

## Programación

### EXAMEN TEMA 7 – ESTRUCTURAS DINÁMICAS DE DATOS

(03/04/2025)

1. (0,5p) ¿A partir de qué interfaces principales se implementan todo tipo de listas, pilas, colas, conjuntos y mapas/diccionarios?
2. (1,5p) ¿En qué consisten las estructuras *FIFO*? Indica el nombre de la clase que las implementa en *Java* y menciona sus métodos principales. Pon ejemplos de uso en aplicaciones reales y ayúdate de dibujos si lo necesitas.
3. (1p) El siguiente código *Java* contiene lógica de implementación de una pila:

```
public static int evaluar(String expresion) {
    Stack<Integer> pila = new Stack<>();
    for (String c : expresion.split(" ")) {
        if (c.matches("\\d+")) {
            pila.push(Integer.parseInt(c));
        } else {
            int b = pila.pop();
            int a = pila.pop();
            switch (c) {
                case "-": pila.push(a - b); break;
                case "*": pila.push(a * b); break;
            }
        }
    }
    return pila.pop();
}

public static void main(String[] args) {
    String expresion = "4 2 - 5 *";
    System.out.println(evaluar(expresion));
}
```

Dibuja cómo va cambiando la pila mientras se ejecuta el programa e indica qué valor numérico retornará el método ***evaluar()***.

4. (1p) ¿Qué diferencia hay entre una estructura *Queue* y *Deque*? Indica cómo cambian los métodos disponibles de una a otra.
5. (1p) ¿Por qué es importante implementar los métodos ***.hashCode()*** y ***.equals()*** en clases personalizadas? Relaciónalo con las estructuras de tipo *Set* y pon ejemplos.

6. (1p) Indica la salida que imprime el siguiente programa:

```
public static void main(String[] args) {
    String[] palabras = {"banana", "manzana", "pera", "banana"};
    Set<String> hashSet = new HashSet<>();
    Set<String> treeSet = new TreeSet<>();
    Set<String> linkedHashSet = new LinkedHashSet<>();
    for (String palabra : palabras) {
        hashSet.add(palabra);
        treeSet.add(palabra);
        linkedHashSet.add(palabra);
    }
    System.out.println("HashSet:" + hashSet);
    System.out.println("TreeSet: " + treeSet);
    System.out.println("LinkedHashSet: " + linkedHashSet);
}
```

7. (1p) Identifica los elementos característicos de los mapas y relacionalos con el funcionamiento de las tablas Hash. Explica qué ocurre cuando hay una colisión y cómo lo solucionan los lenguajes de programación.

8. (2p) ¿Qué estructura de datos será más eficiente usar en estos casos? Explica por qué.

- a) Contar la cantidad de veces que aparece cada palabra en un texto grande.
- b) Buscar rápidamente si un usuario está registrado.
- c) Almacenar un historial de navegación web manteniendo el orden de acceso.
- d) Implementar la función “deshacer” (*Ctrl + Z*) en un editor de texto

9. (0,5p) ¿Qué resultado devolverá el siguiente código *Java* para la lista dada?

```
public static void main(String[] args) {
    ArrayList<String> nombres = new ArrayList<>(Arrays.asList("Patri","Pedro"));
    Iterator<String> it = nombres.iterator();
    while (it.hasNext()) {
        it.remove();
    }
}
```

10. (0,5p) ¿Cuándo es más eficiente usar *StringBuilder* respecto a *String*?