

Algoritmos y Programación 1

Apellido y Nombres:

DNI:

E1	E2	E3	E4	E5	Calificación

Ejercicio 1 { java, arrays }

Implementar según la siguiente especificación, agregando pre y post-condiciones. Incorporar alguna validación de dominio. Implementar la clase **FestivalDeCirco** tal que

1. Se construya a partir de la cantidad máxima de asistentes que pueden asistir, y la cantidad de espectáculos que se brindarán.
2. Agregue un asistente, indicando su nombre y la cantidad de espectáculos a los que concurrirá.
3. Devuelva la cantidad de espectáculos a los que concurrirá un asistente, buscándolo por su nombre.
4. Cuente la cantidad de asistentes.
5. Calcule el dinero total que ingresará, asumiendo que se cobra \$2 por espectáculo al que se asistirá.

Ejercicio 2 { pruebas }

Plantear cinco (5) pruebas para el problema del punto anterior. Los casos planteados deberán ser **conceptualmente** diferentes. Escribir el código en JUnit para tres (3) de esos casos.

Algoritmos y Programación 1

Apellido y Nombres:

DNI:

Ejercicio 3 { memoria }

Realizar los diagramas de Stack y Heap para los momentos #uno y #dos del siguiente fragmento de código.

```
Mesa mesa = new Mesa(7);
Postre flan = new Postre("Helado", 2);
Postre pastafrola = null;
Postre helado = new Postre("Flan", 4);
mesa.agregarPostre(flan, 8);
flan = helado;
// #uno
mesa.agregarPostre(new Postre("Torta de manzana", 1), 1);
Postre strudel = mesa.obtenerPostre(8);
mesa.agregarPostre(pastafrola, 8);
Assert.assertFalse(mesa.obtenerPostre(1) == pastafrola);
// #dos
```

Ejercicio 4 { algoritmos, matrices }

Dada una matriz de **Color** y de tamaño **n x n**, siendo **n** impar y mayor o igual a 3, y un **Color** determinado, escribir un algoritmo para pintar un lienzo que originalmente es de **Color.BLANCO** con el patrón solicitado. Se debe **pintar sobre el lienzo original**. No se recomienda pintar de **Color.BLANCO**.

```
public void pintarLienzo(Color[][] lienzo, Color pincel) {
    // tu código aquí
}
```

Ejercicio 5 { lógica }

Explique y ejemplifique con código Java el concepto de disyunción binaria (OR). Escriba la tabla de verdad para dicho concepto. El ejemplo deberá ser aplicado y no conceptual (las variables deben tener sentido).

Para el pizarron:

Anexo pto 1

```
class FestivalDeCirco {  
    public FestivalDeCirco(int, int) { /* ... */ }  
    public void agregarAsistente(String, int) { /* ... */ }  
    public int obtenerEspectaculos(String) { /* ... */ }  
    public int cantidadDeAsistentes() { /* ... */ }  
    public int dineroRecaudado() { /* ... */ }  
}
```

Anexo pto4

Se proporcionan ejemplos para $n = 3$, $n = 5$ y $n = 7$

