

Universidad Mayor de San Andrés
Facultad de Ciencias Puras y Naturales



CARRERA DE INFORMÁTICA
PROYECTO – PRESENTACIÓN FINAL

Materia: INF-121 Programación II

Grupo: 4

Docente: Lic. Marcelo German Aruquipa Chambi

Datos Personales:

1. Elizabeth Suzaño Condori
2. Jhamil Calixto Mamani Quea
3. Walter Gustavo Flores Yujra

Tema: SISTEMA DE MONITOREO AMBIENTAL EN TIEMPO REAL

Fecha: 11 de junio de 2024

LA PAZ - BOLIVIA

INTRODUCCIÓN

El **sistema de Monitoreo Ambiental en Tiempo Real** es una herramienta crucial para comprender y abordar los desafíos relacionados con los desastres naturales y la contaminación ambiental. Su objetivo es recopilar datos ambientales de diversas fuentes para la **detección temprana de posibles desastres naturales o situaciones de contaminación**.

En esta situación nos enfocaremos en recopilar los datos ambientales del país de Bolivia.

En la actualidad, Bolivia enfrenta una serie de desafíos ambientales que requieren soluciones innovadoras y coordinadas. La región está expuesta a una variedad de desastres naturales, como terremotos, inundaciones, deslizamientos de tierra y huracanes, que pueden tener impactos devastadores en las comunidades y el medio ambiente. Además, la rápida urbanización y el crecimiento industrial han llevado a la contaminación del aire, agua y suelo, lo que representa una amenaza para la salud pública y el bienestar de la población.

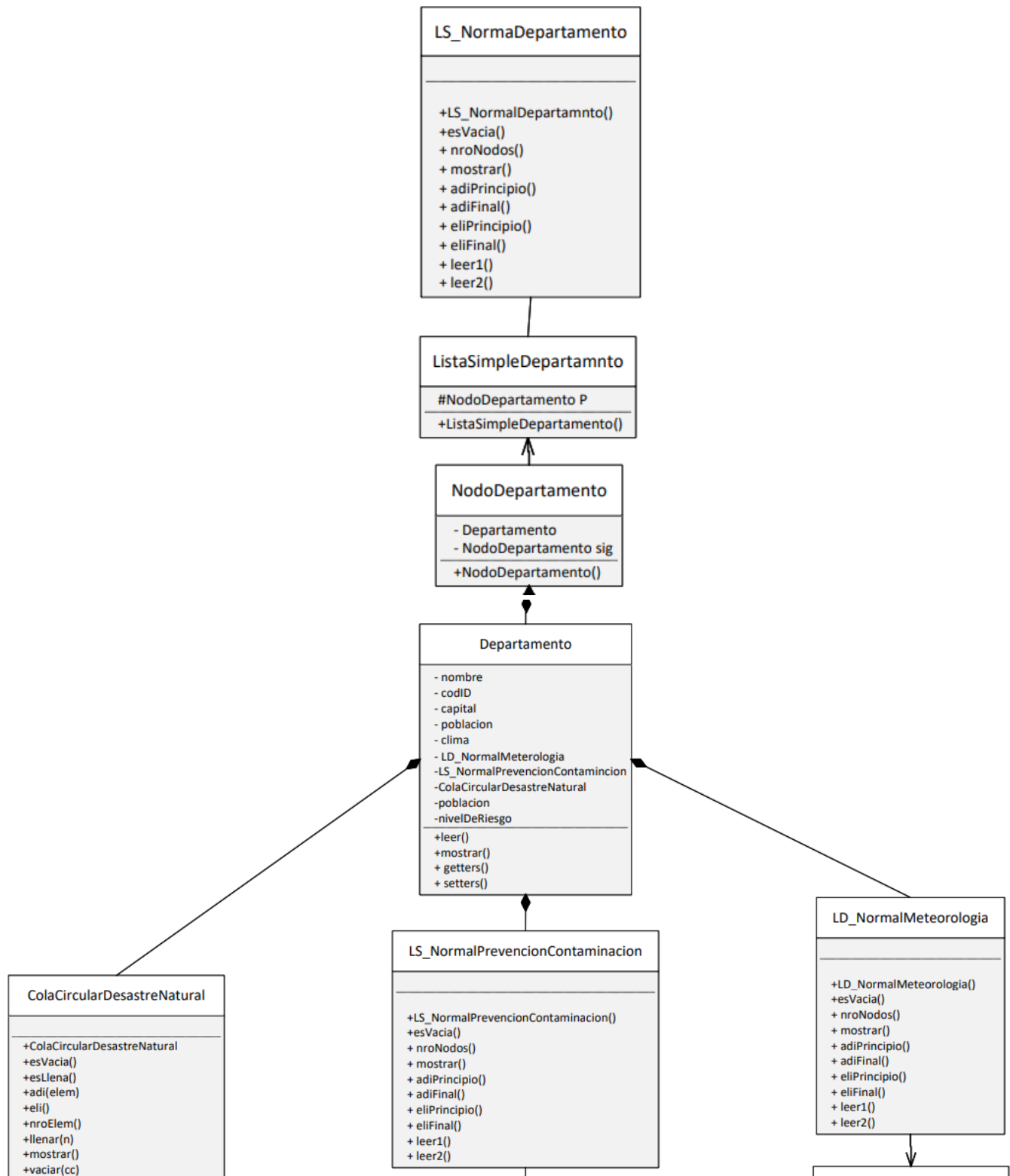
El sistema propuesto para la detección temprana de desastres naturales y situaciones de contaminación en Bolivia, se basa en la recopilación y análisis de datos en tiempo real. El sistema se enfoca en la prevención de desastres mediante la identificación de riesgos y vulnerabilidad, lo que permite una respuesta rápida y coordinada a emergencias ambientales.

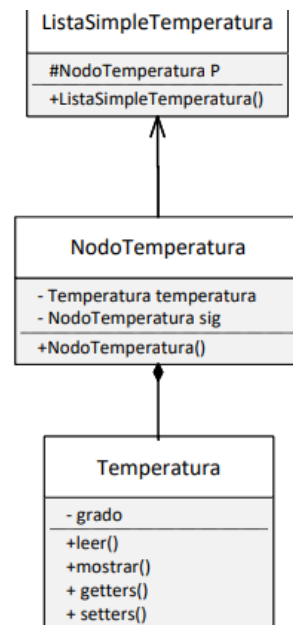
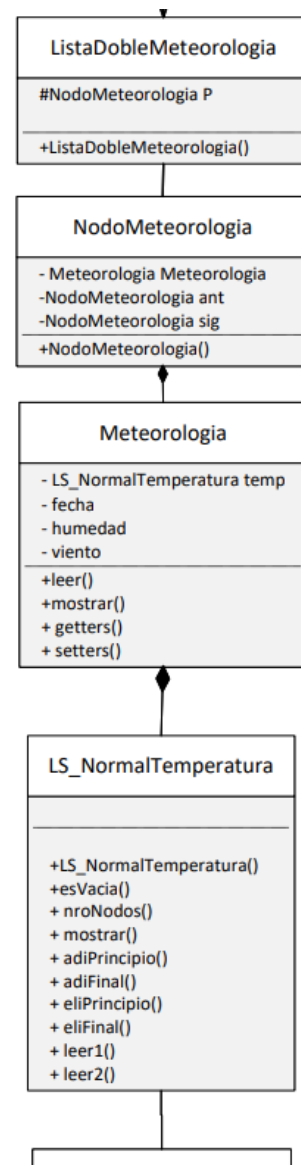
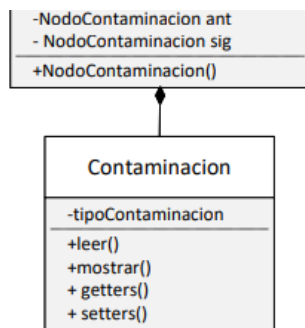
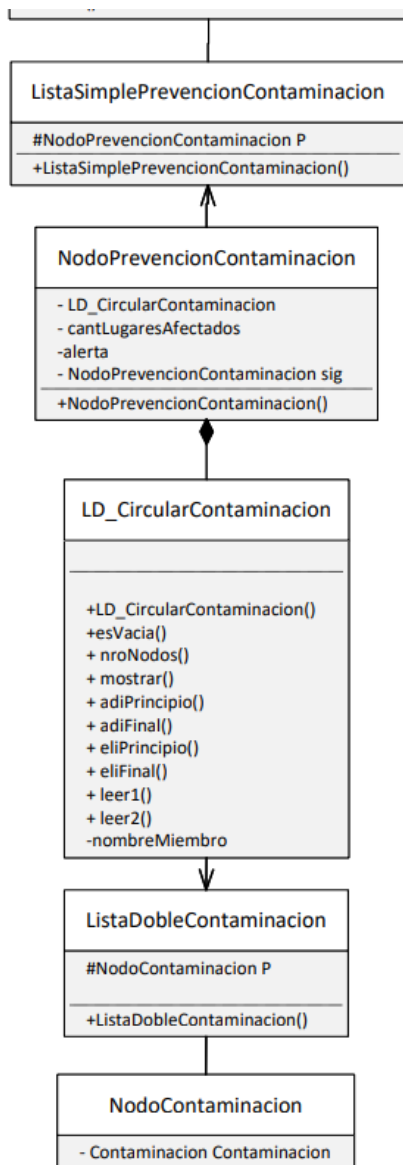
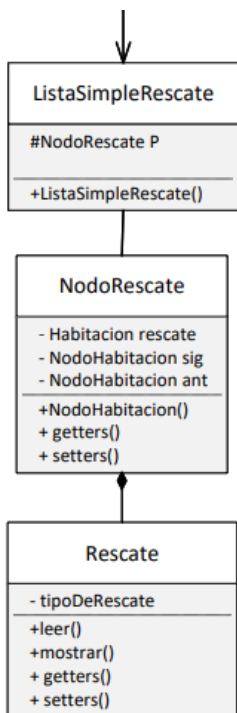
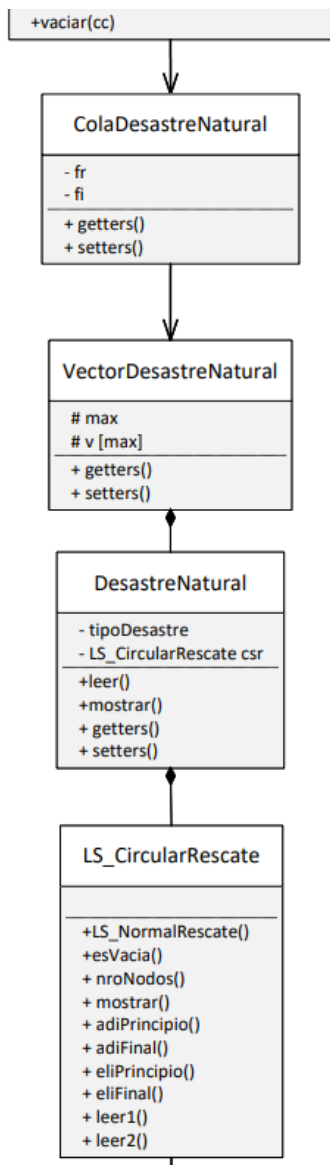
Los objetivos y problemas identificados reflejan la necesidad de abordar los desafíos ambientales en Bolivia de manera integral y proactiva. La detección temprana de desastres naturales y situaciones de contaminación es fundamental para minimizar los impactos negativos en la población y el medio ambiente. Además, el monitoreo de la calidad ambiental y la respuesta rápida a emergencias son aspectos clave para proteger la salud pública y promover el desarrollo sostenible.

El sistema propuesto también reconoce la importancia de la educación ambiental y la sensibilización pública en la conservación del medio ambiente. Al proporcionar información accesible y comprensible sobre los riesgos ambientales y los impactos de las actividades humanas, se puede fomentar una mayor conciencia y acción para la conservación ambiental.

En resumen, el Sistema de Monitoreo Ambiental en Tiempo Real es una herramienta esencial para la prevención y respuesta ante desastres naturales y la protección del medio ambiente.

DIAGRAMA DE CLASES DEFINITIVO





PANORAMA COMPLETO DEL DIAGRAMA DE CLASES



PRINCIPAL (CONSOLA)

//MAIN

```
288         case 4:
289             //4.    Mostrar el nombre y sus datos meteorológicos de los departamentos que
290             //        tienen el clima Y.
291             System.out.println("\n\t\t\t -- PROBLEMA 4 --");
292             System.out.println("\n\tDATOS METEOROLOGICOS DEL CLIMA (Y)");
293             String climaY="Min 17C/ Max 29C";
294             Mostrar_datos_Meteorologicos_de_climaY(Depa,climaY);
295             break;
296         case 5:
297             //5.    Mostrar el nombre de los departamentos que presenten más casos del tipo
298             //        de contaminación X junto con la cantidad de casos.
299             System.out.println("\n\t\t\t -- PROBLEMA 5 --");
300             String tipoContaminacion = "Suelo";
301             mostrarDepartamentoConMasCasosDecontaminacion(Depa, tipoContaminacion);
302             break;
303         default:
304             System.out.println("Opcion invalida.Intente de nuevo.");
305             continue;
306     }
307     System.out.println("Desea realizar otra operacion? (s/n)");
308     String respuesta = sc.next();
309     if(respuesta.equalsIgnoreCase("n")) {
310         break;
311     }
312 }
313 sc.close();
314
315 }
316
252     int q;
253     Scanner sc = new Scanner(System.in);
254     while(true) {
255         System.out.println("\n");
256         System.out.println("=====RESOLUCION DE LOS PROBLEMAS=====");
257         System.out.println("[1] CALCULAR EL PROMEDIO DE TEMPEREATURAS DE LOS DEPARTAMENTOS DE BOLIVIA");
258         System.out.println("[2] DADA UN DEPARTAMENTO X ORDENA SUS TEMPERATURAS DE MAYOR A MENOR");
259         System.out.println("[3] MOSTRAR EL NOMBRE DE LOS DEPARTAMENTOS QUE PRESENTARON EL DESASTRE NATURAL X ");
260         System.out.println("    Y A LA VEZ MOSTRAR SUS DATOS METEOROLOGICOS");
261         System.out.println("[4] MOSTRAR EL NOMBRE Y SUS DATOS METEOROLOGICOS DE LOS DEPARTAMENTOS QUE TIENEN EL ");
262         System.out.println("    CLIMA Y");
263         System.out.println("[5] MOSTRAR EL NOMBRE Y LA CANTIDAD DE CASOS DE CONTAMINACION DE LOS DEPARTAMENTOS ");
264         System.out.println("    QUE PRESENTEN MAS CASOS DEL TIPO DE CONTAMINACION X ");
265         System.out.println("Introduce opcion: ");
266         q = sc.nextInt();
267         switch(q) {
268             case 1:
269                 //1.    Calcular el promedio de temperaturas de los departamentos de Bolivia.
270                 System.out.println("\n\t\t\t -- PROBLEMA 1 --");
271                 double promedio = promedioTemperatura(Depa);
272                 System.out.println("El promedio es: "+promedio);
273                 break;
274             case 2:
275                 //2.    Dado un departamento X ordena sus temperaturas de mayor a menor.
276                 System.out.println("\n\t\t\t -- PROBLEMA 2 --");
277                 System.out.println("\nORDENAR SUS TEMPERATURAS DE MAYOR A MENOR DEL DEPARTAMENTO X");
278                 String deparX="Beni";
279                 ordenar_Temperaturas_de_Mayor_Menor(Depa,deparX);
280                 break;
281             case 3:
282                 //3.    Mostrar el nombre de los departamentos que presentaron el desastre natural X
283                 //        y a la vez mostrar sus datos meteorológicos.
284                 System.out.println("\n\t\t\t -- PROBLEMA 3 --");
285                 mostrarDepartamentosDesastreNaturalX(Depa, "Terremoto");
286                 break;
287             case 4:
288                 //4.    Mostrar el nombre y sus datos meteorológicos de los departamentos que
289                 //        tienen el clima Y.
290                 System.out.println("\n\t\t\t -- PROBLEMA 4 --");
291                 System.out.println("\n\tDATOS METEOROLOGICOS DEL CLIMA (Y)");
292                 String climaY="Min 17C/ Max 29C";
293                 Mostrar_datos_Meteorologicos_de_climaY(Depa,climaY);
294                 break;
295             case 5:
296                 //5.    Mostrar el nombre de los departamentos que presenten más casos del tipo
297                 //        de contaminación X junto con la cantidad de casos.
298                 System.out.println("\n\t\t\t -- PROBLEMA 5 --");
299                 String tipoContaminacion = "Suelo";
300                 mostrarDepartamentoConMasCasosDecontaminacion(Depa, tipoContaminacion);
301                 break;
302             default:
303                 System.out.println("Opcion invalida.Intente de nuevo.");
304                 continue;
305         }
306     }
307     System.out.println("Desea realizar otra operacion? (s/n)");
308     String respuesta = sc.next();
309     if(respuesta.equalsIgnoreCase("n")) {
310         break;
311     }
312 }
313 sc.close();
314
315 }
```

// FUNCIONES Y PROCEDIMIENTOS USADOS PARA RESOLVER LOS PROBLEMAS PLANTEADOS

```
317
318● static void mostrarSoloMeteorologia(LS_NormalDepartamento a) {
319     NodoDepartamento R = a.getP();
320     while(R != null) {
321         NodoMeteorologia S = R.getDepartamento().getMet().getP();
322         System.out.println("\t*-*-*-*-*-* DEPARTAMENTO (" + R.getDepartamento().getNombre() + ") *-*-*-*-*-*");
323         while(S != null) {
324             S.getMeteorologia().mostrar();
325             S = S.getSig();
326         }
327         R = R.getSig();
328     }
329 }
330 //PROBLEMA 1
331● static double promedioTemperatura(LS_NormalDepartamento a) {
332     NodoDepartamento R = a.getP();
333     double cont = 0;
334     double sum = 0;
335     double prom = 0;
336     while(R != null) {
337         NodoMeteorologia S = R.getDepartamento().getMet().getP();
338         while(S != null) {
339             NodoTemperatura T = S.getMeteorologia().getTemp().getP();
340             while(T != null) {
341                 sum = sum + T.getTemperatura().getGrado();
342                 cont++;
343                 T = T.getSig();
344             }
345             S = S.getSig();
346         }
347         R = R.getSig();
348     }
349     prom = sum/cont;
350     return prom;
351 }
352
353 //PROBLEMA 2
354● private static void ordenar_Temperaturas_de_Mayor_Menor(LS_NormalDepartamento a, String deparX) {
355     NodoDepartamento R = a.getP();
356     System.out.println("Departamento X: " + deparX);
357     while (R != null) {
358         Departamento de = R.getDepartamento();
359         if (de.getNombre().equals(deparX)) {
360             NodoMeteorologia S = de.getMet().getP();
361             while (S != null) {
362                 Meteorologia me = S.getMeteorologia();
363                 LS_NormalTemperatura aux1 = new LS_NormalTemperatura();
364                 LS_NormalTemperatura aux2 = new LS_NormalTemperatura();
365                 while (!me.getTemp().esVacia()) {
366                     double mayor = mayorTemperratura(me.getTemp());
367                     while (!me.getTemp().esVacia()) {
368                         NodoTemperatura x = me.getTemp().eliPrincipio();
369                         if (x.getTemperatura().getGrado() == mayor) {
370                             aux1.adifinal(x.getTemperatura());
371                         } else {
372                             aux2.adifinal(x.getTemperatura());
373                         }
374                     }
375                     me.getTemp().setP(aux2.getP());
376                     aux2.setP(null);
377                 }
378                 me.getTemp().setP(aux1.getP());
379                 S = S.getSig();
380                 System.out.println();
381                 me.mostrar();
382             }
383         }
384         R = R.getSig();
385     }
386 }
```

```

387● private static double mayorTemperratura(LS_NormalTemperatura a) {
388     NodoTemperatura R = a.getP();
389     double may = Double.NEGATIVE_INFINITY;
390     while (R != null) {
391         Temperatura tem = R.getTemperatura();
392         if (tem.getGrado() > may) {
393             may = tem.getGrado();
394         }
395         R = R.getSig();
396     }
397     return may;
398 }
399
400 //PROBLEMA 3
401● static void mostrarDepartamentosDesastreNaturalX(LS_NormalDepartamento a, String desNatX) {
402     NodoDepartamento R = a.getP();
403     boolean sw = false;
404     System.out.println("Desastre natural buscado => "+desNatX);
405     System.out.println("Departamentos encontrados...");
406     System.out.println();
407     while(R != null) {
408         ColaCircularDesastreNatural z = R.getDepartamento().getDesastreNatural();
409         ColaCircularDesastreNatural aux = new ColaCircularDesastreNatural();
410         while(!z.esVacia()) {
411             DesastreNatural des = z.eli();
412             if(des.getTipoDesastre().equals(desNatX)) {
413                 sw = true;
414                 System.out.println(R.getDepartamento().getNombre());
415                 System.out.println("Desastre Natural: "+des.getTipoDesastre());
416                 System.out.println("\t-* Rescate de (" +des.getTipoDesastre()+") -*");
417                 des.getCsr().mostrar();
418                 System.out.println("-----");
419             }
420             aux.adi(des);
421         }
422         z.vaciar(aux);
423         R = R.getSig();
424     }
425     if(sw == false) {
426         System.out.println("El desastre natural no se encuentra en el registro...");
427     }
428 }
429

```

```

430 //PROBLEMA 4
431● private static void Mostrar_datos Meteorologicos_de_climaY(LS_NormalDepartamento a, String climaY) {
432     NodoDepartamento R = a.getP();
433     while (R!=null) {
434         Departamento de = R.getDepartamento();
435         if (de.getClima().equals(climaY)) {
436             System.out.println("\n\tEL DEPARTAMENTO QUE TIENE ESE CLIMA ES [" +de.getNombre()+"]");
437             de.getMet().mostrar();
438         }
439         R=R.getSig();
440     }
441 }
442 //PROBLEMA 5
443● public static void mostrarDepartamentoConMasCasosDecontaminacion(LS_NormalDepartamento a, String tipoContaminacion) {
444     NodoDepartamento R = a.getP();
445     int may=0;
446     String depMayor="";
447     while (R!=null) {
448         Departamento de = R.getDepartamento();
449         int cantCont = contar_Casos_Contaminacion(de,tipoContaminacion);
450         System.out.println("Conta "+cantCont+" depa "+de.getNombre());
451         if (cantCont>may) {
452             depMayor=de.getNombre();
453             may=cantCont;
454         }
455         R=R.getSig();
456     }
457     System.out.println("El departamento que tiene mas contaminacion de "+tipoContaminacion+" es "+depMayor+"
458     + " con la cantidad de => "+may);
459 }

```



```

460● private static int contar_Casos_Contaminacion(Departamento de, String tipoContaminacion) {
461     NodoPrevencionContaminacion R = de.getLSprev().getP();
462     int cont=0;
463     while (R!=null) {
464         NodoContaminacion S = R.getLDc().getP();
465         int nroN=R.getLDc().nroNodos();
466         for (int i = 0; i < nroN; i++) {
467             Contaminacion co = S.getContaminacion();
468             if (co.getTipoContaminacion().equals(tipoContaminacion)) {
469                 cont++;
470             }
471             S=S.getSig();
472         }
473         R=R.getSig();
474     }
475     return cont;
476 }
477
478
479 }

```

PANTALLAS DE EJECUCIÓN

```

=====RESOLUCION DE LOS PROBLEMAS=====
[1] CALCULAR EL PROMEDIO DE TEMPEREATURAS DE LOS DEPARTAMENTOS DE BOLIVIA
[2] DADA UN DEPARTAMENTO X ORDENA SUS TEMPERATURAS DE MAYOR A MENOR
[3] MOSTRAR EL NOMBRE DE LOS DEPARTAMENTOS QUE PRESENTARON EL DESASTRE NATURAL X
    Y A LA VEZ MOSTRAR SUS DATOS METEOROLOGICOS
[4] MOSTRAR EL NOMBRE Y SUS DATOS METEOROLOGICOS DE LOS DEPARTAMENTOS QUE TIENEN EL
    CLIMA Y
[5] MOSTRAR EL NOMBRE Y LA CANTIDAD DE CASOS DE CONTAMINACION DE LOS DEPARTAMENTOS
    QUE PRESENTEN MAS CASOS DEL TIPO DE CONTAMINACION X
Introduce opcion:
1

-- PROBLEMA 1 --

El promedio es: 10.5
Desea realizar otra operacion? (s/n)
s

```

```

=====RESOLUCION DE LOS PROBLEMAS=====
[1] CALCULAR EL PROMEDIO DE TEMPEREATURAS DE LOS DEPARTAMENTOS DE BOLIVIA
[2] DADA UN DEPARTAMENTO X ORDENA SUS TEMPERATURAS DE MAYOR A MENOR
[3] MOSTRAR EL NOMBRE DE LOS DEPARTAMENTOS QUE PRESENTARON EL DESASTRE NATURAL X
    Y A LA VEZ MOSTRAR SUS DATOS METEOROLOGICOS
[4] MOSTRAR EL NOMBRE Y SUS DATOS METEOROLOGICOS DE LOS DEPARTAMENTOS QUE TIENEN EL
    CLIMA Y
[5] MOSTRAR EL NOMBRE Y LA CANTIDAD DE CASOS DE CONTAMINACION DE LOS DEPARTAMENTOS
    QUE PRESENTEN MAS CASOS DEL TIPO DE CONTAMINACION X
Introduce opcion:
2

-- PROBLEMA 2 --

ORDENAR SUS TEMPERATURAS DE MAYOR A MENOR DEL DEPARTAMENTO X
Departamento X: Beni

    -* DATOS METEOROLOGIGOS -*
Fecha: 03/02/2023
    -- Temperaturas registradas en este mes --
Temperatura [21.0 grados]
Temperatura [18.0 grados]
Temperatura [8.0 grados]
Temperatura [3.0 grados]
Temperatura [-5.0 grados]
Temperatura [-16.0 grados]
Humedad: 20%
Viento: 32m/s

```

```
        -* DATOS METEOROLOGIGOS -*
Fecha: 15/03/2023
        -- Temperaturas registradas en este mes --
Temperatura [19.0 grados]
Temperatura [14.0 grados]
Temperatura [2.0 grados]
Temperatura [-11.0 grados]
Humedad: 80%
Viento: 17m/s
```

```
        -* DATOS METEOROLOGIGOS -*
Fecha: 26/04/2023
        -- Temperaturas registradas en este mes --
Temperatura [18.0 grados]
Temperatura [-5.0 grados]
Humedad: 10%
Viento: 50m/s
Desea realizar otra operacion? (s/n)
s
```

```
=====RESOLUCION DE LOS PROBLEMAS=====
[1] CALCULAR EL PROMEDIO DE TEMPEREATURAS DE LOS DEPARTAMENTOS DE BOLIVIA
[2] DADA UN DEPARTAMENTO X ORDENA SUS TEMPERATURAS DE MAYOR A MENOR
[3] MOSTRAR EL NOMBRE DE LOS DEPARTAMENTOS QUE PRESENTARON EL DESASTRE NATURAL X
    Y A LA VEZ MOSTRAR SUS DATOS METEOROLOGICOS
[4] MOSTRAR EL NOMBRE Y SUS DATOS METEOROLOGICOS DE LOS DEPARTAMENTOS QUE TIENEN EL
    CLIMA Y
[5] MOSTRAR EL NOMBRE Y LA CANTIDAD DE CASOS DE CONTAMINACION DE LOS DEPARTAMENTOS
    QUE PRESENTEN MAS CASOS DEL TIPO DE CONTAMINACION X
```

Introduce opcion:

3

-- PROBLEMA 3 --

Desastre natural buscado => Terremoto
Departamentos encontrados...

La Paz

Desastre Natural: Terremoto

-* Rescate de (Terremoto) -*

Rescate [tipoRescate=Rescate en estructuras colapsadas]
Rescate [tipoRescate=Rescate con perros de busqueda]
Rescate [tipoRescate=Rescate en espacios confinados]

Chuquisaca

Desastre Natural: Terremoto

-* Rescate de (Terremoto) -*

Rescate [tipoRescate=Rescate en estructuras colapsadas]
Rescate [tipoRescate=Rescate con perros de busqueda]
Rescate [tipoRescate=Rescate en espacios confinados]

Santa Cruz

Desastre Natural: Terremoto

-* Rescate de (Terremoto) -*

Rescate [tipoRescate=Rescate en estructuras colapsadas]
Rescate [tipoRescate=Rescate con perros de busqueda]
Rescate [tipoRescate=Rescate en espacios confinados]

Desea realizar otra operacion? (s/n)

s

```

=====RESOLUCION DE LOS PROBLEMAS=====
[1] CALCULAR EL PROMEDIO DE TEMPEREATURAS DE LOS DEPARTAMENTOS DE BOLIVIA
[2] DADA UN DEPARTAMENTO X ORDENA SUS TEMPERATURAS DE MAYOR A MENOR
[3] MOSTRAR EL NOMBRE DE LOS DEPARTAMENTOS QUE PRESENTARON EL DESASTRE NATURAL X
    Y A LA VEZ MOSTRAR SUS DATOS METEOROLOGICOS
[4] MOSTRAR EL NOMBRE Y SUS DATOS METEOROLOGICOS DE LOS DEPARTAMENTOS QUE TIENEN EL
    CLIMA Y
[5] MOSTRAR EL NOMBRE Y LA CANTIDAD DE CASOS DE CONTAMINACION DE LOS DEPARTAMENTOS
    QUE PRESENTEN MAS CASOS DEL TIPO DE CONTAMINACION X
Introduce opcion:
4

```

```

-- PROBLEMA 4 --

```

```

    DATOS METOROLOGICOS DEL CLIMA (Y)

    EL DEPARTAMENTO QUE TIENE ESE CLIMA ES [Beni]
    -* DATOS METEOROLOGIGOS -*
Fecha: 03/02/2023
    -- Temperaturas registradas en este mes --
Temperatura [21.0 grados]
Temperatura [18.0 grados]
Temperatura [8.0 grados]
Temperatura [3.0 grados]
Temperatura [-5.0 grados]
Temperatura [-16.0 grados]
Humedad: 20%
Viento: 32m/s

```

```

    -* DATOS METEOROLOGIGOS -*
Fecha: 15/03/2023
    -- Temperaturas registradas en este mes --
Temperatura [19.0 grados]
Temperatura [14.0 grados]
Temperatura [2.0 grados]
Temperatura [-11.0 grados]
Humedad: 80%
Viento: 17m/s

```

```

    -* DATOS METEOROLOGIGOS -*
Fecha: 26/04/2023
    -- Temperaturas registradas en este mes --
Temperatura [18.0 grados]
Temperatura [-5.0 grados]
Humedad: 10%
Viento: 50m/s

```

```

Desea realizar otra operacion? (s/n)
s

```

```

=====RESOLUCION DE LOS PROBLEMAS=====
[1] CALCULAR EL PROMEDIO DE TEMPEREATURAS DE LOS DEPARTAMENTOS DE BOLIVIA
[2] DADA UN DEPARTAMENTO X ORDENA SUS TEMPERATURAS DE MAYOR A MENOR
[3] MOSTRAR EL NOMBRE DE LOS DEPARTAMENTOS QUE PRESENTARON EL DESASTRE NATURAL X
    Y A LA VEZ MOSTRAR SUS DATOS METEOROLOGICOS
[4] MOSTRAR EL NOMBRE Y SUS DATOS METEOROLOGICOS DE LOS DEPARTAMENTOS QUE TIENEN EL
    CLIMA Y
[5] MOSTRAR EL NOMBRE Y LA CANTIDAD DE CASOS DE CONTAMINACION DE LOS DEPARTAMENTOS
    QUE PRESENTEN MAS CASOS DEL TIPO DE CONTAMINACION X
Introduce opcion:
5

```

```

-- PROBLEMA 5 --

```

```

conta 4 depa La Paz
conta 7 depa Beni
conta 3 depa Chuquisaca
conta 4 depa Santa Cruz
El departamento que tiene mas contaminacion de Suelo es Beni con la cantidad de => 7
Desea realizar otra operacion? (s/n)
n

```

PRINCIPAL (INTERFAZ)

PÁGINA PRINCIPAL DE LA INTERFAZ

SISTEMA DE MONITOREO AMBIENTAL EN TIEMPO REAL

Datos de los departamentos

Elige una opción:

- Opcion 1
- Opcion 2
- Opcion 3
- Opcion 4
- Opcion 5

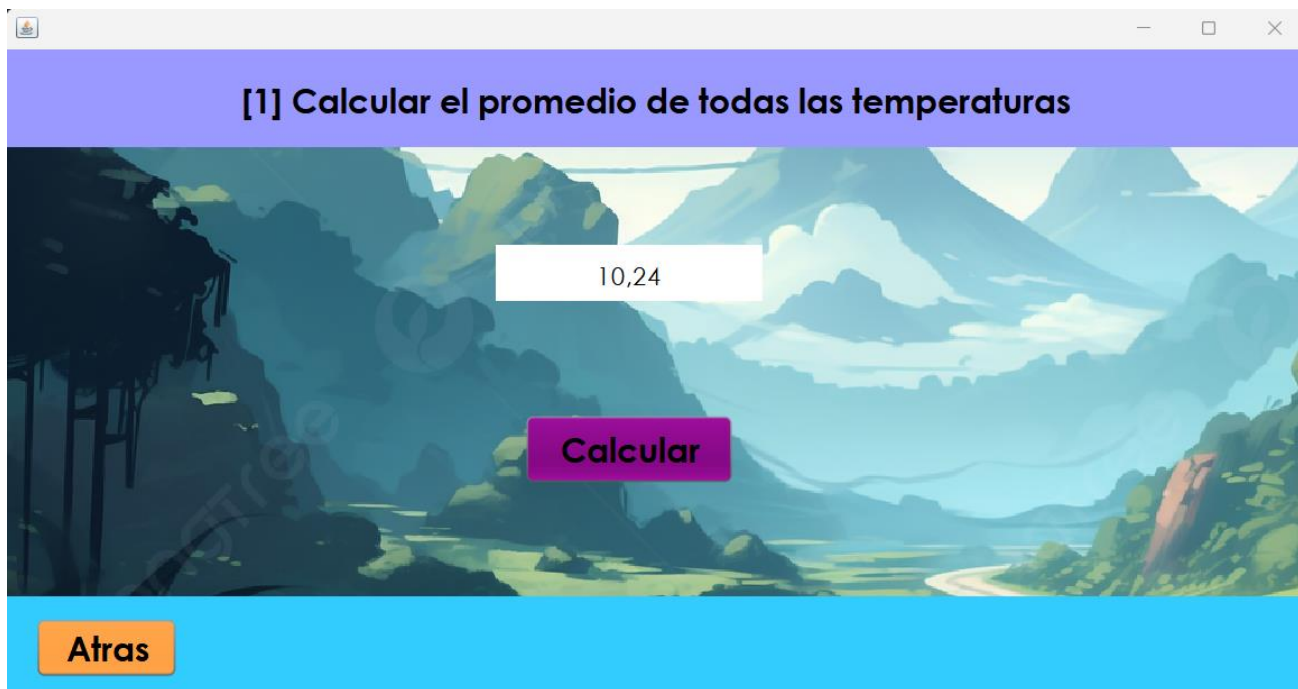
Nombre	Cod Id	Capital	Poblacion	Clima	Niv. Riesgo
La Paz	Lp02	La Paz	2000	Min 5C/Max 21C	Alto
Beni	Ben08	Trinidad	1050	Min 17C/Max 29C	Medio
Chuquisaca	Ch01	Sucre	1500	Min -6C/Max 16C	Alto
Santa Cruz	Scz07	Santa Cruz de la...	2500	Min 16C/Max 28C	Bajo
Pando	Pnd09	Cobija	1100	Min 8C/Max 32C	Alto
Tarija	Tja06	Tarija	4300	Min 9C/Max 25C	Medio
Cochabamba	Cbba03	Cochabamba	1780	Min 4C/Max 22C	Medio
Oruro	Or04	Oruro	4940	Min -6C/Max 16C	Alto
Potosi	Pt05	Potosi	2600	Min -3C/Max 4C	Bajo

Llenar Datos Mostrar Siguiete Agregar Datos Salir

CÓDIGO DE LA FUNCIÓN/PROCEDIMIENTO 1

```
private void botonCalcularActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    NodoDepartamento R = v1.getDep().getP();  
    DecimalFormat df = new DecimalFormat("#.00");  
    double cont = 0;  
    double sum = 0;  
    double prom = 0;  
    while(R != null){  
        NodoMeteorologia S = R.getDepartamento().getMet().getP();  
        while(S != null){  
            NodoTemperatura T = S.getMetereologia().getTemp().getP();  
            while(T != null){  
                sum = sum + T.getTemperatura().getGrado();  
                cont++;  
                T = T.getSig();  
            }  
            S = S.getSig();  
        }  
        R = R.getSig();  
    }  
    prom = sum/cont;  
    this.campoPromedio.setText(String.format("%.2f", prom));  
}
```

EJECUCIÓN DEL PROGRAMA 1



CÓDIGO DE LA FUNCIÓN/PROCEDIMIENTO 2

```
private void botonOrdenarActionPerformed(java.awt.event.ActionEvent evt) {  
164 // TODO add your handling code here:  
165 try {  
166     String deparX = campoDepartamento.getText();  
167     // int dx = Integer.parseInt(deparX);  
168     NodoDepartamento R = v1.getDep().getP();  
169     int sw = 0;  
170     while (R != null) {  
171         Departamento de = R.getDepartamento();  
172         if (de.getNombre().equals(deparX)) {  
173             sw = 1;  
174             NodoMeteorologia S = de.getMet().getP();  
175             while (S != null) {  
176                 Meteorologia me = S.getMetereologia();  
177                 LS_NormalTemperatura aux1 = new LS_NormalTemperatura();  
178                 LS_NormalTemperatura aux2 = new LS_NormalTemperatura();  
179                 aux1.setTemperatura(me.getTemperatura());  
180                 aux2.setTemperatura(me.getTemperatura());  
181                 lista.add(aux1);  
182                 lista.add(aux2);  
183                 S = S.getP();  
184             }  
185             R = R.getP();  
186         }  
187     }  
188     double suma = 0;  
189     for (int i = 0; i < lista.size(); i++) {  
190         suma += lista.get(i).getTemperatura();  
191     }  
192     double promedio = suma / lista.size();  
193     txtResultado.setText(String.valueOf(promedio));  
194 }  
195 }  
196 }
```



```

179         while (!me.getTemp().esVacia()) {
180             double mayor = mayorTemperatura(me.getTemp());
181             while (!me.getTemp().esVacia()) {
182                 NodoTemperatura x = me.getTemp().eliPrincipio();
183                 if (x.getTemperatura().getGrado() == mayor) {
184                     aux1.adiFinal(x.getTemperatura());
185                 } else {
186                     aux2.adiFinal(x.getTemperatura());
187                 }
188             }
189             me.getTemp().setP(aux2.getP());
190             aux2.setP(null);
191         }
192         me.getTemp().setP(aux1.getP());
193         S = S.getSig();
194         System.out.println();
195         me.mostrar();
196     }

```

```

197     }
198     R = R.getSig();
199 }
200 if(sw == 1){
201     JOptionPane.showMessageDialog(null, "Temperaturas ordenadas exitosamente");
202 }
203 else{
204     JOptionPane.showMessageDialog(null, "No se encontro el departamento!!");
205 }
206
207
208 } catch (Exception e) {
209     JOptionPane.showMessageDialog(null, "Insertar datos validos!!");
210 }
211
212 }
213

```

EJECUCIÓN DEL PROGRAMA 2

[2] Dado un departamento X ordenar sus temperaturas de mayor a menor

Ingresa departamento:

Ordenar

.* DATOS METEOROLOGICOS *.

Departamento: Tarija

Fecha: 01/01/2023

.* Temperaturas registradas -*

Temperatura: 30.0grados

Temperatura: 11.0grados

Temperatura: 10.0grados

Temperatura: 0.0grados

Temperatura: -13.0grados

Temperatura: -18.0grados

Humedad: 32%

Viento: 9m/s

Mostrar

Atras

CÓDIGO DE LA FUNCIÓN/PROCEDIMIENTO 3

```

147 private void botonMostrarDesasActionPerformed(java.awt.event.ActionEvent evt) {
148     // TODO add your handling code here:
149     try {
150         String desNatX = campoDesastreNatural.getText();
151         String titulos[] = {"*- Departamentos encontrados -*"};
152         DefaultTableModel pantalla = new DefaultTableModel(null, titulos);
153         NodoDepartamento R = v1.getDep().getP();
154         int sw = 0;
155         while(R != null){
156             ColaCircularDesastreNatural aux = new ColaCircularDesastreNatural();
157             while(!R.getDepartamento().getDesastreNatural().esVacia()){
158                 DesastreNatural x = R.getDepartamento().getDesastreNatural().eli();

```

```

158                 String dn = x.getTipoDesastre();
159                 if(dn.equals(desNatX)){
160                     sw = 1;
161                     String depa[] = {"Departamento: "+R.getDepartamento().getNombre()};
162                     pantalla.addRow(depa);
163                     String desastreNatural[] = {"Desastre Natural: "+dn};
164                     pantalla.addRow(desastreNatural);
165                     String tituloResc[] = {"-- Rescate de (" + dn + ") --"};
166                     pantalla.addRow(tituloResc);
167                     NodoRescate S = x.getCsr().getP();
168                     while(S.getSig() != x.getCsr().getP()){

```

```

169                         String resc1[] = {"Tipo de Rescate: "+
170                             S.getRescate().getTipoRescate()};
171                         pantalla.addRow(resc1);
172                         S = S.getSig();
173                     }
174                     String resc2[] = {"Tipo de Rescate: "+
175                         S.getRescate().getTipoRescate()};
176                     pantalla.addRow(resc2);
177                     String espacio[] = {" "};
178                     pantalla.addRow(espacio);
179                 }
180                 aux.adi(x);
181             }
182             R.getDepartamento().getDesastreNatural().vaciar(aux);
183             R = R.getSig();
184         }
185         pantalla_desastreNatural.setModel(pantalla);
186         if(sw == 0){
187             JOptionPane.showMessageDialog(null, "Desastre natural no encontrado!!");
188         }
189     }

```

```

190     } catch (Exception e) {
191         JOptionPane.showMessageDialog(null, "Insertar datos validos!!");
192     }
193 }
194

```

[3] Mostrar los departamentos que presentaron el desastre natural X

Ingresa desastre natural:

. Departamentos encontrados -

Departamento: La Paz
Desastre Natural: Inundacion
-- Rescate de (Inundacion) --
Tipo de Rescate: Rescate acuatico
Tipo de Rescate: Rescate con botes
Tipo de Rescate: Rescate aereo
Departamento: Tarija
Desastre Natural: Inundacion
-- Rescate de (Inundacion) --
Tipo de Rescate: Rescate acuatico
Tipo de Rescate: Rescate con botes

CÓDIGO DE LA FUNCIÓN/PROCEDIMIENTO 4

```
private void botonMostrarCliActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        String nuevoclimaX;  
        nuevoclimaX=campo_ClimaX.getText();  
        if (!nuevoclimaX.equals("")) {  
            String titulos [] = {"DEPARTAMENTO","FECHA","TEMPERATURA","HUMEDAD","VIENTO"};  
            DefaultTableModel pantalla = new DefaultTableModel(null,titulos);  
            NodoDepartamento R = v1.getDep().getP();  
            while (R!=null) {  
                Departamento de = R.getDepartamento();  
                if (de.getClima().equals(nuevoclimaX)) {  
                    System.out.println("\n\tEL DEPARTAMENTO QUE TIENE ESE CLIMA ES ["  
                        +de.getNombre()+"]");  
                    de.getMet().getP().getMetereologia().mostrar();  
  
                    String fila [] = {de.getNombre(),de.getMet().getP().  
                        getMetereologia().getFecha(),  
                        Integer.toString((int)de.getMet().getP().getMetereologia().getTemp()  
                            .getP().getTemperatura().getGrado())  
                        ,Integer.toString(de.getMet().getP().getMetereologia().getHumedad())  
                        ,de.getMet().getP().getMetereologia().getViento()};  
                    pantalla.addRow(fila);  
                }  
                R=R.getSig();  
            }  
            Pantalla_Clima.setModel(pantalla);  
  
        }else{  
            JOptionPane.showMessageDialog(null, "Datos no validos!!");  
        }  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, "No se encontraron datos");  
    }  
}
```


EJECUCIÓN DEL PROGRAMA 4

[4] Mostrar los departamentos que tienen el clima Y

Ingrese clima:

Min -6C/Max 16C

Mostrar

Departamentos encontrados

DEPARTAM...	FECHA	TEMPERAT...	HUMEDAD	VIENTO
Chuquisaca	12/01/2023	21	63	10m/s
Oruro	08/04/2023	14	9	5m/s

Atras

CÓDIGO DE LA FUNCIÓN/PROCEDIMIENTO 5

```
154 private int contar_Casos_Contaminacion(Departamento de, String tipoContaminacion) {
155     NodoPrevencionContaminacion R = de.getLSprev().getP();
156     int cont=0;
157     while (R!=null) {
158         NodoContaminacion S = R.getLDc().getP();
159         //int nroN=R.getLDc().nroNodos();
160
161         for (int i = 0; i < 10; i++) {
162             Clases.Contaminacion co = S.getContaminacion();
163             if (co.getTipoContaminacion().equals(tipoContaminacion)) {
164                 cont++;
165             }
166             S=S.getSig();
167         }
168         R=R.getSig();
169     }
170     return cont;
171 }
```

```

private void botonMostrarContActionPerformed(java.awt.event.ActionEvent evt) {
173     // TODO add your handling code here:
174     try {
175         String nuevacontamiX;
176         nuevacontamiX=campoContaX.getText();
177         if (!nuevacontamiX.equals("")) {
178             String titulos [] = {"DEPARTAMENTO","CANTIDAD DE CONTAMINACION"};
179             DefaultTableModel pantalla = new DefaultTableModel(null,titulos);
180             NodoDepartamento R = vl.getDep().getP();
181             int may=0;
182             String depMayor="";
183             while (R!=null) {
184                 Departamento de = R.getDepartamento();
185                 int cantCont = contar_Casos_Contaminacion(de,nuevacontamiX);
186                 String fila[]={de.getNombre(),Integer.toString(cantCont)};
187                 System.out.println("conta "+cantCont+" depa "+de.getNombre());
188                 if (cantCont>may) {
189                     depMayor=de.getNombre();
190                     may=cantCont;
191                 }
192                 pantalla.addRow(fila);
193                 R=R.getSig();
194             }
195             Pantalla_Contaminacion.setModel(pantalla);
196             JOptionPane.showMessageDialog(null, "El departamento que tiene mas contaminacion de "
197                 +nuevacontamiX+" es "+depMayor);
198             System.out.println("El departamento que tiene mas contaminacion de "+nuevacontamiX+" es "
199                 +depMayor+"
200                 + " con la cantidad de => "+may);
201             }else{
202                 JOptionPane.showMessageDialog(null, "Dato no valido itroduzca otro dato!!");
203             }
204         }
205     } catch (Exception e) {
206         JOptionPane.showMessageDialog(null, "No se encontraron datos");
207     }
208 }
209
210
211

```

EJECUCIÓN DEL PROGRAMA 5

[5] Mostrar los departamentos que presentaron el tipo de contaminación X

Ingrese contaminación:

DEPARTAMENTO	CANTIDAD DE CONTAMINACION
La Paz	12
Beni	10
Chuquisaca	10
Santa Cruz	4
Pando	13
Tarija	3
Cochabamba	8
Oruro	4
Potosi	13