


<b>FINAL ORDINARIA : TRIMESTRE 1 (1H 30') TRIMESTRE 2 (1H30')</b>	<b>26/06/2020 - 16:00</b>
Nombre y Apellidos:	DNI/NIE: Firma:
1º Desarrollo de Aplicaciones Web (Vespertino) Módulo: Entornos de Desarrollo	 IES Alonso de Avellaneda (Alcalá)

Se realizará la Parte del Trimestre 1 en primer lugar. Al finalizar se hará un descanso de 10 min para luego continuar con la Parte del Trimestre 2.

Es necesario obtener una calificación igual o mayor que 5 en ambos trimestres para poder superar el módulo. En caso contrario, se iría a la prueba final extraordinaria

Los alumnos que tengan pendiente sólo un trimestre deberán obtener una nota igual o mayor que 5 para poder superar el módulo, en caso contrario, deberán presentarse a la prueba final extraordinaria.

Si el/los ejercicio/s resultan plagio de recursos de Internet o de otras pruebas será invalidado

1.

## Trimestre 2

**Ejercicios (10 puntos)** (Recogida de los fuentes del programa en repositorio github de cada alumno)

Todos los ficheros necesarios están en el repositorio de la prueba.

1. Según el código facilitado (5p):

```
public static int Rangos(int x, int y){
    Scanner sc = new Scanner(System.in);
    int num;
    int c = 0;

    if ( x > 0 && y > 0 ){
        num = entrada.nextInt();
        while (num != 0){
            if ( num >= x && num <= y ) {
                System.out.println("\tNúmero en el rango");
                c++;
            } else
                System.out.println("\tNúmero fuera de rango");
            System.out.println("Escribe un número");
            num = entrada.nextInt();
        }//fin While
    }
    else
        c = -1;
    return c;
}
```

- a) Realiza el grafo de complejidad ciclomática completo escribiendo los nodos, flechas y condiciones (3p).
  - b) Calcula la complejidad ciclomática de las tres formas indicando las fórmulas que se necesitan. (0.5p)
  - c) Define el conjunto básico de caminos indicando los nodos de cada uno (0.5p)
  - d) Elabora los casos de prueba para cada camino (1p)
2. Se va a realizar una entrada de datos por teclado de un empleado, se definen 3 campos de entrada y una lista para elegir el puesto. Dicha aplicación acepta los datos de esta forma: (2p)
- Empleado: número de tres dígitos que no empieza por 0.
  - Departamento: en blanco o cualquier número de dos dígitos.
  - Puesto: Analista, Diseñador, Programador u Elige Puesto
- Escribe la tabla con las condiciones de entrada, clases de equivalencia, clases válidas, identificador clase válida, clases no válidas, identificador clase no válida.

3. Crea una batería de pruebas parametrizadas y aserciones con valores límite que verifiquen la salida de dicho método mediante JUnit4. (3p)

**Tip: el objeto Fecha crea una fecha actual y el SimpleDateFormat crea un formato que quieras imprimir según la forma: "yyyy/MM" mostrará "2020/05", etc. Con el método format pasando la fecha de hoy del objeto Date, tendrás un String con ese formato.**

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class Fecha {

    SimpleDateFormat formato;
    Date hoy;

    public Fecha() {
        hoy = new Date();
    }

    public String devuelveFecha(int tipo) {
        String cad = "";
        switch (tipo) {

            case 1: {
                formato = new SimpleDateFormat("yyyy/MM");
                cad = formato.format(hoy);
                break;
            }

            case 2: {
                formato = new SimpleDateFormat("MM/yyyy");
                cad = formato.format(hoy);
                break;
            }

        }
    }
}
```

```
        case 3: {
            formato = new SimpleDateFormat("MM/yy");
            cad = formato.format(hoy);
            break;
        }
        default:{
            cad = "ERROR";
        }
    }

    return cad;
}
}
```