

## ENTORNOS DE DESARROLLO - EXAMEN FINAL 2º EVALUACIÓN

C.F.G.S 1º D.A.M.

11-Marzo-2021

I.E.S. Fernando Aguilar Quignón

Al entregar tu examen, debes ser capaz de explicar cada ejercicio. No está permitida la copia de compañeros u otro medio. No puedes comunicarte con tus compañeros de ninguna manera mientras el examen está en progreso. En caso de detectar copias, el examen se puntuará con un 0.

### Ejercicio 1 - RA3 Pruebas - 1,5 puntos.

#### Sobre las pruebas de software...

- a ¿Qué son las pruebas de software? (0,25)
- b ¿Qué es un caso de prueba? (0,25)
- c ¿Qué diferencia existen entre las técnicas de desarrollo de casos de prueba de caja blanca y caja negra? (0,5)
- d Define el concepto de “Complejidad ciclomática” (No indicar la fórmula). (0,25)
- e Si nuestro programa tiene una CC de 15 ¿Cómo de complicado/arriesgado es nuestro programa? (0,25)

### Ejercicio 2 - RA4 Optimización - 2,5 puntos.

#### Sobre el control de versiones...

- f ¿Qué es un sistema de control de versiones? (0,25)
- g Indica cuatro motivos de porqué usar un sistema de control de versiones nos hará mucho más felices a la hora de desarrollar software (0,5)
- h ¿En qué se basa un sistema de control de versiones distribuido? (0,25)
- i ¿Cuáles son las dos formas de iniciar un proyecto administrado por Git? (0,25)
- j ¿En Git/GitHub qué cuatro áreas/fases existen? Describe cada una de ellas. (0,5)
- k ¿Cuándo utilizarías ramas/branches? (0,25)
- l ¿Cuál es la diferencia entre Git y GitHub? (0,25)
- m Explica con detalle en qué momento se puede dar un conflicto en Github. (0,25)

### Ejercicio 3 - RA4 Optimización - 1 punto.

**Crea la documentación en JavaDoc para la siguiente clase indicando lo siguiente:**

- La descripción general explicando que hace la clase Bolígrafo.
- En la descripción anterior añade también una etiqueta “en línea”.
- Tu nombre como autor de la clase.
- La versión de la clase con la fecha de hoy.
- La referencia a otra clase “Pincel” que se encuentra en el mismo paquete.
- Documenta lo que realizan los métodos.
- Los parámetros de entrada.
- Los parámetros devueltos por los métodos.
- Indica que el método recargar ya está obsoleto.
- Indica que el método recargar puede lanzar una excepción.

```
package Lienzos;
class Boligrafo
{
    protected int color=0;
    protected byte tinta=100;

    public bool pintar (byte gasto) {
        if (gasto>this.tinta)
            return false;
        this.tinta -= gasto;
        Console.WriteLine("Se gastaron {0} unidades de tinta.", gasto);
        return true;
    }
    public void recargar () {
        this.tinta=100;
        Console.WriteLine("Bolígrafo recargado");
    }
}
```

#### Ejercicio 4 - RA3 Pruebas - 3 puntos.

Este código calcula la media de 100 o menos números que son mayores o iguales que cero y que están en unos límites min y max.

1. Dibuja el grafo de flujo. (1 punto)
2. De las distintas maneras que tenemos de calcular la complejidad ciclomática, haz el cálculo de dos formas distintas. (1 punto)
3. Identifica cada uno de los caminos independientes y explica además por qué has llegado a esa solución. (1 punto)

```
public int media(int[] vector, int min, int max ){
    int suma = 0;
    int total_valido = 0;
    int i=0;

    while (vector[i]>=0 && i<100){
        if (vector[i] >=min && vector[i]<=max){
            total_valido++;
            suma+=vector[i];
        }
        i++;
    }
    if (total_valido > 0)
        media = suma/total_valido
    else
        media = -1
}
```

### **Ejercicio 5 - RA4 Optimización - 2 puntos.**

**Para cada uno de los siguientes apartados, escribe el/los comando/s implicados al trabajar con Git/GitHub o adjunta la captura de pantalla en el caso de que se pida.**

1. Una vez configurado Git, ¿qué comando tendrías que enviar para comprobar que tu username y tu email están correctamente configurados? (0,25)
2. Crea un nuevo repositorio de nombre “Repositorio-de-[Nombre]” dentro de tu usuario de Github donde [Nombre] se corresponde con tu nombre y apellidos. Adjunta una captura de pantalla del repositorio una vez que lo hayas creado. (0,25)
3. Escribe tal cual, el comando que tendrías que enviar en la consola para enlazar una carpeta local de tu equipo con el repositorio que acabas de crear en GitHub en el paso anterior. (0,25)
4. ¿Qué comando tendría que enviar para deshacer el último commit y los cambios anteriores del directorio de trabajo volviendo a la versión anterior del repositorio? (0,25)
5. Dentro del repositorio que creaste en el apartado 3, crea un tablero Kanban con las columnas To-Do, Doing, Test, y Done. Añade algunas tareas ficticias y adjunta la captura de pantalla (0,5)
6. Busca en GitHub un repositorio en el que te resulte llamativa las estadísticas relacionadas con los commits de los contribuidores al proyecto. Adjunta una captura de pantalla (0,5)