

**Nombre** \_\_\_\_\_ **Matrícula** \_\_\_\_\_ **Firma** \_\_\_\_\_

1. (10%) El error en el llamado al método `calculaDivisores` y su definición se resuelve:

```
public static int calculaDivisores (int n){  
    String numeroDivi, elemento;  
  
    return numeroDivi;  
}  
  
public static void main(String args[]) {  
    ...  
  
    String respuesta = calculaDivisores(5);  
}
```

- a) Al corregir el nombre del método en la invocación a `calculaDivisores`
- b) Al corregir el tipo de dato del regreso declarado en la firma de la función `int calculaDivisores`
- c) Al corregir el tipo de dato de la variable de regreso de la función `String numeroDivi`
- d) Al corregir el tipo de dato de la variable que recibe la respuesta en el llamado a la función `String respuesta`

2. (10%) La instrucción

```
do { .....  
} while (condición);
```

- a) Se ejecuta solamente una vez
- b) Se ejecuta cero veces
- c) Se ejecuta al menos una vez
- d) Se ejecuta cero veces o más

3. (10%) La condición

```
do {  
} while (seguir=="si");
```

- a) Evalúa que la variable "seguir" sea diferente del valor "si"
- b) Evalúa que la variable "seguir" sea igual al valor "si"
- c) Evalúa que la variable "seguir" sea diferente del valor "si" y continúa en el ciclo
- d) Evalúa que la variable "seguir" sea igual al valor "si" y continúa en el ciclo

4. (10%) La condición

```
while (numero <= n)
```

- a) Vuelve a entrar al ciclo si "numero" es menor que n
- b) Vuelve a entrar al ciclo si "numero" es menor o igual que n
- c) Vuelve a entrar al ciclo si "numero" es mayor que n
- d) Vuelve a entrar al ciclo si "numero" es mayor o igual que n

5. (10%) La instrucción en java donde resultado, n y numero son enteros genera

```
resultado = n % numero;
```

- a) El cociente de n entre numero
- b) El divisor de n entre numero
- c) El residuo de n entre numero
- d) El dividendo de n entre numero

6. (10%) La estructura de programa

```
if ( a > 1)
    if (b > 2)
```

- a) Es un if complejo
- b) Es un if simple
- c) Es un if anidado
- d) Es un if compuesto

7. (10%) En la definición de un ciclo la variable de control se debe

- a) Iniciar, modificar y reiniciar
- b) Iniciar, evaluar y modificar
- c) Declarar, evaluar y modificar
- d) Declarar, iniciar y modificar

8. (10%) Las instrucciones siguientes permiten:

```
double variable;
s = new Scanner(System.in);
variable = s.nextDouble();
```

- a) Imprimir en pantalla el contenido de variable como valor con decimales
- b) Leer del usuario el valor y guardarlo en variable como valor con decimales
- c) Leer del usuario el valor y guardarlo en variable como valor entero
- d) Imprimir en pantalla el contenido de variable como valor entero

9. (10%) La instrucción “else” en una estructura de condición if

- a) Es obligatoria
- b) Solamente se puede utilizar en if complejos
- c) Solamente se puede utilizar en if simples
- d) Es opcional

10. (10%) En el proceso de aprendizaje de la programación, el código en java

- a) Se puede escribir después de hacer el razonamiento y el pseudo-código
- b) Se puede escribir después de hacer el razonamiento
- c) Se puede escribir sin hacer el razonamiento, ni el pseudo-código
- d) Se puede escribir y después desarrollar el razonamiento y el pseudo-código