

Anexo 2

Pruebas para la obtención de títulos de Técnico y Técnico Superior

Convocatoria correspondiente al curso académico 2023-2024

(Resolución de 29 de diciembre de 2023 de la Dirección General de Educación Secundaria, Formación Profesional y Régimen Especial)

DATOS DEL ASPIRANTE			FIRMA
Apellidos:			
Nombre:	D.N.I. N.I.E. o Pasaporte:	Fecha:	

Código del ciclo:	Denominación completa del título:
IFCS02	Técnico Superior en Desarrollo de Aplicaciones Multiplataforma
Clave o código del módulo:	Denominación completa del módulo profesional:
486	Acceso a Datos

INSTRUCCIONES GENERALES PARA LA REALIZACIÓN DE LA PRUEBA
<ul style="list-style-type: none"> El examen tendrá una duración de 2 horas. La prueba consta de un examen tipo test con cuatro opciones de las cuales solamente una es correcta. Cada pregunta se responderá en el espacio dejado al efecto en la hoja de respuestas. Se rellenarán los recuadros para señalar la respuesta seleccionada. Si se quiere rectificar una respuesta contestada, se borrará toda la casilla de la respuesta incorrecta con Tipp-Ex o corrector, tal y como se puede apreciar en el siguiente ejemplo: <ul style="list-style-type: none"> Se elimina la selección de b para seleccionar la opción d: <input type="checkbox"/>a <input type="checkbox"/>b <input type="checkbox"/>c <input checked="" type="checkbox"/>d Se dispondrá de una hoja para borrador (o de varias si se requieren), que será proporcionada por el centro. Esa hoja se entregará obligatoriamente al final junto con el examen, si bien nada de lo escrito en la hoja de borrador se valorará en la corrección. Sólo se utilizará bolígrafo negro o azul y Tipp-Ex o corrector, no permitiéndose usar bolígrafo rojo, lapicero, etcétera. No se podrá emplear ningún dispositivo electrónico. Cualquier tachadura o borrón en una respuesta podrá invalidar toda la puntuación de la pregunta asociada.

CRITERIOS DE CALIFICACIÓN Y VALORACIÓN
<ul style="list-style-type: none"> El test se calificará sobre 10 puntos. Todas las preguntas se calificarán equitativamente con la misma cantidad de puntos. En cada pregunta se plantearán varias respuestas, y se deberá señalar la única que se considere correcta, según el caso. Cada respuesta correcta que se marque se valorará con 0,25 puntos, y si se marca alguna incorrecta, se valorará con una cantidad negativa equivalente a 0,075 puntos, es decir, se descontarán 0,075 puntos. Si no se está seguro de si una respuesta es correcta o no, y no se marca, no sumará ni restará puntos. Calificación final del módulo profesional: <ul style="list-style-type: none"> El alumno obtendrá en el módulo profesional una calificación entera entre 1 y 10. Dicha calificación se calculará: <ul style="list-style-type: none"> Si la calificación conseguida en la prueba es inferior a 5: se truncará dicha calificación. Si la calificación conseguida en la prueba es igual o superior a 5 y los decimales: <ul style="list-style-type: none"> Son inferiores a 0'5: se redondeará al entero inferior más próximo. Son iguales o superiores a 0'5: se redondeará al entero superior más cercano. La anterior regla tiene una excepción: las notas de examen inferiores a 1 se redondearán a 1.



CONTENIDO DE LA PRUEBA

1) ¿Cuál de las siguientes es una práctica adecuada para el manejo de excepciones al realizar operaciones de ficheros que podrían fallar?

- Capturar específicamente la excepción más general, Exception, para simplificar el código.
- Capturar y manejar excepciones específicas como IOException o SecurityException para proporcionar respuestas más detalladas a los errores.
- Ignorar las excepciones que no son críticas para la aplicación.
- Usar un solo bloque try-catch para todas las operaciones de fichero en un método.

Para el siguiente programa en Java:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;

public class DatabaseConnector {
    private static final String URL = "jdbc:postgresql://localhost:5432/mydatabase";
    private static final String USER = "admin";
    private static final String PASSWORD = "admin";

    public static void main(String[] args) {
        try (Connection connection = DriverManager.getConnection(URL, USER, PASSWORD)) {
            connection.setAutoCommit(false); // Desactivar auto-commit para gestionar transacciones

            try (Statement statement = connection.createStatement()) {
                // Ejecución de una sentencia de definición de datos
                statement.execute("CREATE TABLE IF NOT EXISTS users (id SERIAL PRIMARY KEY, username
VARCHAR(255) NOT NULL)");

                // Ejecución de una sentencia de manipulación de datos
                statement.executeUpdate("INSERT INTO users (username) VALUES ('johndoe')");

                // Ejecución de una consulta
                ResultSet resultSet = statement.executeQuery("SELECT * FROM users");
                while (resultSet.next()) {
                    System.out.println("User ID: " + resultSet.getInt("id") + ", Username: " +
resultSet.getString("username"));
                }

                connection.commit(); // Commit de la transacción
            } catch (SQLException e) {
```



```
        connection.rollback(); // Rollback en caso de error
        throw e;
    }
} catch (SQLException ex) {
    ex.printStackTrace();
}
}
}
```

2) ¿Qué efecto tiene `connection.setAutoCommit(false)`; en el contexto de una transacción?

- a. Habilita el auto-commit para mejorar la eficiencia de la ejecución.
- b. Deshabilita el auto-commit para permitir el control manual sobre las transacciones.
- c. No tiene efecto alguno sobre las transacciones dentro de este bloque de código.
- d. Configura la base de datos para rechazar automáticamente todas las transacciones.

3) ¿Qué protocolo de acceso a base de datos utiliza el programa?

- a. JDBC
- b. ODBC
- c. ADO.NET
- d. Entity Framework

4) Si la sentencia `statement.execute("CREATE TABLE IF NOT EXISTS users...")` falla, ¿qué método se invoca para manejar la situación?

- a. `connection.commit()`
- b. `connection.rollback()`
- c. `statement.close()`
- d. `DriverManager.getConnection()`

5) ¿Cuál de las siguientes afirmaciones es correcta respecto a la sentencia `ResultSet resultSet = statement.executeQuery("SELECT * FROM users");`?

- a. La sentencia no afecta la integridad de la base de datos y no requiere transacción.
- b. Esta consulta no puede ejecutarse sin un commit previo.
- c. Es una sentencia de manipulación de datos y modifica la base de datos.
- d. Solo puede ejecutarse si el auto-commit está habilitado.

6) ¿Qué sentencia describe mejor el desfase objeto-relacional demostrado en el programa?

- a. La necesidad de usar SQL en un lenguaje orientado a objetos como Java.
- b. La incompatibilidad de tipos entre SQL y Java que impide ejecutar consultas.
- c. La imposibilidad de ejecutar transacciones en lenguajes orientados a objetos.
- d. La falta de necesidad de gestionar transacciones en aplicaciones Java.

7) ¿Qué operación en el código asegura la integridad de la transacción en caso de éxito?

- a. `statement.execute("CREATE TABLE...")`
- b. `resultSet.next()`
- c. `connection.commit()`
- d. `DriverManager.getConnection()`

Para el siguiente programa en Java:

```
import org.hibernate.Session;
```



```
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.query.Query;

public class HibernateExample {
    public static void main(String[] args) {
        // Configuración y creación de una SessionFactory
        SessionFactory factory = new Configuration()
            .configure("hibernate.cfg.xml")
            .addAnnotatedClass(User.class)
            .buildSessionFactory();

        // Creación de una sesión
        Session session = factory.openSession();

        try {
            // Iniciando una transacción
            session.beginTransaction();

            // Guardar un objeto
            User newUser = new User("johndoe", "John Doe");
            session.save(newUser);

            // Consulta de objetos
            Query<User> query = session.createQuery("from User where username = 'johndoe'", User.class);
            User user = query.uniqueResult();
            System.out.println("User: " + user.getUsername());

            // Comprometer la transacción
            session.getTransaction().commit();
        } finally {
            session.close();
            factory.close();
        }
    }
}

@Entity
@Table(name = "users")
class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
```



```
@Column(name = "username")
private String username;

@Column(name = "full_name")
private String fullName;

// Constructor, getters y setters
public User() {}

public User(String username, String fullName) {
    this.username = username;
    this.fullName = fullName;
}

// getters and setters
}
```

8) ¿Qué archivo se utiliza para la configuración inicial de Hibernate en el ejemplo?

- a. hibernate.properties
- b. hibernate.cfg.xml
- c. application.properties
- d. hibernate.json

9) En el código dado, ¿qué anotación específica que la clase `User` es una entidad gestionada por Hibernate?

- a. `@Entity`
- b. `@Table`
- c. `@Column`
- d. `@Id`

10) ¿Qué método se utiliza para iniciar una transacción en Hibernate?

- a. `session.openTransaction();`
- b. `session.beginTransaction();`
- c. `session.startTransaction();`
- d. `session.getTransaction();`

11) ¿Qué realiza la consulta `session.createQuery("from User where username = 'johndoe'", User.class).uniqueResult();` en el código?

- a. Retorna todos los usuarios con username 'johndoe'.
- b. Retorna el primer usuario con username 'johndoe' encontrado.
- c. Retorna el único usuario con username 'johndoe' o lanza una excepción si hay más de uno.
- d. Crea un nuevo usuario con username 'johndoe'.

12) En el contexto de la persistencia del componente, ¿qué característica es necesaria para guardar y restaurar el estado de `DataAccessComponent`?

- a. Serialización



- b. Empaquetado
- c. Polimorfismo
- d. Reflejo

13) ¿Cuál de las siguientes afirmaciones es verdad sobre el estado de una entidad persistente?

- a. Una vez hayamos cerrado la sesión de Hibernate, la instancia persistente se convertirá en una instancia separada.
- b. Una nueva instancia de una clase persistente no está asociada a una Session.
- c. Puede hacer una instancia transitoria asociándola con una sesión.
- d. Ninguna de las anteriores.

14) Indica cuál de las siguientes no es una ventaja de la creación de componentes.

- a. Permiten el desarrollo software en menos tiempo.
- b. Permiten el desarrollo de software con mayor posibilidad de fallos.
- c. Permiten el desarrollo de software más barato.
- d. Permite el desarrollo de software de mayor calidad.

15) ¿Cuál de las siguientes no es una anotación de Hibernate?

- a. @Id
- b. @JoinTable
- c. @ManyToMany
- d. @Autowired

16) ¿Cuál de las siguientes afirmaciones es verdadera acerca de la anotación @GeneratedValue?

- a. Una anotación @GeneratedValue toma dos parámetros, uno de estrategia y otro generador.
- b. La anotación @GeneratedValue proporciona la generación de clave foránea que se utilizará.
- c. a) y b) son correctas.
- d. Ninguna de las anteriores es cierta.

17) ¿Cuál de las siguientes afirmaciones es verdadera acerca de la anotación @Entity?

- a. La anotación @Entity marca una clase como un vean de entidad
- b. La clase marcada como anotación @Entity debe tener un constructor sin argumentos.
- c. a) y b) son verdaderas.
- d. a) y b) son falsas.

18) ¿Cuál de las siguientes es un componente básico de Hibernate?

- a. Transaction.
- b. sessionFactory.
- c. Session.
- d. Todas son correctas.

19) ¿Cuál de las siguientes afirmaciones es verdadera acerca de JDBC?

- a. JDBC significa Java Database Connectivity
- b. JDBC proporciona un conjunto de API Java para acceder a las bases de datos relacionales del programa Java.
- c. Las APIs JDBC permiten a los programas Java ejecutar instrucciones SQL e interactuar con cualquier base de datos compatible con SQL.
- d. Todas son correctas.

20) ¿Cuál de los siguientes archivos es necesario para configurar el mapeo de clases con Hibernate?

- a. Pom.xml



- b. Hibernate.cfg.xml
- c. Reveng.xml
- d. Todos los anteriores.

21) ¿Qué base de datos embebida es conocida por ser ligera y almacenar la base de datos en forma de ficheros?

- a. MySQL
- b. SQLite
- c. H2
- d. PostgreSQL

22) El método execute(String) en JDBC devuelve true si:

- a. Ejecuta una consulta de lectura
- b. Ejecuta una sentencia DML
- c. Ejecuta una consulta que devuelve un ResultSet
- d. Ejecuta una consulta DDL

23) ¿Cuál es la finalidad de los controladores en JDBC?

- a. Realizar llamadas a procedimientos almacenados
- b. Permitir la conexión a una base de datos desde programas Java
- c. Gestionar las conexiones de red
- d. Definir reglas de acceso a bases de datos

24) ¿Cuál de los siguientes métodos se utiliza para mover el puntero en un ResultSet en JDBC?

- a. moveToNext()
- b. next()
- c. previous()
- d. first()

25) ¿Cuál de las siguientes declaraciones es verdadera con respecto a las transacciones en JDBC?

- a. Las transacciones no se utilizan en JDBC
- b. Una transacción agrupa un conjunto de sentencias SQL que se ejecutan de manera individual
- c. Una transacción garantiza que todas las sentencias se ejecutan o ninguna se ejecuta
- d. Las transacciones solo se utilizan para consultas, no para modificaciones de datos

26) El método executeUpdate en JDBC se utiliza para:

- a. Ejecutar una consulta de lectura
- b. Ejecutar una sentencia de modificación de datos
- c. Ejecutar una consulta DDL
- d. Ejecutar una consulta de búsqueda

27) La anotación @Entity en Hibernate se utiliza para:

- a. Definir una entidad en la base de datos
- b. Marcar una clase como entidad mapeada a una tabla en la base de datos
- c. Definir una entidad en la aplicación
- d. Marcar una clase como entidad sin relación con la base de datos

28) ¿Cuál de las siguientes clases se utiliza para realizar la lectura secuencial de caracteres en Java?

- a. FileReader
- b. FileInputStream
- c. BufferedReader



d. InputStreamReader

29) ¿Cuál es la diferencia clave entre un fichero de acceso aleatorio y un fichero de acceso secuencial?

- a. Los ficheros de acceso aleatorio permiten saltar a cualquier posición en el fichero, mientras que los de acceso secuencial no.
- b. Los ficheros de acceso secuencial son más eficientes que los de acceso aleatorio.
- c. Los ficheros de acceso aleatorio solo permiten la lectura de datos, mientras que los de acceso secuencial permiten lectura y escritura.
- d. No hay diferencia, ambos términos se utilizan indistintamente.

30) ¿Qué hace el método commit() en JDBC en el contexto de transacciones?

- a. Inicia una transacción
- b. Confirma una transacción
- c. Cancela una transacción
- d. Ejecuta una consulta DDL

31) ¿Por qué se recomienda el uso de PreparedStatement en lugar de Statement en JDBC?

- a. Porque PreparedStatement es más rápido
- b. Porque PreparedStatement aporta seguridad y evita ataques por inyección de código
- c. Porque PreparedStatement es más fácil de usar
- d. Porque PreparedStatement permite la ejecución de procedimientos almacenados

32) La interfaz JDBC que proporciona información sobre la base de datos, como las tablas y columnas, se llama:

- a. ResultSet
- b. DatabaseMetaData
- c. ResultSetMetaData
- d. StatementMetaData

33) ¿Cuál es la principal ventaja de usar ReadLine para leer archivos de texto?

- a. Lee archivos binarios de manera eficiente.
- b. Permite leer datos numéricos de un archivo.
- c. Lee una línea completa de texto a la vez, facilitando el procesamiento.
- d. No se utiliza para leer archivos en Java.

34) ¿Qué característica de los JavaBeans permite que estos componentes se comuniquen usando eventos?

- a. Introspección
- b. Manejo de eventos
- c. Propiedades
- d. Persistencia

35) En el patrón Data Access Object (DAO), ¿qué tipo de operaciones suelen devolver un valor booleano?

- a. Consultas de búsqueda que devuelven listas de objetos.
- b. Operaciones de inserción, modificación y borrado.
- c. Métodos que imprimen datos en pantalla.
- d. Métodos que establecen conexiones a la base de datos.



- 36) ¿Cuál de las siguientes no es una desventaja sobre el uso de componentes en el desarrollo de software?**
- a. Aumento de la complejidad inicial debido a la modularización excesiva.
 - b. Dificultad en la gestión de dependencias entre componentes.
 - c. Reducción en la capacidad de reutilización de los componentes.
 - d. Posible impacto negativo en el rendimiento por la sobrecarga de comunicación entre componentes.
- 37) ¿Qué permite la característica de persistencia en los JavaBeans?**
- a. La comunicación entre diferentes beans mediante eventos.
 - b. La modificación de las características de apariencia y comportamiento durante el diseño.
 - c. El almacenamiento del estado del bean para restaurarlo posteriormente.
 - d. La integración del bean en cualquier plataforma de software sin modificación.
- 38) ¿Cuál de estas anotaciones se usa en Springdoc-OpenAPI para documentar el modelo de datos utilizado en una API?**
- a. @Schema
 - b. @RequestBody
 - c. @Parameter
 - d. @Hidden
- 39) ¿Qué permite construir las anotaciones de Springdoc-OpenAPI como @Tag y @Operation en un proyecto de Spring Boot?**
- a. Una base de datos integrada
 - b. Un servidor de aplicaciones
 - c. Una interfaz de usuario de Swagger
 - d. Un gestor de dependencias

Sobre el siguiente código:

```
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.MongoCollection;
import org.bson.Document;

public class MongoExample {
    public static void main(String[] args) {
        MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");
        MongoDatabase database = mongoClient.getDatabase("testdb");
        MongoCollection<Document> collection = database.getCollection("testcollection");

        Document filter = new Document("age", new Document("$gt", 25));
        collection.find(filter).forEach(doc -> System.out.println(doc.toJson()));
    }
}
```

- 40) ¿Qué método del objeto collection se utiliza en el siguiente programa Java para filtrar documentos en MongoDB según un criterio específico?**



Unión Europea

Fondo Social Europeo

"El FSE invierte en tu futuro"



Comunidad de Madrid

- a. find()
- b. insertOne()
- c. updateOne()
- d. deleteOne()