

## **Aplicaciones Distribuidas (2 semana)**

### **Objetivo:**

Evaluar la comprensión de los conceptos clave presentados en los videos de la semana 2: Middleware, Autenticación y Autorización, y Despliegue de aplicaciones distribuidas con Docker, Kubernetes y PM2

### **Instrucciones:**

Responde las siguientes preguntas de manera clara y concisa, demostrando que has comprendido los temas abordados en los videos.

### **Middleware**

¿Cuál es la función principal del middleware en un sistema distribuido?

Menciona dos tipos de middleware y explica brevemente su utilidad.

¿Qué beneficios aporta el uso de middleware en el desarrollo de aplicaciones distribuidas?

### **Autenticación y Autorización**

¿Cuál es la diferencia entre autenticación y autorización?

Describe dos métodos de autenticación comunes y sus ventajas/desventajas.

Explica brevemente qué es el control de acceso basado en roles (RBAC).

### **Despliegue de aplicaciones**

¿Qué es un contenedor Docker y cuáles son sus ventajas?

¿Cuál es la función principal de Kubernetes en el despliegue de aplicaciones?

¿Cómo PM2 facilita la gestión de aplicaciones Node.js en un entorno distribuido?

## 1. Middleware

- **¿Cuál es la función principal del middleware en un sistema distribuido?**

El middleware actúa como un intermediario que facilita la comunicación y gestión de datos entre diferentes componentes de un sistema distribuido. Se encarga de tareas como la gestión de peticiones, el manejo de respuestas, y la ejecución de funciones comunes en las aplicaciones.

- **Menciona dos tipos de middleware y explica brevemente su utilidad.**

**Middleware de Comunicación:** Facilita la comunicación entre diferentes servicios o componentes de un sistema, como RPC (Remote Procedure Call) o sistemas de mensajería (e.g., RabbitMQ).

**Middleware de Seguridad:** Se encarga de gestionar aspectos de seguridad como la autenticación y la autorización, asegurando que solo usuarios autorizados puedan acceder a ciertos recursos.

- **¿Qué beneficios aporta el uso de middleware en el desarrollo de aplicaciones distribuidas?**

Simplifica la arquitectura al manejar tareas comunes, lo que permite a los desarrolladores centrarse en la lógica de negocio.

Mejora la escalabilidad al permitir una separación clara entre los componentes de la aplicación.

Facilita la integración de nuevas tecnologías y servicios al sistema sin requerir cambios significativos en la aplicación.

## 2. Autenticación y Autorización

- **¿Cuál es la diferencia entre autenticación y autorización?**

**Autenticación** es el proceso de verificar la identidad de un usuario, es decir, confirmar que el usuario es quien dice ser.

**Autorización** es el proceso de determinar qué recursos o acciones puede realizar un usuario autenticado dentro de un sistema.

- **Describe dos métodos de autenticación comunes y sus ventajas/desventajas.**

**Contraseña:** Ventaja: Fácil de implementar y usar. Desventaja: Vulnerable a ataques de fuerza bruta y phishing.

**Autenticación de dos factores (2FA):** Ventaja: Aumenta la seguridad al requerir un segundo factor, como un código enviado al móvil. Desventaja: Puede ser incómodo para los usuarios y depender de dispositivos externos.

- **Explica brevemente qué es el control de acceso basado en roles (RBAC).**

RBAC (Role-Based Access Control) es un modelo de control de acceso donde los permisos se asignan a roles en lugar de a usuarios individuales. Los usuarios adquieren permisos en función de los roles que se les asignan, lo que facilita la gestión de permisos en sistemas grandes y complejos.

### **3. Despliegue de Aplicaciones**

- **¿Qué es un contenedor Docker y cuáles son sus ventajas?**

Un contenedor Docker es una unidad estandarizada que encapsula una aplicación y todas sus dependencias, permitiendo que se ejecute de manera consistente en cualquier entorno. Ventajas: Portabilidad, aislamiento, y eficiencia en el uso de recursos.

- **¿Cuál es la función principal de Kubernetes en el despliegue de aplicaciones?**

Kubernetes es una plataforma de orquestación que gestiona contenedores a gran escala, automatizando el despliegue, escalado, y mantenimiento de aplicaciones en entornos distribuidos.

- **¿Cómo PM2 facilita la gestión de aplicaciones Node.js en un entorno distribuido?**

PM2 es un gestor de procesos para aplicaciones Node.js que facilita la administración de múltiples instancias de una aplicación, proporciona herramientas para el monitoreo, reinicio automático en caso de fallos, y permite la carga balanceada de trabajo en entornos distribuidos.