

## Cuestionario Spring Boot Curso Aprendiz

(solo marcar una opción)

### 1. ¿Qué es un framework?

- A) Un entorno de ejecución.
- B) Un conjunto de herramientas de desarrollo de bases de datos.
- C) Una estructura de software que facilita el desarrollo de aplicaciones.
- D) Una herramienta de despliegue de aplicaciones.

### 2. ¿Cuál es la principal diferencia entre Spring Framework y Spring Boot?

- A) Spring Boot incluye herramientas de ORM, mientras que Spring Framework no.
- B) Spring Boot simplifica la configuración y el despliegue de aplicaciones Spring.
- C) Spring Framework es más liviano que Spring Boot.
- D) No hay diferencias, ambos son equivalentes.

### 3. ¿Cómo puedes personalizar consultas derivadas en un JpaRepository en Spring Data?

- A) Solo se pueden utilizar las consultas automáticas generadas a partir de los nombres de los métodos.
- B) Usando métodos dinámicos y las clases Specification o Criteria para consultas complejas.
- C) Extendiendo JpaRepository y definiendo métodos personalizados en una interfaz.
- D) Usando la anotación @Query para definir consultas personalizadas utilizando JPQL o SQL nativo.

### 4. ¿Qué es Hibernate?

- A) Una herramienta para autenticar usuarios.
- B) Un ORM que implementa JPA y permite gestionar persistencia de datos.
- C) Un framework para crear APIs REST.
- D) Una biblioteca para diseñar interfaces gráficas.

### 5. ¿Qué significa “Eager” en el contexto de mapeo de relaciones de Hibernate?

- A) Que la relación se carga bajo demanda.
- B) Que la relación no se carga automáticamente.
- C) Que la relación se carga de inmediato al recuperar la entidad.
- D) Que la relación se carga en caché.

### 6. ¿Qué funcionalidad principal proporciona JpaRepository en Spring Data?

- A) Autenticación de usuarios.
- B) Gestión de sesiones.
- C) Operaciones CRUD predeterminadas y soporte de paginación y filtrado.
- D) Generación de entidades.

**7. En una aplicación Spring Boot que aplica una arquitectura por capas, ¿dónde deberías colocar la lógica de negocio?**

- A) En el controlador, junto con la lógica de manejo de peticiones.
- B) En las entidades de JPA, junto con las anotaciones de mapeo.
- C) En un servicio separado, idealmente una clase anotada con `@Service`.
- D) En la vista, para que los datos estén disponibles en la interfaz.

**8. ¿Qué problemas podría causar el uso de `@RestController` en lugar de `@Controller` en una aplicación Spring Boot?**

- A) La aplicación no podrá conectarse a la base de datos.
- B) `@RestController` consume más memoria y CPU al hacer mapeos de rutas.
- C) La aplicación devolverá JSON en lugar de vistas, lo que podría no ser adecuado para aplicaciones con interfaz de usuario web.
- D) `@RestController` no puede manejar errores, a diferencia de `@Controller`.

**9. En Spring Security, ¿cuál es el propósito de un filtro JWT?**

- A) Realizar la autenticación de los usuarios por token y manejar su validación antes de pasar a los controladores.
- B) Generar tokens JWT automáticamente para cada usuario autenticado.
- C) Guardar la sesión del usuario en la base de datos.
- D) Validar la contraseña del usuario

**10. ¿Qué es un Record en Java, y cuál es su ventaja en aplicaciones Spring Boot?**

- A) Un tipo de entidad para persistir datos en bases de datos.
- B) Una clase inmutable y concisa para almacenar datos, introducida en Java 14.
- C) Una anotación que permite realizar validaciones automáticas.
- D) Una clase abstracta en Spring para implementar lógica de negocio.

**11. ¿Cuál es la ventaja de utilizar un DTO (Data Transfer Object) en aplicaciones Spring Boot?**

- A) Permite almacenar los datos en la base de datos directamente.
- B) Facilita el intercambio de datos entre capas de la aplicación sin exponer las entidades de persistencia.
- C) Permite establecer relaciones One-to-Many entre tablas.
- D) Se utiliza exclusivamente para configurar filtros de seguridad.

**12. ¿Qué método de validación usa `@Valid` en Spring Boot y para qué se utiliza?**

- A) Para comprobar la autenticación del usuario.
- B) Para validar datos de entrada en controladores y servicios usando Java Bean Validation.
- C) Para cargar configuraciones específicas de la base de datos.
- D) Para definir rutas en controladores RESTful.

**13. ¿Qué función cumple @ManyToMany y cómo se representa en una base de datos?**

- A) Mapea una relación donde una entidad está relacionada con una sola instancia de otra entidad.
- B) Mapea una relación en la que varias instancias de una entidad pueden estar relacionadas con varias instancias de otra entidad, generalmente mediante una tabla intermedia.
- C) Mapea una relación en la que las entidades están relacionadas de forma unidireccional.
- D) No tiene uso en aplicaciones Spring Boot.

**14. ¿Cuál de las siguientes opciones define mejor una consulta JPQL en JPA?**

- A) Una consulta orientada a SQL que solo se puede usar con bases de datos PostgreSQL.
- B) Una consulta que utiliza el nombre de los atributos de las entidades en lugar de nombres de columnas de bases de datos.
- C) Una consulta que se utiliza exclusivamente para la búsqueda de entidades relacionadas.
- D) Una consulta orientada a SQL, pero que permite recuperar resultados en JSON.

**15. En Spring Security, ¿cuál es la diferencia entre autenticación y autorización?**

- A) Autenticación verifica la identidad del usuario, mientras que autorización controla sus permisos dentro de la aplicación.
- B) Autenticación asegura que el usuario tenga permisos, mientras que autorización verifica su contraseña.
- C) Ambos son sinónimos y se usan de manera intercambiable.
- D) Autorización define el rol del usuario y autenticación define el token de acceso.

**16. ¿Qué es un filtro en el contexto de una API RESTful con Spring Boot y qué función cumple?**

- A) Un componente que intercepta y modifica las solicitudes y respuestas HTTP en una aplicación.
- B) Una clase que define métodos CRUD.
- C) Una capa de acceso a datos en Spring.
- D) Un sistema de validación automática para datos de usuario.

**17. ¿Cuál es el propósito de @ControllerAdvice en Spring Boot?**

- A) Configurar acceso a bases de datos en todos los controladores.
- B) Definir un controlador global para manejar excepciones y aplicar lógica compartida en todos los controladores de la aplicación.
- C) Configurar filtros de seguridad para controladores específicos.
- D) Aplicar transacciones en métodos específicos dentro de los controladores.

**18. En una aplicación Spring Boot, ¿qué sucede si lanzas una excepción RuntimeException y no hay ningún @ExceptionHandler configurado para manejarla?**

- A) La aplicación la captura automáticamente y devuelve un código 200.
- B) La excepción no será capturada, provocando que el servidor muestre una respuesta 500 con un mensaje de error genérico.
- C) La excepción será ignorada y el programa continuará ejecutándose.
- D) Se capturará y devolverá un código 404

**19. ¿Cuál es la diferencia entre manejar excepciones usando @ControllerAdvice y definir las directamente en un controlador?**

- A) @ControllerAdvice aplica el manejo de excepciones globalmente, mientras que try-catch dentro de un controlador solo afecta a ese controlador.
- B) @ControllerAdvice solo maneja excepciones críticas, mientras que try-catch en un controlador maneja todas las excepciones.
- C) No hay ninguna diferencia.
- D) @ControllerAdvice no permite la personalización de mensajes de error

**20. ¿Cuál es el propósito de @PathVariable y @RequestParam en Spring Boot, y cómo se diferencian?**

- A) Ambos se usan para enviar datos en el cuerpo de la solicitud HTTP.
- B) @PathVariable extrae valores de la URL y @RequestParam toma datos de los parámetros de consulta en la URL.
- C) @RequestParam se usa en servicios SOAP y @PathVariable en servicios REST.
- D) Ambos son métodos para extraer datos JSON de la solicitud.