

Cuestionario

1. **¿Cuál es uno de los principales desafíos en el proceso de construcción del sistema?**
- a) La comunicación entre diferentes sistemas de desarrollo.
 - b) La extracción de versiones específicas de componentes desde el repositorio de control de versiones.
 - c) La generación automática de rutinas de construcción.

Respuesta correcta: b)

2. **¿Cuál es una característica esencial de las plataformas de desarrollo de software?**
- a) Proporcionan un entorno de desarrollo integrado.
 - b) Facilitan la comunicación entre diferentes equipos de desarrollo.
 - c) Son exclusivamente para compartir y reutilizar código.

Respuesta correcta: a) Proporcionan un entorno de desarrollo integrado.

3. **¿Cuál es uno de los entornos que podría ser más pequeño y simple en comparación con el entorno de desarrollo?**
- a) Entorno de desarrollo.
 - b) Servidor de construcción.
 - c) Entorno objetivo.

Respuesta correcta: c) Entorno objetivo.

4. **¿Qué función desempeñan las pruebas automatizadas en la integración continua?**
- a) Aumentar la carga de trabajo de los desarrolladores.
 - b) Verificar el funcionamiento correcto del sistema sin intervención humana.
 - c) Proporcionar un complemento a las pruebas manuales.

Respuesta correcta: b) Verificar el funcionamiento correcto del sistema sin intervención humana.

5. **¿Cuál es una herramienta comúnmente utilizada para el control de versiones en proyectos de desarrollo de software?**
- a) Selenium WebDriver.
 - b) Subversión.
 - c) Eclipse.

Respuesta correcta: b) Subversión.

6. **¿Cuál es uno de los pasos recomendados para implementar la integración continua?**
- a) Ignorar la integración de cambios diarios en el equipo.
 - b) Obtener un servicio de CI que ejecute automáticamente las pruebas.
 - c) Realizar pruebas manuales de manera ocasional.

Respuesta correcta: b) Obtener un servicio de CI que ejecute automáticamente las pruebas.

7. **¿Qué implica el proceso de construcción del sistema?**
- a) Crear un entorno de desarrollo integrado.
 - b) Compilar los componentes del sistema y las librerías externas.
 - c) Desarrollar herramientas para la gestión de versiones.

Respuesta correcta: b) Compilar los componentes del sistema y las librerías externas.

8. **¿Qué herramienta se utiliza para optimizar el proceso de compilación mediante desencadenadores automáticos?**
- a) Herramientas de automatización de pruebas.
 - b) Sistemas de gestión de versiones.
 - c) Automatización de compilaciones.

Respuesta correcta: c) Automatización de compilaciones.

9. ¿Cuál es uno de los propósitos de la construcción del sistema?

- a) Compartir y reutilizar código.
- b) Generar rutinas de construcción automáticamente.**
- c) Facilitar la comunicación entre diferentes equipos de desarrollo.

Respuesta correcta: b) Generar rutinas de construcción automáticamente.

10. ¿Qué herramienta se utiliza para realizar pruebas unitarias con JUnit?

- a) Selenium WebDriver.
- b) Subversión.
- c) Eclipse.**

Respuesta correcta: c) Eclipse.

11. ¿Cuál es uno de los factores técnicos clave en la gestión de entregas de software?

- a) Cultura organizacional
- b) Infraestructura tecnológica**
- c) Estructura de equipos
- d) Gestión del cambio

Respuesta: b) Infraestructura tecnológica

12. ¿Qué tipo de pruebas ayuda a identificar y corregir errores en el software, reduciendo el tiempo y los costos asociados con el desarrollo?

- a) Pruebas de aceptación
- b) Pruebas unitarias
- c) Pruebas de integración**
- d) Pruebas de rendimiento

Respuesta: c) Pruebas de integración

13. ¿Cuál de los siguientes no es un tipo de artefacto almacenado en un repositorio de artefactos?

- a) Código fuente
- b) Documentación técnica
- c) Scripts de despliegue
- d) Licencias de software**

Respuesta: d) Licencias de software

14. ¿Qué etapa del procedimiento de entrega de software implica la implementación del software en el entorno de producción y su puesta a disposición de los usuarios finales?

- a) Planificación de la entrega
- b) Desarrollo del software
- c) Implementación de la entrega**
- d) Monitoreo y soporte

Respuesta: c) Implementación de la entrega

15. ¿Qué factor organizacional promueve prácticas ágiles y la adopción de nuevas tecnologías en la gestión de entregas de software?

- a) Cultura organizacional**
- b) Estructura de equipos
- c) Comunicación interna
- d) Gestión del cambio

Respuesta: a) Cultura organizacional

16. ¿Cuál es uno de los beneficios de utilizar un repositorio de artefactos en el desarrollo y despliegue de software?

- a) Aumenta la complejidad del proceso de despliegue
- b) Facilita la trazabilidad y gestión de versiones**

- c) Incrementa la dependencia de herramientas externas
- d) Reduce la necesidad de comunicación entre equipos

Respuesta: b) Facilita la trazabilidad y gestión de versiones

17.¿Cuál es una actividad clave en el procedimiento de entrega de software que implica ejecutar pruebas unitarias, de integración y de aceptación?

- a) Preparación de la entrega
- b) Desarrollo del software
- c) Pruebas de validación
- d) Implementación de la entrega

Respuesta: c) Pruebas de validación

18.¿Qué tipo de herramienta ayuda a mejorar la productividad, la calidad y la automatización del proceso de desarrollo de software?

- a) Herramientas de gestión de proyectos
- b) Herramientas de control de versiones
- c) Herramientas de desarrollo
- d) Herramientas de monitoreo

Respuesta: c) Herramientas de desarrollo

19.¿Qué etapa del procedimiento de entrega de software implica el monitoreo del rendimiento y la estabilidad del software desplegado?

- a) Implementación de la entrega
- b) Monitoreo y soporte
- c) Pruebas de validación
- d) Preparación de la entrega

Respuesta: b) Monitoreo y soporte

20.¿Qué tipo de estructura de equipos integra equipos de desarrollo y operaciones para mejorar la colaboración y la velocidad de entrega?

- a) Scrum
- b) Agile
- c) DevOps
- d) Kanban

Respuesta: c) DevOps

21.¿Qué es un repositorio?

- a) Un lugar donde se almacena el hardware del proyecto.
- b) Una instancia específica del software.
- c) Un lugar centralizado donde se almacena el código fuente y su historial de cambios.
- d) Un proceso de observar el rendimiento y la salud de un entorno.

Respuesta correcta: Opción C

22.¿Qué permite el control de versiones?

- a) Mantener actualizados los entornos de desarrollo.
- b) Realizar un seguimiento de los cambios en el código fuente.
- c) Automatizar tareas repetitivas en el desarrollo de software.
- d) Crear ramas independientes para el desarrollo de nuevas funcionalidades.

Respuesta correcta: Opción B

23.¿Qué son las ramas en un repositorio?

- a) Líneas de desarrollo independientes dentro de un repositorio.
- b) Unidades de cambio en el código fuente.
- c) Acciones de enviar cambios locales al repositorio central.
- d) Procesos de observar el rendimiento de un entorno.

Respuesta correcta: Opción A

24. ¿Cuál es la función de un commit?

- a) Enviar cambios locales al repositorio central.
- b) Obtener los últimos cambios del repositorio central.
- c) Registrar de forma permanente los cambios preparados en el área de preparación.
- d) Crear etiquetas que marcan puntos específicos en la historia del repositorio.

Respuesta correcta: Opción C

25. ¿Qué implica el provisioning en gestión de entornos?

- a) Crear y configurar un entorno.
- b) Ajustar parámetros de un entorno.
- c) Observar el rendimiento y la salud de un entorno.
- d) Automatizar tareas repetitivas en la gestión de entornos.

Respuesta correcta: Opción A

26. ¿Qué es la configuración en gestión de entornos?

- a) Proceso de ajustar parámetros de un entorno.
- b) Observar el rendimiento y la salud de un entorno.
- c) Crear y configurar un entorno.
- d) Instalar el software necesario en un entorno.

Respuesta correcta: Opción A

27. ¿Cuál es la función del monitoreo en gestión de entornos?

- a) Crear y configurar un entorno.
- b) Ajustar parámetros de un entorno.
- c) Observar el rendimiento y la salud de un entorno.
- d) Automatizar tareas repetitivas en la gestión de entornos.

Respuesta correcta: Opción C

28. ¿Qué implica la automatización en gestión de entornos?

- a) Ajustar parámetros de un entorno.
- b) Observar el rendimiento y la salud de un entorno.
- c) Automatizar tareas repetitivas de forma automática.
- d) Instalar el software necesario en un entorno.

Respuesta correcta: Opción C

29. ¿Qué operaciones son fundamentales para iniciar un proyecto en un sistema de control de versiones?

- a) Añadir archivos nuevos.
- b) Modificar archivos existentes.
- c) Generación de un nuevo repositorio y ramificación.
- d) Confirmación de cambios (Commit) y revisión de cambios.

Respuesta correcta: Opción C

30. ¿Cuál es la función de la fusión de ramas en la gestión de código fuente?

- a) Generar una nueva línea de desarrollo paralela.
- b) Cambiar entre diferentes ramas existentes.
- c) Solucionar los conflictos que pueden surgir al fusionar ramas.
- d) Combinar los cambios realizados en una rama con otra rama de destino.

Respuesta correcta: Opción D

31. ¿Qué es la gestión de código fuente?

- a. Un proceso para almacenar físicamente el código en un servidor
- b. Una práctica para controlar cambios en el código y facilitar la colaboración

- c. Un método para compilar automáticamente el código antes de su despliegue
- d. Una herramienta para realizar pruebas de rendimiento en el código

Respuesta: b) Una práctica para controlar cambios en el código y facilitar la colaboración

32.¿Qué herramienta es comúnmente utilizada para la gestión de código fuente?

- a. Docker
- b. Jenkins
- c. Git
- d. Kubernetes

Respuesta: c) Git

33.¿Qué es la gestión de entornos?

- a. Un proceso para controlar los cambios en el código en diferentes entornos
- b. Una herramienta para crear entornos virtuales aislados para el desarrollo
- c. Un método para automatizar el despliegue de aplicaciones en la nube
- d. Una práctica para mantener la coherencia entre los diferentes entornos de desarrollo, prueba y producción

Respuesta: d) Una práctica para mantener la coherencia entre los diferentes entornos de desarrollo, prueba y producción

34.¿Cuál es el propósito principal de los contenedores en la infraestructura como código?

- a. Virtualizar servidores físicos
- b. Aislar y empaquetar aplicaciones y sus dependencias
- c. Automatizar el aprovisionamiento de recursos de red
- d. Simplificar la implementación de bases de datos relacionales

Respuesta: b) Aislar y empaquetar aplicaciones y sus dependencias

35.¿Cuál de las siguientes es una tecnología comúnmente utilizada para la orquestación de contenedores?

- a. Docker Compose
- b. Ansible
- c. Chef
- d. Terraform

Respuesta: a) Docker Compose

36.¿Cuál es la función principal de la orquestación de contenedores?

- a. Controlar y coordinar la implementación y gestión de contenedores en clústeres
- b. Optimizar el rendimiento de las aplicaciones dentro de los contenedores
- c. Facilitar la comunicación entre diferentes contenedores en una misma máquina
- d. Gestionar el almacenamiento persistente de datos en contenedores

Respuesta: a) Controlar y coordinar la implementación y gestión de contenedores en clústeres

37.¿Qué ventaja ofrece la orquestación de contenedores en comparación con el manejo manual?

- a. Mayor flexibilidad en la configuración de los contenedores
- b. Automatización de tareas repetitivas y escalamiento dinámico
- c. Reducción del tiempo de desarrollo de aplicaciones
- d. Mejora en la seguridad de los contenedores

Respuesta: b) Automatización de tareas repetitivas y escalamiento dinámico

38.¿Qué componente es esencial en la orquestación de contenedores para gestionar el estado de la aplicación?

- a. Load Balancer
- b. Scheduler**
- c. Container Registry
- d. Monitor

Respuesta: b) Scheduler

39.¿Qué tecnología permite la orquestación de contenedores en múltiples nodos?

- a. Kubernetes**
- b. Docker Swarm
- c. Amazon ECS
- d. OpenShift

Respuesta: a) Kubernetes

40.¿Cuál es el papel principal de Kubernetes en la orquestación de contenedores?

- a. Aprovisionar recursos de red para los contenedores
- b. Coordinar la implementación, escala y gestión de contenedores en clústeres**
- c. Crear contenedores a partir de imágenes de aplicaciones
- d. Proporcionar una interfaz de usuario gráfica para gestionar contenedores

Respuesta: b) Coordinar la implementación, escala y gestión de contenedores en clústeres

41.¿Qué característica principal diferencia a RKT de Docker?

- a) Portabilidad de contenedores en diferentes entornos
- b) Aislamiento de aplicaciones en contenedores
- c) Enfoque modular y descentralizado**
- d) Implementación de la especificación OCI

42.¿Cuál de las siguientes herramientas NO está diseñada específicamente para la ejecución de contenedores?

- a. Docker
- b. RunC
- c. Containerd
- d. RKT**

43.¿Qué tecnología es la base fundamental del aislamiento de contenedores?

- a. Virtualización completa de sistemas operativos
- b. Espacios de nombres y grupos de control de Linux**
- c. Máquinas virtuales
- d. Hipervisores

44.¿Cuál de las siguientes etapas NO forma parte del ciclo de vida de las aplicaciones con Docker?

- a. Creación de imágenes de contenedores
- b. Compilación del código fuente de la aplicación**
- c. Distribución y despliegue de imágenes
- d. Ejecución de contenedores

45.¿Qué característica de Docker lo convierte en una herramienta atractiva para desarrolladores?

- a. Compatibilidad con varios formatos de imagen de contenedor
- b. Enfoque modular y descentralizado
- c. Sistema de imágenes y registros para almacenar y distribuir aplicaciones**
- d. Cumplimiento estricto con la especificación OCI

46.¿Cuál de las siguientes afirmaciones sobre los contenedores es INCORRECTA?

- a. Son una forma ligera y eficiente de virtualizar aplicaciones.
- b. Requieren de un sistema operativo completo para funcionar.**
- c. Permiten compartir recursos de manera eficiente entre diferentes contenedores.

d. Facilitan la portabilidad de las aplicaciones a diferentes entornos.

47.¿Cuál de las siguientes afirmaciones es INCORRECTA sobre RunC?

- a. Implementa la especificación OCI para la ejecución de contenedores Linux.
- b. Permite iniciar y ejecutar contenedores a partir de imágenes compatibles con OCI.
- c. Fue desarrollado originalmente como parte del proyecto Docker.
- d. Es una herramienta gráfica que facilita la gestión de contenedores.

48.¿Qué permite a los desarrolladores concentrarse en la lógica de negocio gracias a los contenedores?

- a. La abstracción de la capa de infraestructura
- b. La facilidad de distribución y despliegue de aplicaciones
- c. El aislamiento de las aplicaciones entre sí
- d. Todas las opciones anteriores

49.¿Cuál de las siguientes características NO es una ventaja de utilizar contenedores para la implementación de aplicaciones?

- a. Aislamiento de aplicaciones y sus dependencias
- b. Reducción de la complejidad de la gestión de infraestructura
- c. Mayor consumo de recursos en comparación con las máquinas virtuales
- d. Portabilidad de aplicaciones a diferentes entornos

50.¿Qué herramienta de código abierto se utiliza para orquestar la implementación y gestión de aplicaciones en contenedores a gran escala?

- a. Docker Swarm
- b. Kubernetes
- c. Containerd
- d. Podman

51.¿Cuál es el propósito principal de la gestión del código fuente?

- a. Mantener un registro de los cambios realizados en el código base.
- b. Administrar los recursos de hardware en un entorno de desarrollo.
- c. Facilitar la interacción entre equipos de ventas y marketing.
- d. Optimizar el rendimiento de la red de una empresa.

Respuesta: A) Mantener un registro de los cambios realizados en el código base.

52.¿Qué aspectos incluye la gestión de entornos en el desarrollo de software?

- a. Administración de recursos humanos.
- b. Configuración y control de los diferentes entornos de ejecución del software.
- c. Gestión de la publicidad en línea.
- d. Desarrollo de estrategias de marketing digital.

Respuesta: B) Configuración y control de los diferentes entornos de ejecución del software.

53.¿Qué papel desempeñan los equipos de desarrollo en la gestión de código fuente y entornos?

- a. Supervisar la gestión financiera del proyecto.
- b. Realizar análisis de mercado para el software desarrollado.
- c. Escribir, mantener y mejorar el código base del proyecto.
- d. Administrar las redes informáticas de la empresa.

Respuesta: C) Escribir, mantener y mejorar el código base del proyecto.

54.¿Cuál es una práctica recomendada para la gestión efectiva del código fuente?

- a. Hacer pocas confirmaciones para reducir el historial de cambios.
- b. No revisar los cambios antes de confirmarlos.
- c. Utilizar sistemas de control de versiones para la gestión del código fuente.
- d. No documentar los procedimientos realizados.

Respuesta: C) Utilizar sistemas de control de versiones para la gestión del código fuente.

55.¿Qué aspecto es fundamental para la gestión de entornos según el texto?

- a. Gestión de la cadena de suministro.
- b. Resolución de problemas.**
- c. Administración de recursos financieros.
- d. Desarrollo de productos de software.

Respuesta: B) Resolución de problemas.

56.¿Qué aspecto humano es crucial para la gestión efectiva del código fuente y los entornos de desarrollo?

- a) Comunicación clara y colaboración.**
- b) Mantener un ambiente competitivo entre los equipos.
- c) Restringir el acceso a la información entre los equipos.
- d) Minimizar la interacción entre los miembros del equipo

Respuesta: A) Comunicación clara y colaboración.

57.¿Qué función desempeña el equipo de operaciones en la gestión de entornos de desarrollo?

- a) Escribir y mantener el código base del proyecto.
- b) Administrar los entornos de desarrollo, pruebas y producción.**
- c) Realizar análisis de mercado para el software desarrollado.
- d) Desarrollar estrategias de marketing digital para el producto.

Respuesta: B) Administrar los entornos de desarrollo, pruebas y producción.

58.¿Cuál de las siguientes prácticas NO es recomendada para la gestión efectiva del código fuente?

- (a) Realizar confirmaciones frecuentes para registrar los cambios.
- (b) No revisar los cambios antes de confirmarlos.**
- (c) Utilizar sistemas de control de versiones para la gestión del código.
- (d) Elaborar documentación sobre los procedimientos realizados.

Respuesta B) No revisar los cambios antes de confirmarlos.

59.¿Cuál es uno de los propósitos de realizar ramificaciones en la gestión del código fuente?

- (a) Reducir la flexibilidad del equipo de desarrollo.
- (b) Incrementar los conflictos entre los miembros del equipo.
- (c) Permitir que los desarrolladores trabajen en paralelo en diferentes líneas de desarrollo.**
- (d) Aumentar la carga de trabajo de los equipos.

Respuesta: C) Permitir que los desarrolladores trabajen en paralelo en diferentes líneas de desarrollo.

60.¿Por qué es importante realizar capacitaciones sobre nuevas metodologías y tecnologías en la gestión de código fuente y entornos?

- (a) Para limitar el desarrollo profesional de los miembros del equipo.
- (b) Para asegurar que los equipos permanezcan estancos en sus conocimientos.
- (c) Para fomentar la mejora continua y mantenerse actualizado con las últimas prácticas.**
- (d) Para evitar la colaboración entre equipos.

Respuesta: C) Para fomentar la mejora continua y mantenerse actualizado con las ultimas prácticas.