### TERMINOS CLAVES: **.NET**

**Términos clave:**

* **.NET Core**: Plataforma de desarrollo abierta y multiplataforma.
* **ASP.NET**: Framework para desarrollar aplicaciones web.
* **C#**: Lenguaje de programación principal usado en .NET.
* **Entity Framework**: ORM para trabajar con bases de datos.

**Escenarios:**

* Desarrollo de aplicaciones web, de escritorio y móviles.
* Servicios web y APIs RESTful.
* Microservicios.

**React**

**Términos clave:**

* **Componentes**: Bloques de construcción de las aplicaciones React.
* **JSX**: Sintaxis que combina JavaScript con HTML.
* **State**: Estado interno de los componentes.
* **Props**: Propiedades pasadas a los componentes.

**Escenarios:**

* Desarrollo de interfaces de usuario interactivas.
* Aplicaciones web de una sola página (SPA).

**Node.js**

**Términos clave:**

* **Event-driven**: Modelo de programación basado en eventos.
* **Non-blocking I/O**: Operaciones de entrada/salida asíncronas.
* **NPM**: Gestor de paquetes de Node.js.

**Escenarios:**

* Servidores web y APIs.
* Aplicaciones en tiempo real (chat, colaboración en tiempo real).

**Cloud Platforms**

**AWS (Amazon Web Services)**

**Términos clave:**

* **EC2**: Máquinas virtuales en la nube.
* **S3**: Almacenamiento de objetos.
* **Lambda**: Funciones serverless.

**Escenarios:**

* Almacenamiento de datos, hospedaje de aplicaciones, servicios serverless.

**Azure (Microsoft)**

**Términos clave:**

* **Azure Virtual Machines**: Máquinas virtuales en la nube.
* **Azure Blob Storage**: Almacenamiento de objetos.
* **Azure Functions**: Funciones serverless.

**Escenarios:**

* Desarrollo y hospedaje de aplicaciones, servicios de datos y analítica.

**GCP (Google Cloud Platform)**

**Términos clave:**

* **Compute Engine**: Máquinas virtuales en la nube.
* **Cloud Storage**: Almacenamiento de objetos.
* **Cloud Functions**: Funciones serverless.

**Escenarios:**

* Hospedaje de aplicaciones, análisis de datos, machine learning.

**Containerization**

**Docker**

**Términos clave:**

* **Containers**: Entornos ligeros y aislados para aplicaciones.
* **Dockerfile**: Archivo de configuración para construir imágenes de contenedores.
* **Docker Hub**: Registro de imágenes de Docker.

**Escenarios:**

* Desarrollo, despliegue y ejecución de aplicaciones de manera consistente.

**Kubernetes**

**Términos clave:**

* **Pods**: Unidades básicas de ejecución en Kubernetes.
* **Clusters**: Conjunto de nodos que ejecutan contenedores.
* **Helm**: Herramienta para gestionar paquetes de Kubernetes.

**Escenarios:**

* Orquestación y gestión de contenedores a gran escala.

**CI/CD Pipelines**

**Términos clave:**

* **Continuous Integration (CI)**: Integración continua de cambios de código.
* **Continuous Delivery (CD)**: Despliegue continuo de aplicaciones.
* **Jenkins, GitLab CI/CD, GitHub Actions**: Herramientas populares de CI/CD.

**Escenarios:**

* Automatización del proceso de desarrollo, pruebas y despliegue de software.

**DevOps Practices**

**Términos clave:**

* **Infrastructure as Code (IaC)**: Gestión de la infraestructura mediante código.
* **Monitoring and Logging**: Supervisión y registro de aplicaciones.
* **Collaboration and Communication**: Mejora de la comunicación entre equipos de desarrollo y operaciones.

**Escenarios:**

* Mejora de la eficiencia y la colaboración en el desarrollo y operaciones de software.

**SQL**

**Términos clave:**

* **Relational Databases**: Bases de datos relacionales.
* **Queries**: Consultas para manipular datos.
* **Transactions**: Operaciones atómicas en la base de datos.

**Escenarios:**

* Almacenamiento y gestión de datos estructurados.

**NoSQL**

**Términos clave:**

* **Document Stores (e.g., MongoDB)**: Bases de datos basadas en documentos.
* **Key-Value Stores (e.g., Redis)**: Almacenamiento basado en clave-valor.
* **Columnar Stores (e.g., Cassandra)**: Bases de datos orientadas a columnas.

**Escenarios:**

* Almacenamiento de datos no estructurados o semi-estructurados, alta escalabilidad y rendimiento.