1. **Indique qué desventaja tiene una arquitectura con 2 capas con clientes de escritorio (capa 1: Interfaz gráfica de escritorio y capa Modelo; capa 2: servidor de Base de Datos) frente a una arquitectura con 3 capas, también con clientes de escritorio (capa 1: Interfaz gráfica de escritorio y capa Acceso al Servicio; capa 2: capa Servicios y capa Modelo; capa 3: servidor de Base de Datos):**

   a) Es menos eficiente porque hay invocaciones remotas.
   b) No permite cambiar la Base de Datos utilizada sin modificar la capa Interfaz gráfica de escritorio.
   c) No permite hacer cambios en la capa Modelo sin necesidad de reinstalar nada en las máquinas cliente, mientras que la arquitectura en 3 capas sí lo permite.
   d) No permite hacer cambios en la capa Interfaz gráfica de escritorio sin necesidad de reinstalar nada en las máquinas cliente, mientras que la arquitectura en 3 capas sí lo permite.

2. **Indique en qué capas ha utilizado el API JDBC durante el desarrollo de su práctica:**

   a) En la capa Lógica de Negocio.
   b) En la capa Acceso a Datos.
   c) En la capa Servicios.
   d) a) y b) son correctas.

3. **¿Qué puede decir con respecto a este código?**

```
@Override
public ... create(Connection connection, ...) {

    String queryString = ...

    try (PreparedStatement preparedStatement = connection.prepareStatement(
            queryString, Statement.RETURN_GENERATED_KEYS)) {

       ...

       return ...
    } catch (...) {
        ...
    }

}
```

   a) Corresponde a la capa Acceso a Datos.
   b) Corresponde a la capa Lógica de Negocio.
   c) Corresponde a la capa Servicios REST.
   d) Corresponde a la capa Acceso a Servicios.

4. **Indique la afirmación correcta:**

   a) Cuando la conexión a la base de datos se obtiene a partir de un `DataSource`, la implementación de un caso de uso en la capa Modelo no necesita invocar a `Connection.close`, ni implícitamente (try-with-resources) ni explícitamente.
   b) Cuando se invoca a `close` sobre una conexión que se obtuvo de un pool de conexiones, la conexión contra la base de datos no se cierra.
   c) Una vez obtenida una conexión a la base de datos (desprecie este tiempo), si se lanza una consulta, es más rápida su ejecución si la conexión se obtuvo de una invocación a `DriverManager.getConnection` que si se obtuvo mediante la invocación del método `getConnection` de un `DataSource`.
   d) Todas las implementaciones de `DataSource` tienen la obligación de usar la estrategia de pool de conexiones.

5. En base al enfoque aconsejado en la asignatura para la realización de pruebas de integración, suponiendo que un caso de prueba necesite insertar datos en la Base de Datos antes de invocar al caso de uso a probar (e.g. para probar un caso de uso que haga una búsqueda es necesario que en la Base de Datos haya datos sobre los que poder hacer la búsqueda), indique cómo debe crearlos:

a) Siempre debe utilizar los DAOs que ofrezcan las operaciones pertinentes.
b) Si existen servicios de la capa Modelo que ofrezcan métodos adecuados para crear esos datos, debe utilizarlos.
c) Debe ejecutar un script SQL que tenga las sentencias necesarias para insertar los datos.
d) Debe ejecutar directamente las sentencias SQL necesarias, haciendo uso del API JDBC.

6. Dado el siguiente documento JSON:

```
[{ "title": "Learning Java",
   "formats": ["PDF", "EPUB"],
   "price": 1
}]
```

Indique cuál de las siguientes afirmaciones es correcta:
a) Se trata de un array que tiene un único elemento de tipo objeto, y ese objeto tiene a su vez tres campos cuyos valores son de tipo string, array y número.
b) Se trata de un array que tiene un único elemento de tipo objeto, y ese objeto tiene dos campos cuyos valores son de tipo string y un campo cuyo valor es de tipo número.
c) Se trata de un objeto que tiene tres campos cuyos valores son de tipo string, array y número.
d) Se trata de un objeto que tiene dos campos cuyos valores son de tipo string y un campo cuyo valor es de tipo número.

7. Suponga que está diseñando un Servicio Web según el enfoque REST y que desea modelar una funcionalidad que permite a un usuario inscribirse en una carrera de running. Como datos de entrada, es necesario proporcionar el identificador de usuario, el identificador de la carrera y la tarjeta bancaria con la que se realizará el pago de la inscripción. Ya existe un recurso colección con el URL `/carreras` que representa a las carreras a las que se puede inscribir un usuario. ¿Qué opción de diseño escogería de acuerdo al enfoque estudiado en la asignatura?

a) El servicio permitirá inscribirse en una carrera con el identificador 5 haciendo una petición POST al URL `/carreras/5` pasando los datos de entrada como parámetros de la petición.
b) El servicio permitirá inscribirse en una carrera con el identificador 5 haciendo una petición PUT al URL `/carreras/5/inscribir`, pasando el resto de datos de entrada como parámetros de la petición.
c) Las dos anteriores son válidas.
d) El servicio permitirá inscribirse en una carrera haciendo una petición POST al URL `/inscripciones` pasando los datos de entrada como parámetros de la petición.

8. En el contexto de Servicios Web REST, suponga un intermediario genérico de cache. Asumiendo que ninguna respuesta enviada por el servicio indica explícitamente que no se puede cachear, diga que afirmación es correcta teniendo en cuenta las convenciones seguidas en la asignatura:

a) Sería correcto que el intermediario cachease la respuesta a una petición GET `/books/1234` que ha devuelto un código de respuesta 500 Internal Error.
b) Sería correcto que el intermediario cachease la respuesta a una petición POST `/books/` que ha devuelto un código de respuesta 500 Internal Error.
c) Todas las anteriores.
d) Ninguna de las anteriores.

9. Dado el siguiente fragmento de un fichero de definición de la interfaz de un servicio Apache Thrift:

```
struct Instance {
  1: i64 instanceId
  2: string description
}
```

Indique qué afirmación NO es correcta:

a) Se corresponde con la declaración de un tipo definido por el usuario, que podrá utilizarse como tipo de los argumentos que recibe una función.
b) Se corresponde con la declaración de un tipo definido por el usuario, llamado `Instance`, que tiene un campo de tipo entero de 64 bits y otro de tipo cadena.
c) Se corresponde con la declaración de un tipo definido por el usuario, que podrá utilizarse como tipo de las excepciones que puede lanzar una función.
d) Si se utiliza el compilador del IDL de Apache Thrift a Java, para este tipo definido por el usuario se generará una clase cuyo nombre será "Instance".

10. Analice el siguiente fragmento de código correspondiente a una clase de pruebas de integración de la capa modelo, implementada con JUnit 5.

```
private static ProductService productService = null;

@BeforeAll
public static void init() {
    ...
    productService = ProductServiceFactory.getService();
}

private void removeProduct(Long productId) {
    productService.removeProduct(productId);
}

@Test
public void testCreateProductAndFindProduct() throws InputValidationException,
               InstanceNotFoundException {

    // Se crea un producto válido en memoria
    Product newProduct = getValidProduct();
    // Se crea el producto en BD
    Product createdProduct = productService.createProduct(newProduct);

    try {
        // Se recupera el producto de BD
        Product foundProduct =
            productService.findProduct(createdProduct.getProductId());
        // Se comprueba que el producto recuperado de BD tenga los valores
        // oportunos y sea igual al creado
        assertEquals(...);
        ...
        assertEquals(createdProduct, foundProduct);

    } finally {
        // Se elimina el producto de BD
        removeProduct(createdProduct.getProductId());

    }

}
```

Teniendo en cuenta que el método privado `removeProduct` de la clase de pruebas es un método que se utilizará desde los casos de prueba que necesiten borrar productos que se hayan creado desde ellos (como, por ejemplo, desde el caso de prueba `testCreateProductAndFindProduct`), y que el método `removeProduct` de la interfaz `ProductService` lanza la excepción `InstanceNotFoundException` cuando no existe un producto con el identificador que se le pasa

como parámetro, indique qué afirmación es correcta en base al enfoque aconsejado en la asignatura para la realización de pruebas de integración de la capa Modelo:

a) En la implementación del método privado `removeProduct` de la clase de pruebas debería declararse la excepción `InstanceNotFoundException` en la cláusula `throws`.

b) En la implementación del método privado `removeProduct` de la clase de pruebas debería capturarse la excepción `InstanceNotFoundException` y relanzarse como una `RuntimeException`.

c) En la implementación del método privado `removeProduct` de la clase de pruebas debería capturarse la excepción `InstanceNotFoundException` y continuar con la ejecución sin realizar ninguna acción en el bloque `catch`.

d) La implementación del método privado `removeProduct` de la clase de pruebas es correcta tal y como está en el enunciado, y los casos de prueba desde los que se llame deben declarar la excepción `InstanceNotFoundException` en su cláusula `throws` (como se hace, por ejemplo, en el caso de prueba `testCreateProductAndFindProduct` mostrado en el enunciado).