



## EXAMEN TEORICO MODULO 2 IFCD 0210 , 5 septiembre 2025

**CURSO: Desarrollo de aplicaciones con tecnologías web (IFCD0210)**

**MODULO MF0492\_3: PROGRAMACION WEB EN EL ENTORNO SERVIDOR**

Nombre y apellidos: Alejandro Dal Bosco Martínez

---

### \*\*\* UNIDAD 1844 (back-end, java) \*\*\*

1. En java, para acceder al atributo String nombre de la clase Peliculas, en modo public o private:

- con el private, accedemos mediante `string Peliculas.nombre( );`
- con el public, sería `peliculas(String nombre( ));`
- con el private, sería `Peliculas.getNombre( );`
- con el public, sería `Peliculas.getNombre;`

2. Para indicar que una variable es constante en Java, usamos:

- `const`
- `let`
- `var`
- `final`

3- El número 8,3625f

- es un float
- es un int
- es un double
- es un byte

4- Para leer un número entero del usuario, donde 'teclado' es la instancia del Scanner, la última instrucción donde el usuario entra el número, sería:

```
int entrada = input(nombre.usuario(int))
int entrada = scanner.variable_de_usuario
int entrada = teclado (new Scanner(int))
int entrada = teclado.nextInt( )
```

5. Para crear una nueva instancia de la clase Tren, con los atributos nombre, numVagones y KmHora, usamos el siguiente esquema:

- Tren tren1 = new Tren("Talgo", 10, 80);
- tren1 = new tren(Talgo; 10; 80);
- new tren = tren1("Talgo", 10, 80);
- Tren <tren> tren1 = new Tren < > ( );

6. Para comparar un valor de entrada de teclado con "Carlitos", usamos:

- if("entrada" == "Carlitos") {
- if(entrada.equals("Carlitos") {
- if(entrada = "Carlitos") {
- if(entrada.equals = Carlitos) {

7. Sobre los getters y setters:

- el getter lee, y el setter retorna un resultado
- son métodos obligatorios, y que no pueden modificarse
- el setter lee y el getter recibe parámetros
- el setter recibe parámetros y el getter devuelve datos

8. Las clases acostumbran a contener:

- funciones y métodos
- otras clases y ArrayLists
- atributos y propiedades
- atributos y métodos

9. La diferencia principal entre array y ArrayList es:

- el primero contiene una cantidad inmutable de elementos y el segundo no
- el primero suele usar tipos primitivos y el segundo, objetos
- el primero se ejecuta más rápido que el segundo
- todas las anteriores son ciertas

10. En la condición siguiente:

```
while(!salir) {  
    // mostrar menú y scanner para input de usuario etc  
    if(opcion == 0){salir = true}
```

- la condición está mal expresada
  - saldrá del bucle en cuanto la opción sea 0
  - saldrá del bucle en cuanto la opción sea !0
  - genera un bucle infinito si salir es true

1. En mySQL, la clave foránea en esta tabla:

- Relaciona esta tabla con la llave foránea de la otra tabla
- Relaciona una tabla con sus propios datos
- Relaciona un dato con la cabecera de esta tabla
- Relaciona esta tabla con la llave primaria de la otra tabla

2. En esta consulta mySQL: [instrucción que falta] FROM usuarios WHERE id = 2;

¿Qué instrucción se usa para borrar toda la fila de una tabla, incluyendo su contenido?

- Drop
- Remove
- Delete
- Erase

3. Para hacer una consulta conjunta de dos tablas:

- Se usa Next
- Se usa Join
- Se usa sum()
- Se usa avg()

4. ¿Como buscarías a 'Josefa' si solo sabes que su nombre acaba en 'sefa'?

- WHERE nombre LIKE 'sefa'
- WHERE nombre IS LIKE 'Josef'
- WHERE nombre LIKE '%sefa'
- WHERE nombre LIKE 'sefa%'

5. Si tienes muchos valores iguales pero solo quieres ver una lista de los que son diferentes:

- Usa SELECT DIFFERENT
- Usa SELECT \* NOT SAME
- Usa SELECT UNIQUE
- Usa SELECT DISTINCT

6. ¿Con qué instrucción ordenarías una lista de años en orden inverso?

- SELECT anio FROM tabla ORDER BY anio DESC

- SELECT anio FROM tabla BY ORDER DESC
- SELECT anio FROM lower TO higher
- SELECT anio FROM tabla ORDER anio BY YEAR DESC

7. ¿Cómo mostrarías los registros donde el año (anio) sea mayor que 1992?

- con: SELECT anio FROM tabla WHERE >2022
- con: SELECT \* anio FROM 1992
- con: SELECT \* FROM tabla WHERE anio>1992
- con: SELECT \* FROM tabla WHERE anio <2012 AND anio>1982

8. ¿Cómo seleccionarías los años entre 1990 y 2020, ambos incluidos?

- con BETWEEN 1990 AND 2020
- con anio>1990 AND anio<2020
- con anio AND 1990 OR anio>1990
- con anio FROM 1990 TO 2020

9. JDBC nos permite...

- conectar el front-end con el back-end
- conectar la base de datos phpMyAdmin con WorkBench
- conectar javascript con la base de datos
- conectar java con la base de datos

10. Y JPA, además, incluye:

- un framework llamado spring
- @Anotaciones
- fetch
- todas las anteriores

## **\*\*UNIDAD 1846 (APIS Y CONEXIONES FRONT-END/BACK-END) \*\*\***

1.Las APIs

- Nos permiten solo leer datos de una base de datos externa
- Sirven para Adjuntar Programas e Interfaces
- Nos permiten realizar diversas operaciones CRUD con BD externas o locales
- Solo se usan en teléfonos móviles (Java-Android)

2. ¿Cuál es la expresión correcta?

- `fetch('/ditto[2].name')`
- `fetch("api:ejemplo/com")`
- `fetch("ejemplo.com/2.jpg")`
- `fetch('http://ejemplo.com')`

3. ¿Qué significa Request-Response?

- Una API que nos permite hacer preguntas y respuestas
- Un proceso que se da en Java para conectar funciones
- Se realiza una conexión con el front-end y se guarda en SQL
- Se realiza una petición desde la API y se recibe una respuesta

4. ¿En la cadena de promesas fetch-then-catch para qué sirve el catch?

- Para controlar qué ocurre si no funciona la conexión fetch o los procesos then
- Para resetear el puerto de conexión a localhost:8080
- Para salir del ámbito de la variable 'data' (resultados del fetch)
- Para evitar que se exponga información sensible a usuarios no autorizados

5. Las Apis online..

- Son solo compatibles con algunos sistemas operativos
- Pueden servir para particulares o empresas

- Puedes usarlas sin conexión a internet
- No puedes conectarte a dos APIs diferentes desde el mismo documento JS

6. Una diferencia relevante entre Postman y Thunder Client es...

- Thunder Client está diseñado específicamente como extensión ligera para VS Code, mientras que Postman ofrece una plataforma más completa y autónoma para pruebas de APIs.

- Solo Thunder Client permite hacer peticiones REST.
- Postman es completamente gratuito, mientras que Thunder Client es solo de pago.
- Ambos son de código abierto y mantenidos por la comunidad.

7. Para mostrar los datos que te sirve la API en pantalla...

- necesitas obligatoriamente un framework en front-end (React, Vue, Angular)
- necesitas solo fetch JS

- necesitas HTML, JS y tal vez CSS

- necesitas siempre una suscripción a la API con API KEY

8. En este código:

```
let libros = [  
  { id: 1,  
    lenguaje: "java",  
    edición: "tercera"  
    autor: "Herber Schildt" },  
  { id: 2,  
    lenguaje: "C++",  
    edición: "segunda"  
    autor: "E.Balagurusamy" }  
]
```

el valor de la variable "libros" :

- Está definida incorrectamente
- Son dos objetos en un array
- Son cuatro objetos clave-valor

- Son 8 elementos de array

9. En esta variable JS anterior con estructura clave-valor, ¿Cómo accederías al autor del segundo libro?

- console.log(data.libro[2].edicion)
- console.log(libros[1].autor)
- console.log(data.libros[2].autor)
- console.log(libros[1][3])

10. En las APIS, las operaciones CRUD desde JS se ejecutan mediante las instrucciones:

- ADD, PUT, PUSH, DELETE
- POST, PATCH, DELETE, PULL
- REMOVE, POST, GET, INSERT
- GET, PUT, DELETE, POST