERTIFICAR | CAMPO DE FORMACIÓN |
|--------------------------------------|--|---------------|-----------------------|---------------------|------------------------------|--------------------|
| <b>PRIMER AÑO (1° CUATRIMESTRE)</b>  |  |               |                       |                     |                              |                    |
| 1                                    | Administración de Sistemas Linux                     | cuatrimestral | 4                     | 64                  | -                            | Fundamento         |
| 2                                    | Programación para Automatización I                   | cuatrimestral | 5                     | 80                  | -                            | Fundamento         |
| 3                                    | Redes y Conectividad Cloud                           | cuatrimestral | 4                     | 64                  | -                            | Fundamento         |
| 4                                    | Cultura DevOps y Agilidad                            | cuatrimestral | 4                     | 64                  | -                            | General            |
| <b>PRIMER AÑO (2° CUATRIMESTRE)</b>  |  |               |                       |                     |                              |                    |
| 5                                    | Virtualización y Cloud Fundamentals                  | cuatrimestral | 4                     | 64                  | 1 - 3                        | Específica         |
| 6                                    | Programación para Automatización II                  | cuatrimestral | 4                     | 64                  | 2                            | Fundamento         |
| 7                                    | Gestión de Datos y Bases NoSQL                       | cuatrimestral | 4                     | 64                  | 2                            | Específica         |
| 8                                    | Seguridad en Infraestructura y Networking Cloud      | cuatrimestral | 4                     | 64                  | 1 - 3                        | Específica         |
| 9                                    | Inglés Técnico I                                     | cuatrimestral | 3                     | 48                  | -                            | General            |
| <b>SEGUNDO AÑO (1° CUATRIMESTRE)</b> |  |               |                       |                     |                              |                    |
| 10                                   | Arquitectura de Sistemas Distribuidos                | cuatrimestral | 6                     | 96                  | 3 - 7                        | Específica         |
| 11                                   | Contenedores y Microservicios                        | cuatrimestral | 5                     | 80                  | 1 - 5                        | Específica         |
| 12                                   | Infraestructura como Código (IaC)                    | cuatrimestral | 5                     | 80                  | 5 - 6                        | Específica         |
| 13                                   | CI/CD y DevSecOps                                    | cuatrimestral | 4                     | 64                  | 4 - 6                        | Práctica           |
| 14                                   | Orquestación de Contenedores (Kubernetes)            | cuatrimestral | 5                     | 80                  | -                            | Específica         |
| <b>SEGUNDO AÑO (2° CUATRIMESTRE)</b> |  |               |                       |                     |                              |                    |
| 15                                   | Inglés Técnico II                                    | cuatrimestral | 4                     | 64                  | 9                            | Específica         |
| 16                                   | Gobernanza Cloud y FinOps                            | cuatrimestral | 5                     | 80                  | 5 - 8                        | Específica         |
| 17                                   | Arquitecturas Serverless y Event-Driven              | cuatrimestral | 5                     | 80                  | 10 - 12                      | Específica         |
| 18                                   | Observabilidad y Monitoreo                           | cuatrimestral | 5                     | 80                  | 10 - 13                      | Específica         |
| <b>TERCER AÑO (1° CUATRIMESTRE)</b>  |  |               |                       |                     |                              |                    |
| 19                                   | Site Reliability Engineering (SRE)                   | cuatrimestral | 6                     | 96                  | 14 - 18                      | Práctica           |
| 20                                   | Platform Engineering e Infraestructura como Producto | cuatrimestral | 5                     | 80                  | 12 - 14                      | General            |
| 21                                   | Inglés Técnico III                                   | cuatrimestral | 4                     | 64                  | 15                           | General            |
| 22                                   | Proyecto Integrador Final                            | cuatrimestral | 6                     | 96                  | -                            | Práctica           |





## 9. – Contenidos y Bibliografía sugerida

### 1 - Administración de Sistemas Linux

Arquitectura general del sistema operativo GNU/Linux, rol del kernel y espacio de usuario, gestión de procesos y memoria, estructura del sistema de archivos y estándar FHS, instalación y configuración inicial de servidores Linux, diferencias entre distribuciones orientadas a entornos empresariales, administración de usuarios y grupos, sistema de permisos tradicionales y listas de control de acceso (ACL), delegación de privilegios mediante sudo, administración y gestión de servicios con systemd, monitoreo de procesos y análisis de logs del sistema, fundamentos de hardening y buenas prácticas de seguridad en servidores Linux, particionado de discos y gestión de sistemas de archivos, administración de volúmenes lógicos con LVM y nociones básicas de RAID, configuración de red en servidores Linux, acceso remoto seguro mediante SSH, montaje automático de dispositivos y gestión avanzada de logs, fundamentos de scripting en Bash, estructuras de control, funciones y manejo de errores, procesamiento de texto y redirecciones, automatización de tareas administrativas repetitivas, programación de tareas con cron y mantenimiento preventivo del sistema.

#### Bibliografía sugerida

- Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). *UNIX and Linux system administration handbook* (5th ed.). Pearson.
- Shotts, W. E. (2019). *The Linux command line* (2nd ed.). No Starch Press.
- Ward, B. (2021). *How Linux works: What every superuser should know* (3rd ed.). No Starch Press.
- Nakamura, E., & Nakamura, L. (2020). *Linux hardening in hostile networks*. Apress.
- Sobell, M. G. (2017). *A practical guide to Linux commands, editors, and shell programming* (4th ed.). Pearson.

### 2 - Programación para Automatización I

Fundamentos de programación estructurada, variables y tipos de datos primitivos y compuestos, operadores aritméticos, lógicos y relacionales, estructuras de control condicionales y repetitivas, definición y uso de funciones, modularización del código y buenas prácticas de legibilidad y organización, principios de programación orientada a objetos, definición de clases, atributos y métodos, encapsulamiento y separación de responsabilidades, instanciación y reutilización de componentes, aplicación de estructuras orientadas a objetos en proyectos de automatización, lectura y escritura de archivos, manejo de rutas y directorios, ejecución de procesos del sistema desde código, captura y gestión de errores, desarrollo de utilidades para tareas administrativas, fundamentos del protocolo HTTP y métodos principales, intercambio de datos en formato JSON, autenticación básica y mediante token, envío y recepción de solicitudes a APIs REST, integración de servicios web en scripts de automatización, desarrollo de scripts para copias de seguridad y gestión de archivos, automatización de despliegues básicos, consultas a servicios remotos para monitoreo, integración de funcionalidades en herramientas de línea de comandos, uso de Git para control de versiones, gestión de repositorios y ramas, introducción al testing básico de funciones, documentación técnica de scripts, buenas prácticas para trabajo colaborativo en entornos tecnológicos.

#### Bibliografía básica sugerida (APA 7ª edición)

- Downey, A. B. (2015). *Think Python: How to think like a computer scientist* (2nd ed.). O'Reilly Media.





- Matthes, E. (2019). *Python crash course* (2nd ed.). No Starch Press.
- McKinney, W. (2022). *Python for data analysis* (3rd ed.). O'Reilly Media.
- Beazley, D. M. (2013). *Python essential reference* (4th ed.). Addison-Wesley.
- Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress.

### 3 - Redes y Conectividad Cloud

Modelo OSI y modelo TCP/IP, capas y funciones, relación entre ambos modelos, direccionamiento IP, clases de direcciones, CIDR y subnetting, cálculo y diseño básico de subredes, puertos y protocolos principales (HTTP, HTTPS, SSH, FTP, DNS), fundamentos del sistema de nombres de dominio y resolución de nombres, conceptos de switching y funcionamiento de redes LAN, fundamentos de routing estático y dinámico en entornos básicos, segmentación de red mediante VLAN, traducción de direcciones (NAT) y su aplicación en redes corporativas, fundamentos de firewalls y control de tráfico entrante y saliente, arquitectura de redes virtuales en entornos cloud, estructura y componentes de Virtual Private Cloud (VPC), subredes públicas y privadas, Internet Gateway y NAT Gateway, tablas de ruteo en la nube, grupos de seguridad y listas de control de acceso, balanceadores de carga y distribución de tráfico, herramientas de análisis y diagnóstico de red (ping, traceroute, netstat, ss, nslookup), verificación de conectividad y análisis de latencia, identificación de fallos de configuración en IP, DNS y ruteo, lectura básica de registros de red, metodología sistemática de troubleshooting en entornos locales y cloud.

#### Bibliografía sugerida

- Kurose, J. F., & Ross, K. W. (2021). *Computer networking: A top-down approach* (8th ed.). Pearson.
- Tanenbaum, A. S., & Wetherall, D. J. (2021). *Computer networks* (6th ed.). Pearson.
- Odom, W. (2020). *CCNA 200-301 official cert guide* (Vols. 1–2). Cisco Press.
- White, R., & Banks, E. (2018). *Computer networking problems and solutions: An innovative approach to building resilient, modern networks*. Addison-Wesley.
- Amazon Web Services. (2023). *AWS well-architected framework*. AWS.

### 4 - Cultura DevOps y Agilidad

Origen y evolución del movimiento DevOps como respuesta a la separación tradicional entre desarrollo y operaciones, principios culturales de colaboración, automatización, medición y mejora continua, modelo CALMS (Culture, Automation, Lean, Measurement, Sharing) como marco conceptual, trabajo interdisciplinario y reducción de silos organizacionales, impacto de DevOps en la calidad, velocidad y confiabilidad de los sistemas, principios del Manifiesto Ágil, frameworks ágiles como Scrum (roles, eventos y artefactos), Kanban (flujo de trabajo y mejora continua) y nociones de Extreme Programming (XP), planificación iterativa y gestión de backlog, organización y priorización de tareas en equipos técnicos, métricas ágiles para seguimiento de proyectos (velocity, lead time, burn-down chart), fundamentos de Integración Continua (CI) y Entrega/Despliegue Continuo (CD), automatización del ciclo de vida del software, diseño y funcionamiento de pipelines como herramienta de estandarización y control de calidad, relación entre automatización y reducción de errores humanos, mejora de tiempos de entrega, DevOps como modelo organizacional orientado a la mejora continua, cultura de responsabilidad compartida y aprendizaje organizacional.

#### Bibliografía sugerida

- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.





- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum guide*. Scrum.org.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The science of DevOps*. IT Revolution Press.
- Beck, K. (2004). *Extreme programming explained: Embrace change* (2nd ed.). Addison-Wesley.

## 5 - Virtualización y Cloud Fundamentals

Conceptos de virtualización como mecanismo de abstracción del hardware, hipervisores tipo 1 (bare metal) y tipo 2 (hosted), virtualización completa y paravirtualización, creación y administración básica de máquinas virtuales, escenarios de uso en entornos empresariales, definición y evolución del Cloud Computing, modelos de servicio IaaS, PaaS y SaaS, modelos de despliegue de nube pública, privada e híbrida, principio de responsabilidad compartida entre proveedor y cliente, comparación entre infraestructura on-premise y entornos cloud, servicios de cómputo en la nube (instancias y máquinas virtuales), servicios de almacenamiento en bloque, objeto y archivo, fundamentos de redes virtuales y subredes, configuraciones básicas de seguridad en entornos cloud, introducción comparativa a proveedores líderes de servicios en la nube, gestión básica de recursos mediante consola web, conceptos de alta disponibilidad y tolerancia a fallos, escalabilidad vertical y horizontal, balanceadores de carga y mecanismos de Auto Scaling, principios de arquitectura distribuida, modelos de facturación en la nube, estimación y dimensionamiento inicial de recursos, buenas prácticas para optimización de costos y prevención de sobreaprovisionamiento, despliegue integral de infraestructura básica en la nube, configuración de máquina virtual, almacenamiento y red, aplicación de principios de disponibilidad y seguridad básica, documentación técnica del entorno implementado y evaluación integral de la solución desplegada.

### Bibliografía sugerida

- Stallings, W. (2018). *Cloud computing: Concepts, technology & architecture*. Pearson.
- Rittinghouse, J. W., & Ransome, J. F. (2017). *Cloud computing: Implementation, management, and security* (2nd ed.). CRC Press.
- Portnoy, M. (2016). *Virtualization essentials* (2nd ed.). Wiley.
- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.
- Amazon Web Services. (2023). *AWS well-architected framework*. AWS.

## 6 - Programación para Automatización II

Programación avanzada orientada a entornos productivos, manejo estructurado de excepciones y control de errores, implementación de logging estructurado y niveles de registro, validación y sanitización de datos de entrada, organización del código en módulos reutilizables y mantenibles, buenas prácticas para desarrollo de herramientas de automatización robustas, uso de SDKs oficiales de proveedores cloud para creación y gestión programática de recursos, autenticación segura y gestión de credenciales, automatización de instancias, almacenamiento y redes desde código, gestión del ciclo de vida de recursos en entornos cloud, prevención de configuraciones manuales mediante automatización, integración con servicios externos a través de APIs, uso de colas de mensajería y arquitecturas pub/sub, implementación de webhooks y disparadores de eventos, diseño de flujos automatizados basados en eventos, gestión de errores y reintentos en sistemas distribuidos, diseño e implementación de soluciones integradas que automaticen provisión, configuración y monitoreo de infraestructura, integración con APIs externas y servicios cloud,





registro estructurado de eventos, documentación técnica del sistema desarrollado y presentación formal de la solución implementada.

### Bibliografía sugerida

- Beazley, D. M. (2013). *Python essential reference* (4th ed.). Addison-Wesley.
- Matthes, E. (2019). *Python crash course* (2nd ed.). No Starch Press.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.
- Amazon Web Services. (2023). *AWS well-architected framework*. AWS.

## 7 - Gestión de Datos y Bases NoSQL

Fundamentos de bases de datos relacionales, estructura tabular, claves primarias y foráneas, nociones de normalización básica, limitaciones del modelo relacional en escenarios distribuidos y de alta escala, surgimiento de bases de datos NoSQL en contextos de sistemas distribuidos, criterios de selección de tecnologías según tipo de aplicación, volumen de datos, latencia y requerimientos de consistencia, bases de datos documentales y almacenamiento en formato JSON, bases clave-valor y su aplicación en sistemas de caching y gestión de sesiones, bases de datos columnares para grandes volúmenes de información, bases orientadas a grafos y modelado de relaciones complejas, ventajas y limitaciones de cada modelo, servicios administrados de bases de datos en la nube, diferencias entre despliegue autogestionado y gestionado, configuración básica de instancias de bases de datos en proveedores cloud, backups automáticos, monitoreo y escalado, buenas prácticas de seguridad y control de acceso, conceptos de replicación y particionamiento (sharding), consistencia fuerte y consistencia eventual, relación con el teorema CAP, alta disponibilidad en sistemas distribuidos, impacto de la arquitectura de datos en la resiliencia y desempeño del sistema.

### Bibliografía sugerida

- Sadalage, P. J., & Fowler, M. (2013). *NoSQL distilled: A brief guide to the emerging world of polyglot persistence*. Addison-Wesley.
- Kleppmann, M. (2017). *Designing data-intensive applications*. O'Reilly Media.
- Redmond, E., & Wilson, J. R. (2012). *Seven databases in seven weeks*. Pragmatic Bookshelf.
- Stonebraker, M., & Hellerstein, J. M. (2018). *Readings in database systems* (5th ed.). MIT Press.
- Amazon Web Services. (2023). *Database whitepaper*. AWS.

## 8 - Seguridad en Infraestructura y Networking Cloud

Conceptos fundamentales de ciberseguridad aplicados a infraestructuras tecnológicas, modelo de seguridad CIA (confidencialidad, integridad y disponibilidad), identificación de amenazas comunes en entornos cloud y redes empresariales, análisis de superficie de ataque en infraestructuras virtualizadas, principios de seguridad por diseño y defensa en profundidad, segmentación de redes y aislamiento de entornos, configuración y administración de firewalls, fundamentos de sistemas de detección y prevención de intrusiones (IDS/IPS), hardening de sistemas operativos y servicios expuestos, buenas prácticas para minimizar la exposición pública de recursos en la nube, gestión de identidades digitales en entornos cloud, administración de usuarios, roles y políticas de acceso (IAM), principio de mínimo privilegio, autenticación multifactor (MFA), gestión segura de credenciales y secretos, control de acceso basado en roles (RBAC), registros de auditoría y monitoreo de eventos de seguridad, fundamentos de cumplimiento normativo en infraestructuras





cloud, evaluación de configuraciones inseguras, gestión básica de incidentes de seguridad, documentación de eventos y aplicación de procesos de mejora continua en entornos tecnológicos.

#### **Bibliografía sugerida**

- Stallings, W., & Brown, L. (2018). *Computer security: Principles and practice* (4th ed.). Pearson.
- Anderson, R. (2020). *Security engineering: A guide to building dependable distributed systems* (3rd ed.). Wiley.
- National Institute of Standards and Technology. (2018). *Framework for improving critical infrastructure cybersecurity (Version 1.1)*. NIST.
- Rittinghouse, J. W., & Ransome, J. F. (2017). *Cloud computing: Implementation, management, and security* (2nd ed.). CRC Press.
- Amazon Web Services. (2023). *AWS security best practices whitepaper*. AWS.

#### **9 - Inglés Técnico I**

Fundamentos de comprensión lectora en inglés aplicado a tecnologías de la información y ciberseguridad, vocabulario técnico específico de informática, redes y seguridad digital, estructuras gramaticales básicas para lectura e interpretación de textos técnicos, análisis de manuales, documentación técnica y artículos especializados, comprensión de especificaciones técnicas y estándares internacionales, lectura e interpretación de normativas y reportes de seguridad, terminología relacionada con sistemas operativos, redes, criptografía y gestión de incidentes, redacción básica de informes técnicos en inglés, elaboración de correos y comunicaciones formales en entornos tecnológicos, comprensión auditiva de presentaciones técnicas y conferencias especializadas, uso de glosarios técnicos y herramientas de traducción asistida, análisis de casos prácticos mediante documentación original en inglés, desarrollo de habilidades para interpretar documentación de fabricantes y proveedores tecnológicos, fortalecimiento de la autonomía en la lectura de fuentes académicas y técnicas en idioma inglés.

#### **Bibliografía sugerida**

- Glendinning, E. H., & McEwan, J. (2013). *Oxford English for information technology* (2nd ed.). Oxford University Press.
- Hewings, M. (2013). *Advanced grammar in use* (3rd ed.). Cambridge University Press.
- Murphy, R. (2019). *English grammar in use* (5th ed.). Cambridge University Press.
- National Institute of Standards and Technology. (2018). *Framework for improving critical infrastructure cybersecurity (Version 1.1)*. NIST.
- Stallings, W. (2018). *Computer security: Principles and practice* (4th ed.). Pearson.

#### **10 - Arquitectura de Sistemas Distribuidos**

Concepto de sistema distribuido y diferencias con arquitecturas monolíticas, impacto de la latencia y la comunicación en red en el diseño arquitectónico, concurrencia y ejecución paralela en múltiples nodos, sincronización y coordinación entre procesos distribuidos, condiciones de carrera y consistencia de estado, teorema CAP y sus implicancias prácticas en sistemas reales, consistencia fuerte y consistencia eventual, replicación de datos y tolerancia a fallos, particionamiento (sharding) y distribución de carga, análisis de trade-offs arquitectónicos, arquitecturas basadas en servicios y desacoplamiento funcional, comunicación síncrona mediante REST, fundamentos de gRPC y comunicación binaria eficiente, serialización de datos y contratos de servicio, manejo de errores, latencia y timeouts en llamadas remotas, comunicación asíncrona y arquitecturas orientadas a





eventos, mensajería mediante colas y sistemas publish/subscribe, procesamiento basado en eventos y desacoplamiento, garantías de entrega y manejo de fallos en sistemas de mensajería, escalabilidad horizontal y balanceo de carga, patrones de resiliencia como retry, timeout y circuit breaker, tolerancia a fallos y recuperación automática, diseño para alta disponibilidad, análisis de requerimientos no funcionales (rendimiento, disponibilidad, seguridad y escalabilidad), selección de patrones arquitectónicos adecuados, identificación de puntos únicos de fallo, diseño integral de una arquitectura distribuida, documentación técnica y justificación de decisiones arquitectónicas.

### **Bibliografía sugerida**

- Kleppmann, M. (2017). *Designing data-intensive applications*. O'Reilly Media.
- Newman, S. (2021). *Building microservices* (2nd ed.). O'Reilly Media.
- Tanenbaum, A. S., & Van Steen, M. (2017). *Distributed systems: Principles and paradigms* (3rd ed.). Pearson.
- Richardson, C. (2018). *Microservices patterns*. Manning Publications.
- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.

### **11 - Contenedores y Microservicios**

Concepto de contenedor y diferencias con máquinas virtuales, arquitectura de virtualización ligera basada en contenedores, Docker como herramienta de contenedorización, imágenes, contenedores y ciclo de vida, construcción de imágenes mediante Dockerfile, gestión de versiones y uso de registros de imágenes, definición de aplicaciones multicontenedor mediante Docker Compose, orquestación básica de múltiples contenedores, persistencia de datos y volúmenes, configuración de redes internas y entornos de desarrollo reproducibles, principios de arquitectura de microservicios, desacoplamiento y responsabilidad única, comunicación entre servicios, comparación con arquitecturas monolíticas, redes bridge y overlay, exposición de puertos y configuración de servicios, descubrimiento de servicios y gestión de dependencias entre contenedores, buenas prácticas para construcción de imágenes seguras, reducción de superficie de ataque, gestión de secretos en entornos contenerizados, actualización y escaneo básico de vulnerabilidades, diseño e implementación de una aplicación basada en microservicios, contenerización integral de la solución, documentación técnica y despliegue funcional del sistema implementado.

### **Bibliografía sugerida**

- Burns, B., & Beda, J. (2019). *Kubernetes: Up and running* (2nd ed.). O'Reilly Media.
- Newman, S. (2021). *Building microservices* (2nd ed.). O'Reilly Media.
- Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, 2014(239).
- Richardson, C. (2018). *Microservices patterns*. Manning Publications.
- Hightower, K., Burns, B., & Beda, J. (2017). *Kubernetes: The definitive guide*. O'Reilly Media.

### **12 - Infraestructura como Código (IaC)**

Concepto de Infraestructura como Código (IaC) como paradigma de automatización y gestión programática de infraestructura, enfoque declarativo y principios de configuración reproducible, ventajas frente a configuraciones manuales y reducción de errores humanos, fundamentos de herramientas de provisión automatizada, sintaxis básica de Terraform, definición de recursos, variables, outputs y proveedores, creación automatizada de infraestructura en entornos cloud,





planificación y aplicación de cambios, importancia de los archivos de estado en IaC, gestión de estado remoto y colaboración en equipos de trabajo, bloqueo y control de cambios concurrentes, diseño modular y reutilización de componentes, creación y organización de módulos, estructuración de proyectos de infraestructura, buenas prácticas de nomenclatura y organización de código, integración de IaC con pipelines de Integración y Entrega Continua (CI/CD), validación automatizada y control de cambios en infraestructura, revisión de código aplicada a definiciones de infraestructura, provisión automatizada de múltiples entornos (desarrollo, pruebas y producción), documentación técnica de la infraestructura implementada y validación funcional de la solución desplegada.

### Bibliografía sugerida

- Brikman, Y. (2019). *Terraform: Up & running* (2nd ed.). O'Reilly Media.
- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
- Morris, K. (2020). *Infrastructure as code: Managing servers in the cloud* (2nd ed.). O'Reilly Media.
- Amazon Web Services. (2023). *AWS well-architected framework*. AWS

### 13 - CI/CD y DevSecOps

Concepto de pipeline de Integración Continua y Entrega/Despliegue Continuo (CI/CD), automatización del ciclo de vida del software desde el versionado hasta el despliegue, diseño estructurado de pipelines, integración automática de código y validación temprana de errores, ejecución automatizada de pruebas unitarias e integrales, control de calidad del código y análisis de métricas básicas de integración continua, monitoreo de resultados de build y gestión de artefactos, incorporación de principios DevSecOps en el ciclo de desarrollo, integración de controles de seguridad dentro de pipelines automatizados, análisis estático de código y verificación de dependencias, detección de vulnerabilidades en imágenes de contenedores, gestión segura de secretos en entornos automatizados, automatización de despliegues en infraestructura cloud, integración con contenedores y herramientas de orquestación, construcción de pipelines completos funcionales que integren código, infraestructura y seguridad, documentación técnica del flujo implementado y buenas prácticas de automatización segura.

### Bibliografía sugerida

- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The science of DevOps*. IT Revolution Press.
- Rahman, M., & Williams, L. (2020). *DevSecOps: Integrating security into DevOps*. Apress.
- Burns, B., & Beda, J. (2019). *Kubernetes: Up and running* (2nd ed.). O'Reilly Media.

### 14 - Orquestación de Contenedores (Kubernetes)

Arquitectura general de Kubernetes, componentes del cluster y su funcionamiento, rol del control plane y nodos de trabajo, principios básicos de orquestación de contenedores, gestión declarativa del estado del sistema, objetos fundamentales de Kubernetes como Pods, ReplicaSets y Deployments, definición y exposición de aplicaciones mediante Services, gestión declarativa de





recursos a través de manifiestos YAML, ConfigMaps y Secrets para configuración desacoplada, administración de volúmenes y almacenamiento persistente, gestión de configuración en entornos dinámicos, networking interno y externo en clusters, control de tráfico mediante Ingress, políticas de red y segmentación de comunicaciones, escalado manual y automático de aplicaciones, estrategias de actualización y despliegue (rolling update, rollback), diseño para alta disponibilidad en clusters distribuidos, monitoreo básico del estado de recursos, despliegue integral de una aplicación en entorno Kubernetes, aplicación de buenas prácticas de organización, seguridad y estructuración de namespaces, documentación técnica del entorno implementado.

### **Bibliografía sugerida**

- Burns, B., & Beda, J. (2019). *Kubernetes: Up & running* (2nd ed.). O'Reilly Media.
- Hightower, K., Burns, B., & Beda, J. (2017). *Kubernetes: The definitive guide*. O'Reilly Media.
- Newman, S. (2021). *Building microservices* (2nd ed.). O'Reilly Media.
- Richardson, C. (2018). *Microservices patterns*. Manning Publications.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.

### **15 - Inglés Técnico II**

Profundización en comprensión lectora de textos técnicos avanzados en ciberseguridad y tecnología digital, análisis de documentación especializada sobre seguridad informática, redes, criptografía y gestión de incidentes, interpretación de normativas y estándares internacionales redactados en inglés (ISO, NIST, RFC), ampliación del vocabulario técnico específico en ciberseguridad, análisis de reportes de vulnerabilidades y documentos de fabricantes, comprensión de papers técnicos y artículos académicos del área, redacción de informes técnicos y ejecutivos en inglés, elaboración de documentación de incidentes y reportes de auditoría en idioma inglés, comunicación escrita formal en entornos profesionales tecnológicos, comprensión auditiva de presentaciones, webinars y conferencias técnicas internacionales, simulación de reuniones técnicas en inglés, análisis crítico de documentación de seguridad y manuales de configuración, uso autónomo de fuentes técnicas internacionales, desarrollo de habilidades para certificaciones técnicas que requieran comprensión de material en inglés.

### **Bibliografía sugerida**

- Glendinning, E. H., & McEwan, J. (2013). *Oxford English for information technology* (2nd ed.). Oxford University Press.
- Hewings, M. (2013). *Advanced grammar in use* (3rd ed.). Cambridge University Press.
- Murphy, R. (2019). *English grammar in use* (5th ed.). Cambridge University Press.
- National Institute of Standards and Technology. (2018). *Framework for improving critical infrastructure cybersecurity (Version 1.1)*. NIST.
- Stallings, W., & Brown, L. (2018). *Computer security: Principles and practice* (4th ed.). Pearson.

### **16 - Gobernanza Cloud y FinOps**

Concepto de gobernanza en entornos cloud, marcos de referencia y buenas prácticas para gestión organizacional de infraestructura en la nube, establecimiento de políticas y estándares para provisión y uso de recursos, control organizacional y jerarquización de cuentas y proyectos, definición de roles y responsabilidades en la administración cloud, fundamentos de gestión de riesgos en infraestructura tecnológica, cumplimiento normativo en entornos cloud y alineación con estándares internacionales, evaluación de configuraciones inseguras y adopción de controles





preventivos, monitoreo de cumplimiento y auditoría de recursos, introducción a FinOps como disciplina de optimización financiera en la nube, análisis de consumo y eficiencia de recursos, identificación de desperdicios y sobreaprovisionamiento, estrategias de optimización de costos, dimensionamiento adecuado de infraestructura, métricas financieras aplicadas a infraestructura cloud, elaboración de reportes técnicos y ejecutivos, interpretación de indicadores para toma de decisiones estratégicas, alineación entre equipos técnicos y financieros, mejora continua basada en datos de uso y desempeño.

### **Bibliografía sugerida**

- Weinman, J. (2012). *Clouconomics: The business value of cloud computing*. Wiley.
- Rittinghouse, J. W., & Ransome, J. F. (2017). *Cloud computing: Implementation, management, and security* (2nd ed.). CRC Press.
- FinOps Foundation. (2023). *FinOps framework*. FinOps Foundation.
- National Institute of Standards and Technology. (2018). *Framework for improving critical infrastructure cybersecurity (Version 1.1)*. NIST.
- Amazon Web Services. (2023). *AWS well-architected framework*. AWS.

### **17 - Arquitecturas Serverless y Event-Driven**

Concepto de Serverless Computing y evolución del modelo de ejecución bajo demanda, Function as a Service (FaaS) y sus características principales, escalabilidad automática y modelo de facturación basado en consumo, comparación entre arquitecturas tradicionales y serverless, uso de servicios gestionados en la nube como bases de datos, almacenamiento y mensajería, desacoplamiento de componentes mediante servicios administrados, reducción de gestión operativa de infraestructura, arquitectura basada en eventos y procesamiento asíncrono, mensajería y sistemas publish/subscribe, desencadenadores de eventos (event triggers) y automatización de flujos, integración entre funciones y servicios cloud, diseño de soluciones escalables y resilientes bajo paradigma serverless, evaluación de ventajas y limitaciones del modelo, análisis de latencia, costos y vendor lock-in, buenas prácticas de seguridad y control de acceso en entornos serverless, documentación técnica de arquitectura implementada y justificación de decisiones de diseño.

### **Bibliografía sugerida**

- Roberts, M., & Chapin, J. (2017). *Serverless architectures on AWS*. Manning Publications.
- Burns, B., & Beda, J. (2019). *Kubernetes: Up & running* (2nd ed.). O'Reilly Media.
- Newman, S. (2021). *Building microservices* (2nd ed.). O'Reilly Media.
- Kleppmann, M. (2017). *Designing data-intensive applications*. O'Reilly Media.
- Amazon Web Services. (2023). *Serverless architectures whitepaper*. AWS.

### **18 - Observabilidad y Monitoreo**

Concepto de observabilidad en sistemas distribuidos y diferencia con monitoreo tradicional, componentes fundamentales de observabilidad como logs, métricas y trazas distribuidas, instrumentación de aplicaciones para recolección de datos operativos, fundamentos de monitoreo de infraestructura y aplicaciones, uso de herramientas como Prometheus para recolección de métricas, configuración de exporters y consultas básicas, visualización y análisis de métricas mediante Grafana, construcción de dashboards operativos, definición de indicadores clave de confiabilidad (SLIs) y objetivos de nivel de servicio (SLOs), medición de disponibilidad, latencia y rendimiento, fundamentos de ingeniería de confiabilidad del sitio (SRE), diseño de alertas y umbrales adecuados, gestión de eventos e incidentes operativos, análisis posterior a incidentes





(postmortem) y aprendizaje organizacional, mejora continua basada en datos y métricas, integración de monitoreo en arquitecturas distribuidas y entornos cloud, buenas prácticas para operación resiliente de sistemas productivos.

#### **Bibliografía sugerida**

- Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media.
- Beyer, B., Murphy, N. R., Rensin, D. K., Kawahara, J., & Thorne, S. (2018). *The site reliability workbook*. O'Reilly Media.
- Turnbull, J. (2014). *The art of monitoring*. James Turnbull.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.

#### **19 - Site Reliability Engineering (SRE)**

Origen y principios de Site Reliability Engineering (SRE), relación entre desarrollo y operaciones en entornos de alta disponibilidad, confiabilidad como atributo central del sistema, definición y medición de indicadores de nivel de servicio (SLIs) y objetivos de nivel de servicio (SLOs), cultura de responsabilidad compartida entre equipos, protocolos de gestión de incidentes en entornos productivos, clasificación y priorización de incidentes, coordinación de equipos durante eventos críticos, análisis postmortem sin culpabilización y aprendizaje organizacional, concepto de error budgets y su aplicación en la toma de decisiones técnicas, equilibrio entre innovación y estabilidad operativa, automatización como herramienta para reducción de tareas repetitivas (toil), identificación y eliminación de procesos manuales innecesarios, mejora continua orientada a confiabilidad, monitoreo proactivo y prevención de fallos, resiliencia y diseño para tolerancia a errores, integración de prácticas SRE en entornos cloud y sistemas distribuidos.

#### **Bibliografía sugerida**

- Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media.
- Beyer, B., Murphy, N. R., Rensin, D. K., Kawahara, J., & Thorne, S. (2018). *The site reliability workbook*. O'Reilly Media.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The science of DevOps*. IT Revolution Press.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.

#### **20 - Platform Engineering e Infraestructura como Producto**

Concepto de Platform Engineering como evolución del enfoque DevOps, definición de Internal Developer Platforms (IDP) como entornos autoservicio para equipos de desarrollo, principios de autoservicio y reducción de fricción operativa, separación de responsabilidades entre equipos de plataforma y equipos de producto, infraestructura como producto orientada a experiencia del desarrollador (Developer Experience), estandarización de entornos y provisión automatizada de recursos, diseño de plantillas reutilizables y catálogos de servicios internos, definición de “golden paths” como caminos recomendados y validados para desarrollo y despliegue, reducción de complejidad operativa mediante abstracción de infraestructura, alineación entre arquitectura





técnica y objetivos organizacionales, gobernanza de plataformas internas, métricas de adopción y eficiencia de la plataforma, mejora continua basada en feedback de equipos, diseño organizacional orientado a producto y confiabilidad, integración entre arquitectura de plataforma, seguridad, automatización y escalabilidad, documentación técnica y definición de estándares internos.

### **Bibliografía sugerida**

- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The science of DevOps*. IT Revolution Press.
- Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media.
- Newman, S. (2021). *Building microservices* (2nd ed.). O'Reilly Media.

### **21 – Inglés Técnico III**

Comprensión avanzada de documentación técnica especializada en entornos cloud, DevOps y arquitectura de sistemas distribuidos, análisis crítico de whitepapers técnicos y documentación oficial de proveedores cloud, interpretación de estándares internacionales, guías de buenas prácticas y documentación de frameworks tecnológicos, ampliación del vocabulario técnico específico en infraestructura, automatización, seguridad y arquitectura distribuida, redacción de documentación técnica estructurada en inglés (manuales, reportes de implementación, documentación de arquitectura), elaboración de informes ejecutivos orientados a perfiles no técnicos, comunicación oral en contextos profesionales tecnológicos, simulación de presentaciones técnicas y defensas de proyectos en inglés, análisis y redacción de incident reports y postmortems en idioma inglés, comprensión auditiva de conferencias técnicas internacionales, uso autónomo de fuentes académicas y técnicas en inglés, preparación para certificaciones internacionales que requieran comprensión técnica avanzada, desarrollo de competencias comunicacionales para interacción en equipos globales y entornos tecnológicos internacionales.

### **Bibliografía sugerida**

- Glendinning, E. H., & McEwan, J. (2013). *Oxford English for information technology* (2nd ed.). Oxford University Press.
- Hewings, M. (2013). *Advanced grammar in use* (3rd ed.). Cambridge University Press.
- Murphy, R. (2019). *English grammar in use* (5th ed.). Cambridge University Press.
- Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media.
- Amazon Web Services. (2023). *AWS well-architected framework*. AWS.

### **22 – Proyecto Integrador Final**

Formulación integral de un proyecto tecnológico aplicado a entornos cloud, automatización y arquitectura distribuida, análisis de requerimientos funcionales y no funcionales, delimitación del alcance del proyecto y definición de objetivos técnicos medibles, diseño arquitectónico con selección tecnológica fundamentada, elaboración de diagramas de arquitectura y justificación técnica de decisiones adoptadas, provisión automatizada de infraestructura mediante herramientas de Infraestructura como Código (IaC), configuración estructurada de entornos de desarrollo, pruebas y producción, integración de pipelines CI/CD para automatización de construcción, pruebas y despliegue, incorporación de controles de seguridad y validaciones automatizadas en el ciclo de





entrega, implementación de monitoreo y observabilidad del sistema, definición de métricas de rendimiento y disponibilidad, ejecución de pruebas de carga y ajustes de optimización, aplicación de buenas prácticas de resiliencia y escalabilidad, documentación técnica integral del proyecto (arquitectura, infraestructura, seguridad y operación), elaboración de informe ejecutivo, presentación y defensa formal ante tribunal académico, integración de competencias técnicas, organizacionales y estratégicas desarrolladas a lo largo de la carrera.

### **Bibliografía sugerida**

- Humble, J., & Farley, D. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
- Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media.
- Kleppmann, M. (2017). *Designing data-intensive applications*. O'Reilly Media.
- Brikman, Y. (2019). *Terraform: Up & running* (2nd ed.). O'Reilly Media.

## **10.- METODOLOGÍA DE ENSEÑANZA Y EVALUACIÓN**

### **10.1.- Enfoque metodológico**

La propuesta pedagógica de la Tecnicatura Universitaria en Gestión de Infraestructura Cloud y DevOps se fundamenta en un enfoque de formación tecnológica, aplicada y orientada al desarrollo de competencias profesionales acordes al perfil del egresado y a las demandas del sector tecnológico y de la economía del conocimiento.

El proceso de enseñanza y aprendizaje se concibe como un trayecto formativo progresivo que articula fundamentos conceptuales, procedimientos técnicos y actitudes profesionales, priorizando la vinculación sistemática entre los contenidos académicos y las prácticas propias del desempeño en la administración de infraestructuras cloud, automatización de procesos, integración continua y gestión de entornos digitales productivos.

La carrera adopta un enfoque de formación por competencias, entendidas como la integración de saberes conceptuales, operativos y actitudinales que permiten al estudiante desempeñarse adecuadamente en entornos tecnológicos reales o simulados, bajo estándares de calidad, seguridad informática, eficiencia operativa y normativa vigente en materia de protección de datos.

### **Estrategias de enseñanza**

Las estrategias de enseñanza se orientan a favorecer la apropiación de fundamentos técnicos y el desarrollo de habilidades operativas mediante modalidades didácticas acordes a una tecnicatura universitaria con fuerte impronta práctica y tecnológica.





Entre las principales estrategias se contemplan:

- Clases teórico-prácticas destinadas a la introducción, análisis y aplicación de los fundamentos de programación, redes, sistemas operativos, cloud computing y DevOps.
- Laboratorios técnicos orientados a la configuración de entornos virtualizados, despliegue de servicios, automatización mediante infraestructura como código y gestión de pipelines CI/CD.
- Resolución de situaciones problemáticas vinculadas a fallas de infraestructura, incidentes operativos, escalabilidad y seguridad en entornos cloud.
- Talleres integradores en los espacios de Práctica Profesionalizante, con énfasis en la implementación completa de infraestructuras digitales bajo criterios de disponibilidad, resiliencia y monitoreo.
- Desarrollo de proyectos colaborativos que simulan entornos productivos reales, favoreciendo el trabajo en equipo bajo metodologías ágiles.
- Análisis de casos reales de transformación digital, migración a la nube y automatización de procesos tecnológicos.

Estas estrategias promueven una participación activa del estudiante, fortalecen el aprendizaje basado en problemas y facilitan la transferencia de los saberes a contextos laborales concretos.

### **Articulación entre teoría y práctica**

La propuesta pedagógica prevé una articulación permanente entre teoría y práctica a lo largo de toda la carrera. Los contenidos conceptuales se abordan como sustento necesario para comprender la arquitectura de sistemas, el funcionamiento de redes, los principios de automatización y las buenas prácticas de seguridad informática.

Las Prácticas Profesionalizantes I, II, III, IV y V constituyen un eje central del diseño curricular, permitiendo la incorporación progresiva de competencias técnicas vinculadas a la configuración de servidores, administración de contenedores, automatización de despliegues, implementación de servicios cloud y monitoreo de infraestructura.

El Trabajo Final Integrador posibilita la síntesis de los saberes adquiridos mediante el desarrollo de un proyecto integral que simula o interviene en un entorno real de infraestructura tecnológica, fortaleciendo la inserción profesional del estudiante.





### **Modalidad de cursado y uso de entornos virtuales**

En coherencia con la naturaleza tecnológica de la carrera, la propuesta combina instancias presenciales y virtuales de manera planificada.

Las actividades presenciales se destinan prioritariamente al desarrollo de laboratorios técnicos, evaluaciones prácticas, simulaciones de entornos productivos y trabajo colaborativo supervisado.

Las instancias virtuales se orientan al abordaje de contenidos teóricos, acceso a documentación técnica, utilización de plataformas de aprendizaje, repositorios de código, laboratorios en la nube y seguimiento de actividades asincrónicas.

El uso de entornos virtuales no solo cumple función pedagógica complementaria, sino que forma parte del objeto mismo de estudio, dado que la administración de infraestructuras cloud implica el manejo de plataformas digitales distribuidas.

### **Rol del formador y del estudiante**

En el marco de esta propuesta, el formador asume funciones de planificación, guía técnica, supervisión de prácticas y acompañamiento en la resolución de problemas tecnológicos, garantizando la coherencia entre contenidos, competencias y criterios de evaluación.

El estudiante es considerado un sujeto activo de su formación, responsable de la adquisición progresiva de conocimientos técnicos, habilidades operativas y actitudes profesionales acordes al perfil del egresado, promoviendo la autonomía, la actualización permanente y la resolución responsable de problemas.

### **Articulación con el sistema de evaluación**

La propuesta pedagógica se articula de manera coherente con un sistema de evaluación por competencias, asegurando correspondencia entre objetivos formativos, estrategias didácticas y criterios de acreditación. De este modo, enseñanza, aprendizaje y evaluación se integran en un proceso orientado a la mejora continua y la pertinencia profesional.

### **10.2.- Evaluación**

La evaluación de los aprendizajes en la Tecnicatura Universitaria en Gestión de Infraestructura Cloud y DevOps se estructura a partir de normas de competencia que definen, para





cada espacio curricular, los objetivos de aprendizaje, los criterios de desempeño y las evidencias observables que permiten verificar el dominio técnico del estudiante.

La evaluación se concibe como un proceso continuo, formativo y estratégico, orientado a determinar los niveles de competencia alcanzados en la implementación, automatización y administración de infraestructuras tecnológicas.

### **Ejes de la evaluación integral**

La evaluación se organiza en torno a cuatro ejes articulados:

- a. **Fundamentos tecnológicos (Saber):** Se evalúa la comprensión de principios de programación, redes, sistemas operativos, arquitectura cloud y seguridad informática.
- b. **Dominio operativo y tecnológico (Saber hacer):** Se observa y valora la ejecución de configuraciones, despliegues automatizados, implementación de pipelines CI/CD, administración de contenedores y monitoreo de servicios.
- c. **Gestión de calidad y seguridad (Saber hacer con estándares):** Se verifica la aplicación de buenas prácticas, políticas de seguridad, respaldo de información, control de accesos y documentación técnica adecuada.
- d. **Desempeño profesional y transversal (Saber ser):** Se evalúa el trabajo en equipo, la comunicación técnica, la resolución de problemas, la responsabilidad ética y el compromiso con la actualización permanente.

### **Evidencias de competencia**

Las evidencias se organizan en:

- **Evidencias de conocimiento:** evaluaciones teóricas, cuestionarios técnicos, análisis de casos y documentación conceptual.
- **Evidencias de desempeño:** observación directa en laboratorios, configuración de infraestructuras, implementación de servicios cloud y resolución de incidentes.
- **Evidencias de producto:** repositorios de código, configuraciones documentadas, pipelines funcionales, entornos desplegados e informes técnicos.

### **La evaluación como proceso de mejora continua**

La evaluación no se limita a la acreditación formal, sino que constituye un mecanismo de retroalimentación permanente del proceso formativo. Permite validar la pertinencia de las



estrategias didácticas, la adecuación de las prácticas profesionalizantes y la actualización tecnológica de los contenidos, garantizando que la formación responda efectivamente a las demandas del sector productivo digital.

