# Cuestionario

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

Respuesta correcta: b)

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

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

1. **¿Cuál es uno de los entornos que podría ser más pequeño y simple en comparación con el entorno de desarrollo?**
2. Entorno de desarrollo.
3. Servidor de construcción.
4. Entorno objetivo.

Respuesta correcta: c) Entorno objetivo.

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

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

1. **¿Cuál es una herramienta comúnmente utilizada para el control de versiones en proyectos de desarrollo de software?**
2. Selenium WebDriver.
3. Subversión.
4. Eclipse.

Respuesta correcta: b) Subversión.

1. **¿Cuál es uno de los pasos recomendados para implementar la integración continua?**
2. Ignorar la integración de cambios diarios en el equipo.
3. Obtener un servicio de CI que ejecute automáticamente las pruebas.
4. Realizar pruebas manuales de manera ocasional.

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

1. **¿Qué implica el proceso de construcción del sistema?**
2. Crear un entorno de desarrollo integrado.
3. Compilar los componentes del sistema y las librerías externas.
4. Desarrollar herramientas para la gestión de versiones.

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

1. **¿Qué herramienta se utiliza para optimizar el proceso de compilación mediante desencadenadores automáticos?**
2. Herramientas de automatización de pruebas.
3. Sistemas de gestión de versiones.
4. Automatización de compilaciones.

Respuesta correcta: c) Automatización de compilaciones.

1. **¿Cuál es uno de los propósitos de la construcción del sistema?**
2. Compartir y reutilizar código.
3. Generar rutinas de construcción automáticamente.
4. Facilitar la comunicación entre diferentes equipos de desarrollo.

Respuesta correcta: b) Generar rutinas de construcción automáticamente.

1. **¿Qué herramienta se utiliza para realizar pruebas unitarias con JUnit?**
2. Selenium WebDriver.
3. Subversión.
4. Eclipse.

Respuesta correcta: c) Eclipse.

1. **¿Cuál es uno de los factores técnicos clave en la gestión de entregas de software?**
2. Cultura organizacional
3. Infraestructura tecnológica
4. Estructura de equipos
5. Gestión del cambio

Respuesta: b) Infraestructura tecnológica

1. **¿Qué tipo de pruebas ayuda a identificar y corregir errores en el software, reduciendo el tiempo y los costos asociados con el desarrollo?**
2. Pruebas de aceptación
3. Pruebas unitarias
4. Pruebas de integración
5. Pruebas de rendimiento

Respuesta: c) Pruebas de integración

1. **¿Cuál de los siguientes no es un tipo de artefacto almacenado en un repositorio de artefactos?**
2. Código fuente
3. Documentación técnica
4. Scripts de despliegue
5. Licencias de software

Respuesta: d) Licencias de software

1. **¿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?**
2. Planificación de la entrega
3. Desarrollo del software
4. Implementación de la entrega
5. Monitoreo y soporte

Respuesta: c) Implementación de la entrega

1. **¿Qué factor organizacional promueve prácticas ágiles y la adopción de nuevas tecnologías en la gestión de entregas de software?**
2. Cultura organizacional
3. Estructura de equipos
4. Comunicación interna
5. Gestión del cambio

Respuesta: a) Cultura organizacional

1. **¿Cuál es uno de los beneficios de utilizar un repositorio de artefactos en el desarrollo y despliegue de software?**
2. Aumenta la complejidad del proceso de despliegue
3. Facilita la trazabilidad y gestión de versiones
4. Incrementa la dependencia de herramientas externas
5. Reduce la necesidad de comunicación entre equipos

Respuesta: b) Facilita la trazabilidad y gestión de versiones

1. **¿Cuál es una actividad clave en el procedimiento de entrega de software que implica ejecutar pruebas unitarias, de integración y de aceptación?**
2. Preparación de la entrega
3. Desarrollo del software
4. Pruebas de validación
5. Implementación de la entrega

Respuesta: c) Pruebas de validación

1. **¿Qué tipo de herramienta ayuda a mejorar la productividad, la calidad y la automatización del proceso de desarrollo de software?**
2. Herramientas de gestión de proyectos
3. Herramientas de control de versiones
4. Herramientas de desarrollo
5. Herramientas de monitoreo

Respuesta: c) Herramientas de desarrollo

1. **¿Qué etapa del procedimiento de entrega de software implica el monitoreo del rendimiento y la estabilidad del software desplegado?**
2. Implementación de la entrega
3. Monitoreo y soporte
4. Pruebas de validación
5. Preparación de la entrega

Respuesta: b) Monitoreo y soporte

1. **¿Qué tipo de estructura de equipos integra equipos de desarrollo y operaciones para mejorar la colaboración y la velocidad de entrega?**
2. Scrum
3. Agile
4. DevOps
5. Kanban

Respuesta: c) DevOps

1. **¿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

1. **¿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

1. **¿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

1. **¿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

1. **¿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

1. **¿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

1. **¿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

1. **¿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

1. **¿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

1. **¿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

1. **¿Qué es la gestión de código fuente?**
   1. Un proceso para almacenar físicamente el código en un servidor
   2. Una práctica para controlar cambios en el código y facilitar la colaboración
   3. Un método para compilar automáticamente el código antes de su despliegue
   4. Una herramienta para realizar pruebas de rendimiento en el código

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

1. **¿Qué herramienta es comúnmente utilizada para la gestión de código fuente?**
   1. Docker
   2. Jenkins
   3. Git
   4. Kubernetes

Respuesta: c) Git

1. **¿Qué es la gestión de entornos?**
   1. Un proceso para controlar los cambios en el código en diferentes entornos
   2. Una herramienta para crear entornos virtuales aislados para el desarrollo
   3. Un método para automatizar el despliegue de aplicaciones en la nube
   4. 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

1. **¿Cuál es el propósito principal de los contenedores en la infraestructura como código?**
   1. Virtualizar servidores físicos
   2. Aislar y empaquetar aplicaciones y sus dependencias
   3. Automatizar el aprovisionamiento de recursos de red
   4. Simplificar la implementación de bases de datos relacionales

Respuesta: b) Aislar y empaquetar aplicaciones y sus dependencias

1. **¿Cuál de las siguientes es una tecnología comúnmente utilizada para la orquestación de contenedores?**
   1. Docker Compose
   2. Ansible
   3. Chef
   4. Terraform

Respuesta: a) Docker Compose

1. **¿Cuál es la función principal de la orquestación de contenedores?**
   1. Controlar y coordinar la implementación y gestión de contenedores en clústeres
   2. Optimizar el rendimiento de las aplicaciones dentro de los contenedores
   3. Facilitar la comunicación entre diferentes contenedores en una misma máquina
   4. Gestionar el almacenamiento persistente de datos en contenedores

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

1. **¿Qué ventaja ofrece la orquestación de contenedores en comparación con el manejo manual?**
   1. Mayor flexibilidad en la configuración de los contenedores
   2. Automatización de tareas repetitivas y escalamiento dinámico
   3. Reducción del tiempo de desarrollo de aplicaciones
   4. Mejora en la seguridad de los contenedores

Respuesta: b) Automatización de tareas repetitivas y escalamiento dinámico

1. **¿Qué componente es esencial en la orquestación de contenedores para gestionar el estado de la aplicación?**
   1. Load Balancer
   2. Scheduler
   3. Container Registry
   4. Monitor

Respuesta: b) Scheduler

1. **¿Qué tecnología permite la orquestación de contenedores en múltiples nodos?**
   1. Kubernetes
   2. Docker Swarm
   3. Amazon ECS
   4. OpenShift

Respuesta: a) Kubernetes

1. **¿Cuál es el papel principal de Kubernetes en la orquestación de contenedores?**
   1. Aprovisionar recursos de red para los contenedores
   2. Coordinar la implementación, escala y gestión de contenedores en clústeres
   3. Crear contenedores a partir de imágenes de aplicaciones
   4. Proporcionar una interfaz de usuario gráfica para gestionar contenedores

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

1. **¿Qué característica principal diferencia a RKT de Docker?**
2. Portabilidad de contenedores en diferentes entornos
3. Aislamiento de aplicaciones en contenedores
4. Enfoque modular y descentralizado
5. Implementación de la especificación OCI
6. **¿Cuál de las siguientes herramientas NO está diseñada específicamente para la ejecución de contenedores?**
   1. Docker
   2. RunC
   3. Containerd
   4. RKT
7. **¿Qué tecnología es la base fundamental del aislamiento de contenedores?**
   1. Virtualización completa de sistemas operativos
   2. Espacios de nombres y grupos de control de Linux
   3. Máquinas virtuales
   4. Hipervisores
8. **¿Cuál de las siguientes etapas NO forma parte del ciclo de vida de las aplicaciones con Docker?**
   1. Creación de imágenes de contenedores
   2. Compilación del código fuente de la aplicación
   3. Distribución y despliegue de imágenes
   4. Ejecución de contenedores
9. **¿Qué característica de Docker lo convierte en una herramienta atractiva para desarrolladores?**
   1. Compatibilidad con varios formatos de imagen de contenedor
   2. Enfoque modular y descentralizado
   3. Sistema de imágenes y registros para almacenar y distribuir aplicaciones
   4. Cumplimiento estricto con la especificación OCI
10. **¿Cuál de las siguientes afirmaciones sobre los contenedores es INCORRECTA?**
    1. Son una forma ligera y eficiente de virtualizar aplicaciones.
    2. Requieren de un sistema operativo completo para funcionar.
    3. Permiten compartir recursos de manera eficiente entre diferentes contenedores.
    4. Facilitan la portabilidad de las aplicaciones a diferentes entornos.
11. **¿Cuál de las siguientes afirmaciones es INCORRECTA sobre RunC?**
    1. Implementa la especificación OCI para la ejecución de contenedores Linux.
    2. Permite iniciar y ejecutar contenedores a partir de imágenes compatibles con OCI.
    3. Fue desarrollado originalmente como parte del proyecto Docker.
    4. Es una herramienta gráfica que facilita la gestión de contenedores.
12. **¿Qué permite a los desarrolladores concentrarse en la lógica de negocio gracias a los contenedores?**
    1. La abstracción de la capa de infraestructura
    2. La facilidad de distribución y despliegue de aplicaciones
    3. El aislamiento de las aplicaciones entre sí
    4. Todas las opciones anteriores
13. **¿Cuál de las siguientes características NO es una ventaja de utilizar contenedores para la implementación de aplicaciones?**
    1. Aislamiento de aplicaciones y sus dependencias
    2. Reducción de la complejidad de la gestión de infraestructura
    3. Mayor consumo de recursos en comparación con las máquinas virtuales
    4. Portabilidad de aplicaciones a diferentes entornos
14. **¿Qué herramienta de código abierto se utiliza para orquestar la implementación y gestión de aplicaciones en contenedores a gran escala?**
    1. Docker Swarm
    2. Kubernetes
    3. Containerd
    4. Podman
15. **¿Cuál es el propósito principal de la gestión del código fuente?**
    1. Mantener un registro de los cambios realizados en el código base.
    2. Administrar los recursos de hardware en un entorno de desarrollo.
    3. Facilitar la interacción entre equipos de ventas y marketing.
    4. Optimizar el rendimiento de la red de una empresa.

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

1. **¿Qué aspectos incluye la gestión de entornos en el desarrollo de software?**
   1. Administración de recursos humanos.
   2. Configuración y control de los diferentes entornos de ejecución del software.
   3. Gestión de la publicidad en línea.
   4. Desarrollo de estrategias de marketing digital.

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

1. **¿Qué papel desempeñan los equipos de desarrollo en la gestión de código fuente y entornos?**
   1. Supervisar la gestión financiera del proyecto.
   2. Realizar análisis de mercado para el software desarrollado.
   3. Escribir, mantener y mejorar el código base del proyecto.
   4. Administrar las redes informáticas de la empresa.

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

1. **¿Cuál es una práctica recomendada para la gestión efectiva del código fuente?**
   1. Hacer pocas confirmaciones para reducir el historial de cambios.
   2. No revisar los cambios antes de confirmarlos.
   3. Utilizar sistemas de control de versiones para la gestión del código fuente.
   4. No documentar los procedimientos realizados.

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

1. **¿Qué aspecto es fundamental para la gestión de entornos según el texto?**
   1. Gestión de la cadena de suministro.
   2. Resolución de problemas.
   3. Administración de recursos financieros.
   4. Desarrollo de productos de software.

**Respuesta: B) Resolución de problemas.**

1. **¿Qué aspecto humano es crucial para la gestión efectiva del código fuente y los entornos de desarrollo?**
   1. Comunicación clara y colaboración.
   2. Mantener un ambiente competitivo entre los equipos.
   3. Restringir el acceso a la información entre los equipos.
   4. Minimizar la interacción entre los miembros del equipo

#### Respuesta: A) Comunicación clara y colaboración.

1. **¿Qué función desempeña el equipo de operaciones en la gestión de entornos de desarrollo?**
   1. Escribir y mantener el código base del proyecto.
   2. Administrar los entornos de desarrollo, pruebas y producción.
   3. Realizar análisis de mercado para el software desarrollado.
   4. Desarrollar estrategias de marketing digital para el producto.

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

1. **¿Cuál de las siguientes prácticas NO es recomendada para la gestión efectiva del código fuente?**
2. Realizar confirmaciones frecuentes para registrar los cambios.
3. No revisar los cambios antes de confirmarlos.
4. Utilizar sistemas de control de versiones para la gestión del código.
5. Elaborar documentación sobre los procedimientos realizados.

Respuesta B) No revisar los cambios antes de confirmarlos.

1. **¿Cuál es uno de los propósitos de realizar ramificaciones en la gestión del código fuente?**
2. Reducir la flexibilidad del equipo de desarrollo.
3. Incrementar los conflictos entre los miembros del equipo.
4. Permitir que los desarrolladores trabajen en paralelo en diferentes líneas de desarrollo.
5. Aumentar la carga de trabajo de los equipos.

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

1. **¿Por qué es importante realizar capacitaciones sobre nuevas metodologías y tecnologías en la gestión de código fuente y entornos?**
2. Para limitar el desarrollo profesional de los miembros del equipo.
3. Para asegurar que los equipos permanezcan estancos en sus conocimientos.
4. Para fomentar la mejora continua y mantenerse actualizado con las últimas prácticas.
5. Para evitar la colaboración entre equipos.

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