

**Reglas del juego:**

- Tal como se le dijo antes, durante este examen usted no puede pedir ABSOLUTAMENTE nada prestado a sus compañeros, ni hablar con ellos.
- Su nombre y su firma a la aceptación del compromiso de no hacer fraude, en la hoja de respuestas a este examen, deben ir en lapicero. Si emplea más de una hoja márkelas TODAS de igual forma. Además debe firmar todas las hojas del cuestionario, en la parte inferior.
- Por ningún motivo puede salir del salón, antes de terminar el examen. De manera que si se retira se considerará que terminó su trabajo.
- Puede utilizar solo el libro y las guías teóricas que entregadas por el profesor; pero se le recomienda que lo haga cuando esté seguro de qué quiere buscar, en caso contrario estas consultas se convierten en una distracción que le hace perder tiempo.
- No se resolverá **ninguna** duda durante el examen, así que si algo no le queda claro y usted requiere hacer algún tipo de supuesto, ESCRÍBALO y de una breve explicación de la razón por la cual lo supone.

La forma de evaluación de este parcial, es:

| Ítem                                        | Valor      |
|---------------------------------------------|------------|
| <b>Primer problema:</b>                     | <b>1.5</b> |
| Entradas                                    | 0.25       |
| Salidas                                     | 0.25       |
| Identificar parámetros y retorno del método | 0.25       |
| Proceso                                     | 0.75       |
| <b>Segundo problema:</b>                    | <b>3.5</b> |
| Método 1:                                   | 1.0        |
| Sintaxis                                    | 0.5        |
| Correspondencia con el análisis             | 0.5        |
| Método 2:                                   | 1.0        |
| Identificar los problemas                   | 0.5        |
| Explicar cómo solucionaría los problemas    | 0.5        |
| Método 3:                                   | 1.5        |
| Entradas                                    | 0.25       |
| Salidas                                     | 0.25       |
| Identificar parámetros y retorno del método | 0.25       |
| Proceso                                     | 0.75       |

**Punto 1**

Para el siguiente punto, **se espera que usted haga un análisis completo** para llegar a la solución. De ello depende que se le contrate para presentar la solución completa.

Una empresa está desarrollando una aplicación para el área de recursos humanos y lo ha contratado a usted para **realizar el análisis de una clase ejecutable** que les

permita visualizar por pantalla el calendario de cualquier mes indicado por el usuario, para el año 2008. El sistema debe pedir el mes y luego imprimir en pantalla los días del mismo. Ejemplo: Si alguien digita el mes 2 que corresponde a febrero el sistema debería imprimir en pantalla la siguiente información:

| Lu | Ma | Mi | Ju | Vi | Sa | Do |
|----|----|----|----|----|----|----|
|    |    |    |    | 1  | 2  | 3  |
| 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 |    |    |

La solución debe tener un método que recibe por parámetros dos arreglos. Uno de tipo int el cual almacena en cada posición el número del día de la semana en que comienza cada mes del año 2008, el otro también de tipo int almacena en cada posición la cantidad de días que tiene cada mes. La siguiente figura ilustra el estado de los arreglos:

Número de día de la semana en que inicia cada mes: (0. lunes, 1. martes...)

|       |         |       |       |      |       |       |        |      |     |     |     |
|-------|---------|-------|-------|------|-------|-------|--------|------|-----|-----|-----|
| 1     | 4       | 5     | 1     | 3    | 6     | 1     | 4      | 0    | 2   | 5   | 0   |
| enero | febrero | marzo | abril | mayo | junio | julio | agosto | sept | oct | nov | dic |

Note que febrero empieza en el día 4, esto hace que usted deba empezar a pintar el calendario en la cuarta columna de los días tal como se muestra en el ejemplo.

Número de días de cada mes:

|       |         |       |       |      |       |       |        |      |     |     |     |
|-------|---------|-------|-------|------|-------|-------|--------|------|-----|-----|-----|
| 31    | 29      | 31    | 30    | 31   | 30    | 31    | 31     | 30   | 31  | 30  | 31  |
| enero | febrero | marzo | abril | mayo | junio | julio | agosto | sept | oct | nov | dic |

## Punto 2

Una cadena de supermercados con presencia en cuatro ciudades de Colombia, pagó por el desarrollo de una aplicación que le permitiera almacenar la información de las unidades de productos vendidas durante un mes en Cali, Bogotá, Medellín y Barranquilla y consultar ciertos reportes relacionados con las ventas.

La aplicación debía incluir las siguientes funcionalidades, desarrolladas cada una en un método diferente:

- **Método 1:** Crear una lista con los precios unitarios de cada producto. No se sabe de antemano la cantidad de productos, el usuario debe ir indicando si desea agregar o no un nuevo precio unitario. La cantidad de productos que maneja el supermercado corresponderá a la cantidad de precios unitarios ingresados por el usuario.
- **Método 2:** Almacenar la información de las unidades vendidas en cada ciudad para cada producto.
- **Método 3:** Mostrar por pantalla el valor total en pesos que se vendió en cada ciudad, teniendo en cuenta todos los productos.

A continuación se presenta un ejemplo de la información de precios unitarios y unidades vendidas que podría existir en el sistema en un momento determinado:

|            | Cali  | Bogotá | Medellín | Barranquilla |       |
|------------|-------|--------|----------|--------------|-------|
| Producto 1 | 40000 | 2700   | 4320     | 8300         | 10000 |
| Producto 2 | 10000 | 5000   | 500      | 3200         | 12000 |
| Producto 3 | 9300  | 8000   | 2000     | 6000         | 35000 |
| .          | .     | .      | .        | .            | .     |
| .          | .     | .      | .        | .            | .     |
| .          | .     | .      | .        | .            | .     |
| Producto n | 3600  | 5700   | 800      | 6450         | 8000  |

Precios Unitarios  
en pesos

Los productos son identificados con un número consecutivo que empieza en 1.

El problema que tiene ahora la cadena de supermercados, es que la persona a la que contrataron, dejó el trabajo a medias. Lo que ellos necesitan es que usted les ayude a completar o corregir lo que está pendiente.

1. El código del primer método no fue desarrollado. Lo que usted debe hacer es **DESARROLLAR EL CÓDIGO CON BASE EN EL ANÁLISIS QUE ENTREGÓ LA PERSONA.**

ANÁLISIS:

```

/*
 * Método: preciosProductos
 * Parámetros: Ninguno
 * Retorno: Un ArrayList con el precio de cada uno de los productos que maneja la
 * cadena de supermercados en todo el país.
 * Entradas: - El precio de cada producto (Repetitiva)
 *           - 1 Si desea continuar, 0 en caso contrario
 * Salidas: Ninguna
 * Proceso:
 * - Crear una lista de tipo ArrayList para almacenar el precio de venta de cada
 * producto.
 * - Definir las variables continuar para controlar si el usuario desea ingresar
 * un nuevo producto y contador iniciada en 1 para manejar el identificador de
 * cada producto.
 * - Declarar una variable de tipo Double para almacenar cada valor que se desea
 * guardar en el ArrayList.
 * - Hacer
 *   - Pedir el precio unitario del producto con código contador y capturarlo
 *   en una variable de tipo double.
 *   - Envolver en una variable tipo Double (wrapper) el número digitado por
 *   el usuario y almacenarla en el ArrayList.
 *   - Preguntar al usuario si desea continuar o no.
 *   - Incrementar el contador que identifica el producto.
 * Mientras el usuario dese continuar
 * - Retornar el ArrayList
 */

```

2. En cuanto al segundo método, la persona entregó el análisis y el código, sin embargo, el código no está funcionando correctamente. Lo que usted debe hacer es **IDENTIFICAR LOS PROBLEMAS QUE TIENE EL CÓDIGO E INDICAR COMO LOS SOLUCIONARÍA**.

ANÁLISIS:

```
/*
 * Método: almacenarVentas
 * Parámetros: La cantidad de productos y un arreglo con los nombres de las
 * ciudades.
 * Retorno: Una matriz de tipo int con las unidades vendidas de cada producto en
 * cada ciudad.
 * Entradas: La cantidad de unidades vendidas de cada producto en cada ciudad
 * (repetitiva)
 * Salidas: Ninguna
 * Proceso:
 * - Crear una matriz de longitud número de productos por 4, de tipo int.
 * - Para cada producto y para cada ciudad preguntar la cantidad de unidades
 * vendidas y almacenarlas en la matriz en la posición correspondiente.
 * - Retornar la matriz.
 */
```

CÓDIGO:

```
public static int[][] almacenarVentas(int cantProductos, String[] ciudades)
{
    Scanner lector = new Scanner(System.in);
    int[][] ventas = new int[4][cantProductos];
    for(int filas=0;filas<cantProductos;filas++)
    {
        System.out.println("Unidades vendidas para el producto #"+(filas+1));
        for(int col=0;col<4;col)
        {
            System.out.print("En "+ciudades[col]+":");
            ventas[col][filas]=lector.nextInt();
        }
    }
    return ventas;
}
```

3. En cuanto al tercer método, la persona contratada no hizo absolutamente nada. Lo que usted debe hacer es **DESARROLLAR ÚNICAMENTE EL ANÁLISIS** para dar solución a lo planteado en el método 3.