

# Glosario de Términos Técnicos

## ➤ Siglas y Acrónimos

**DNS** (Domain Name System)

Sistema de nombres de dominio que traduce nombres de dominio legibles por humanos en direcciones IP numéricas.

**HTTPS** (HyperText Transfer Protocol Secure)

Protocolo de transferencia de hipertexto seguro que utiliza cifrado SSL/TLS para proteger las comunicaciones web.

**HTTP** (HyperText Transfer Protocol)

Protocolo de transferencia de hipertexto utilizado para la transmisión de datos en la web.

**SSL** (Secure Sockets Layer)

Protocolo de seguridad que proporciona cifrado de datos entre cliente y servidor.

**TLS** (Transport Layer Security)

Sucesor de SSL, protocolo criptográfico que proporciona seguridad en las comunicaciones a través de redes.

**IP** (Internet Protocol)

Protocolo de internet que define el formato de los paquetes de datos y el esquema de direccionamiento en la red.

**PTR** (Pointer Record)

Registro DNS que realiza la resolución inversa, convirtiendo direcciones IP en nombres de dominio.

**A** (Address Record)

Registro DNS que asocia un nombre de dominio con una dirección IPv4.

**PHP** (Hypertext Preprocessor)

Lenguaje de programación de código abierto especialmente adecuado para el desarrollo web.

**TLD** (Top Level Domain)

Dominio de nivel superior, como .com, .org, .es, etc.

**TTL** (Time To Live)

Valor en segundos que indica cuánto tiempo puede un registro DNS permanecer en caché antes de requerir una nueva consulta al servidor autoritativo.

**SOA (Start of Authority)**

Registro DNS que contiene información administrativa sobre la zona, incluyendo el servidor primario, contacto del administrador y parámetros de sincronización.

**NS (Name Server)**

Registro DNS que especifica qué servidores son autoritativos para una zona determinada.

**➤ Términos Técnicos Generales****Docker**

Plataforma que permite empaquetar aplicaciones y sus dependencias en contenedores portátiles y autosuficientes.

**Contenedor**

Unidad de software que empaqueta código y todas sus dependencias para que la aplicación se ejecute de manera rápida y confiable en diferentes entornos informáticos.

**Docker Compose**

Herramienta para definir y ejecutar aplicaciones Docker multi-contenedor mediante archivos YAML de configuración.

**Bridge (Driver)**

Controlador de red en Docker que permite la comunicación entre contenedores en el mismo host mediante una red privada virtual.

**Bind9**

Servidor DNS de código abierto más utilizado en Internet, desarrollado por ISC (Internet Systems Consortium).

**Nginx**

Servidor web de alto rendimiento, también utilizado como proxy inverso, balanceador de carga y caché HTTP.

**Certificados Auto-firmados**

Certificados SSL/TLS generados localmente sin la validación de una autoridad certificadora externa, útiles para entornos de desarrollo.

**Volumen (Docker)**

Mecanismo para persistir datos generados y utilizados por contenedores Docker, permitiendo compartir archivos entre el host y los contenedores.

**Enlace Simbólico**

Archivo especial que apunta a otro archivo o directorio, permitiendo acceder al mismo contenido desde diferentes ubicaciones.

**Host/Anfitrión**

Máquina física o virtual que ejecuta el sistema Docker y aloja los contenedores.

**nslookup**

Herramienta de línea de comandos para consultar servidores DNS y obtener información sobre registros de dominio.

**GitHub**

Plataforma de alojamiento de código que utiliza Git para el control de versiones y colaboración en proyectos de desarrollo.

**Git**

Sistema de control de versiones distribuido para rastrear cambios en el código fuente durante el desarrollo de software.

**Commit**

Instantánea de los cambios realizados en el repositorio Git, que incluye una descripción de las modificaciones.

➤ **Conceptos de Seguridad y Arquitectura**

**Módulo/Modularidad**

Diseño arquitectónico que divide el sistema en componentes independientes y reutilizables.

**Escalabilidad**

Capacidad del sistema para crecer y gestionar mayor carga de trabajo agregando recursos.

**Aislamiento**

Separación lógica de servicios para evitar interferencias y mejorar la seguridad.

**Repositorio**

Almacén central donde se guarda y gestiona el código fuente y sus versiones.

**Redirección HTTP → HTTPS**

Configuración que fuerza a los usuarios a utilizar conexiones cifradas redirigiendo automáticamente las peticiones HTTP a HTTPS.

**Zona DNS**

Porción del espacio de nombres de dominio administrado por un servidor DNS específico.

**Registro DNS**

Entrada en la base de datos DNS que mapea nombres de dominio a direcciones IP u otros valores.

**Resolución de Nombres**

Proceso de traducir un nombre de dominio en su dirección IP correspondiente.

**Bloque Server (Nginx)**

Sección de configuración en Nginx que define cómo manejar peticiones para un dominio o sitio web específico.

**Registro de Recursos DNS**

Entrada individual en una zona DNS que mapea un nombre a un valor específico (IP, servidor de nombres, alias, etc.).

**Número de Serie (Serial)**

Valor numérico en el registro SOA que se incrementa cada vez que se modifica la zona DNS, utilizado para sincronización entre servidores.

**FastCGI**

Protocolo de comunicación entre el servidor web (Nginx) y aplicaciones de procesamiento dinámico (como PHP), que mejora el rendimiento mediante conexiones persistentes.

**PowerShell**

Lenguaje de scripting y shell de línea de comandos desarrollado por Microsoft, utilizado para automatización de tareas administrativas en Windows.

## ➤ Términos de Automatización

**Script**

Archivo de comandos ejecutables que automatiza tareas repetitivas o complejas.

**.gitignore**

Archivo que especifica qué archivos o directorios Git debe ignorar y no versionar (como certificados, claves privadas, etc.).

**Orquestación**

Coordinación automatizada de múltiples contenedores y servicios para trabajar conjuntamente.

**Despliegue**

Proceso de poner en funcionamiento una aplicación o servicio en un entorno específico.

**Reproducible**

Característica que permite recrear exactamente el mismo entorno en diferentes máquinas o momentos.

## ➤ Herramientas y Comandos

**docker compose up -d --build**

Comando que construye las imágenes y levanta todos los servicios definidos en el archivo docker-compose.yml en modo desacoplado (segundo plano).

**docker inspect**

Comando que muestra información detallada sobre contenedores, redes o volúmenes Docker.

**resolv.conf**

Archivo de configuración del sistema que especifica qué servidores DNS utilizar para la resolución de nombres.

**Caché**

Almacenamiento temporal de datos frecuentemente accedidos para mejorar el rendimiento.

**VS Code**

Editor de código fuente desarrollado por Microsoft, ampliamente utilizado para desarrollo web.

**Firefox**

Navegador web que respeta la configuración DNS del sistema operativo, recomendado para pruebas locales.

➤ **Instrucciones de Dockerfile**

**FROM**

Instrucción que especifica la imagen base sobre la que se construirá el contenedor.

**RUN**

Ejecuta comandos durante la construcción de la imagen Docker, típicamente para instalar paquetes o configurar el sistema.

**EXPOSE**

Declara qué puertos utilizará el contenedor, documentando la intención sin publicarlos automáticamente al host.

**CMD**

Define el comando predeterminado que se ejecutará cuando se inicie el contenedor.

**apt-get**

Gestor de paquetes de sistemas basados en Debian/Ubuntu, utilizado para instalar, actualizar y eliminar software.

➤ **Parámetros de docker-compose.yml**

**services:**

Sección que define los diferentes contenedores que compondrán la aplicación multi-contenedor.

**build:**

Especifica la ruta del Dockerfile o contexto de construcción para crear la imagen del contenedor.

**container\_name:**

Asigna un nombre personalizado al contenedor para facilitar su identificación y gestión.

**hostname:**

Define el nombre de host interno que tendrá el contenedor dentro de la red Docker.

**ports:**

Mapea puertos del contenedor a puertos del host, formato "puerto\_host:puerto\_contenedor", permitiendo acceso externo a servicios internos.

**volumes:**

Monta directorios o archivos del host en el contenedor, permitiendo persistencia de datos y edición externa de configuraciones.

**networks:**

Define a qué red(es) Docker pertenece el contenedor, permitiendo la comunicación entre contenedores.

**dns:**

Configura qué servidores DNS utilizará el contenedor para resolución de nombres.

**cap\_add:**

Otorga capacidades específicas del kernel Linux al contenedor, como permisos administrativos de red.

**command:**

Sobrescribe el comando CMD del Dockerfile con instrucciones personalizadas de inicio.

**restart:**

Define la política de reinicio automático del contenedor (no, always, on-failure, unless-stopped).

**depends\_on:**

Establece dependencias de inicio entre contenedores, controlando el orden de arranque.

**driver:**

Especifica el tipo de controlador a utilizar para redes o volúmenes (bridge, host, overlay, etc.).

**ipam:**

Configuración de gestión de direcciones IP (IP Address Management) para redes personalizadas.

**subnet:**

Define el rango de direcciones IP disponibles para una red Docker personalizada.

**gateway:**

Especifica la dirección IP de la puerta de enlace para una red Docker.