

```

class ContadorArbol{
    public Integer contarOcurrencias(ArbolGeneral<Character> a,
    Character dato){
        // La cantidad de las ocurrencias en el arbol es:
        //   La cantidad de ocurrencias del HijoIzquierdo
        //       +
        //       1 (si mi raiz cumple)
        //       +
        //   la cantidad de ocurrencias de los hijos derechos

        Integer cantidadOcurrencias = 0;

        ListaGenerica<ArbolGeneral<Character>> hijos = a.getHijos();

        // Llamo contarOcurrencias con hijo izquierdo (si hay)
        if(!hijos.esVacia()){
            cantidadOcurrencias +=
this.contarOcurrencias(hijos.proximo(), dato);
        }

        // Cuento 1 si la raiz cumple condicion
        if(a.getDatoRaiz().equals(dato)){
            cantidadOcurrencias++;
        }

        //Llamo contarOcurrencias con el resto de los hijos(los derechos)
        while(!hijos.fin()){
            cantidadOcurrencias +=
this.contarOcurrencias(hijos.proximo(), dato);
        }

        return cantidadOcurrencias;
    }

    public Integer contarOcurrencias2(ArbolGeneral<Character> a,
    Character dato){
        int[] ocurrencias = {0};

        this.contarOcurrenciasConParametro(a, dato, ocurrencias);

        return ocurrencias;
    }

    -----

    private void
    contarOcurrenciasConParametro(ArbolGeneral<Character> a, Character
    dato, Integer[] cantidadOcurrencias){

        ListaGenerica<ArbolGeneral<Character>> hijos = a.getHijos();

```

```

        // Llamo contarOcurrencias con hijo izquierdo (si hay)
        if(!hijos.esVacia()){
            this.contarOcurrencias2(hijos.proximo(), dato,
cantidadOcurrencias);
        }

        // Cuento 1 si la raiz cumple condicion
        if(a.getDatoRaiz().equals(dato)){
            cantidadOcurrencias[0]++;
        }

        //Llamo contarOcurrencias con el resto de los hijos
        while(!hijos.fin()){
            this.contarOcurrencias(hijos.proximo(), dato,
cantidadOcurrencias);
        }
    }
}

```