UNIVERSIDAD SIGLO 21

TRABAJO PRÁCTICO

Estudiante: Daiana Micaela Arena

Cursada: EDH

DNI: 38629115

El objetivo del primer trabajo práctico era generar el código de ambas funciones y presentar su complejidad en notación O.

Abordaré la problemática planteada al implementar los métodos requeridos: **"cantidadOficinasActivas()"** y **"encontrarPrimeraOficinaActiva()"**.

Para poder probar el código a la hora de resolverlo, fue necesario implementar el Main, Edificio y Oficina, las clases que enseñaré a continuación:

```java
import Exercises.TP1.src.edificio.Edificio;

You, 17 minutes ago | 1 author (You)
public class App {
    Run | Debug
    public static void main(String[] args) throws Exception {

        int o = 9;
        double s = 10;

        Edificio edificio = new Edificio(o:9, s:10);

        System.out.println(edificio.mostrar());
        System.out.println(edificio.cantidadoficinasactivas(o,s));
        System.out.println(edificio.encontrarprimeroficinaactiva(o, s).mostrar());
    }
}
```

```java
public class Edificio {

    private int numOficinas;
    private double superficie;

    //constructor
    public Edificio(int o, int s) {···

    public boolean obtenersensor(int p, int o){···


    public int cantidadoficinasactivas (int numOf, double superficie){···



    public Oficina encontrarprimeroficinaactiva(int numOf, double superficie){···



    public String mostrar() {···

}
```

```java
package Exercises.TP1.src.oficina;

...
public class Oficina {

    private int piso;
    private int oficina;

    public Oficina(int p, int o) {
        this.piso = p;
        this.oficina = o;
    }



     public String mostrar() {
        return "Oficina " + oficina + " del piso " + piso;
    }


}
```

Adicional a eso, desarrolle rápidamente la función, nombrada en el enunciado como "obtenersensor( )" para poder utilizarla en los siguientes apartados que explicaré en este TP. Para fines prácticos, solo arroja 'true' en aquellas oficinas de n° par:

```java
public boolean obtenersensor(int p, int o){

    return  (o%2==0);
}
```

Ahora sí, explicaré las funciones pedidas por el trabajo:

En primer lugar, ***public int cantidadoficinasactivas (int numOf, double superficie):***

```java
public int cantidadoficinasactivas (int numOf, double superficie){

    //devolverá la cantidad de oficinas activas del edificio (valor entre 0 y 90).
    int totalActivo = 0;
    int p = 1;
    int o = 1;

    do {

        boolean check = obtenersensor( p,  o);
        //System.out.println(" p: "+p+"o: " + o + " check: " +  check);
        if (check == true) {
            totalActivo++;
        }

        if (o < numOf){
            o++;
        }  else {
            p++;
            o=1;
        }

    } while (p ≤ superficie);


    return totalActivo;

}
```

A continuación, ***public Oficina encontrarprimeroficinaactiva(int numOf, double superficie):***

```java
public Oficina encontrarprimeroficinaactiva(int numOf, double superficie){
    //devolverá un objeto de la clase oficina (con los valores de piso y oficina)
    //que representará la primera oficina activa encontrada partiendo desde el primer piso.
    int p = 1;
    int o = 1;
    int pAct = 0;
    int oAct = 0;

    do {
        // llama a obtener sensor
        boolean check = obtenersensor( p,  o);
        if (check == true) {
              pAct = p;
              oAct = o;
              p = (int) superficie + 1;
        } else {
            if (o < numOf){
                o++;
            }  else {
                p++;
                o=1;
            }
        }

    } while (p ≤ superficie);


    Oficina primerActivo = new Oficina(pAct, oAct);
    return primerActivo;

}
```

La complejidad algorítmica de un ciclo do-while (como lo son ambas de mis funciones) ***se considera lineal, lo que se denota como O(n), donde "n" representa el número de iteraciones que se realizarán en el ciclo*** (en el caso de nuestra app N=90 ya que son 10 pisos con 9 oficinas cada uno). Esto significa que el tiempo de ejecución del algoritmo aumenta de manera proporcional al tamaño de la entrada o el número de veces que se ejecuta el ciclo (en nuestro caso, al aumentar oficinas o pisos).

Otra posible solución sería implementar dos ciclos for anidados uno por los departamentos y otro por los pisos, en cuyo caso la complejidad sería cuadrática. Es por esto que, en mi desarrollo, opté por la opción anterior.