

# UNIVERSIDAD MAYOR DE SAN ANDRES

## FACULTAD DE CIENCIAS PURAS Y NATURALES

### CARRERA DE INFORMATICA



**Materia:** Programación II

**Paralelo:** "B"

<b>Nombres:</b>	Callisaya Coca Limbert Alex	9921335
	Crespo Villamil Jorge Daniel	10060117
	Hino Flores Luis Arturo	10070467
	Quispe Mamani Brayan	12449577

**Proyecto:** Nro 7

**Tema:** Sistema de Gestión de Proyectos Deportivo Municipales

**Fecha:** 10/12/2024

**Docente:** Lic. Marcelo German Aruquipa Chambi

**La Paz – Bolivia**

# **Sistema de Gestión de Proyectos Deportivo Municipales**

Esta plataforma se centrará en la organización de proyectos deportivos a nivel municipal, desde la inscripción de equipos hasta la planificación de torneos. Se buscará facilitar el seguimiento de resultados y la asignación de recursos a diferentes disciplinas deportivas.

## **Contexto:**

Los distintos deportes de alto rendimiento es un campo complejo que exige un análisis detallado del rendimiento de los atletas para optimizar su entrenamiento y mejorar sus habilidades.

## **Objetivos:**

Sistema de Gestión de Proyectos Deportivo Municipales tiene como objetivo asegurar que el sistema no solo gestione eficientemente los proyectos deportivos, sino que también fomente la participación, promueva la transparencia y esté en constante evolución para adaptarse a las necesidades locales. Algunos objetivos clave:

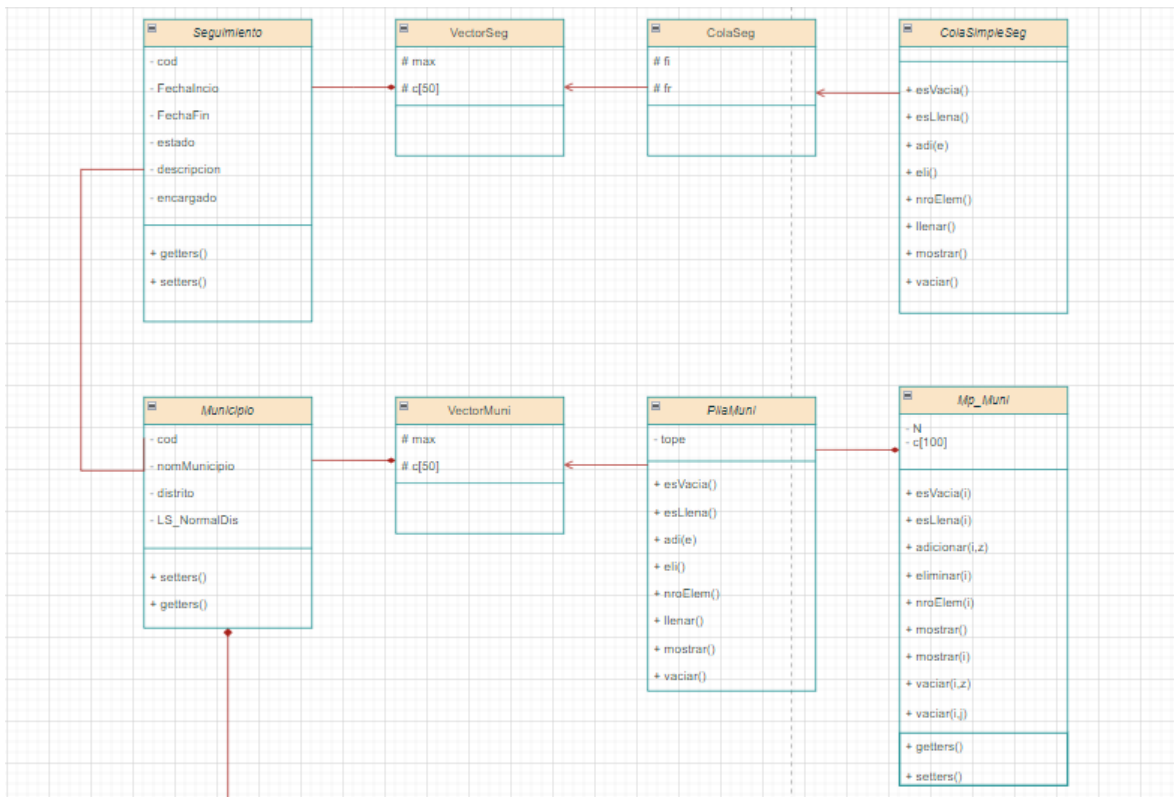
- **Centralizar la gestión de proyectos deportivos:** Proveer una plataforma única para la inscripción, seguimiento y control de todos los proyectos y eventos deportivos.
- **Automatizar la inscripción de equipos y deportistas:** Simplificar el proceso de registro, permitiendo a los equipos inscribirse de forma digital.
- **Optimizar la planificación de torneos:** Agilizar la creación y asignación de horarios, emparejamientos y sedes de los torneos deportivos.
- **Seguimiento de resultados y estadísticas:** Facilitar el registro y seguimiento en tiempo real de resultados, clasificaciones, y estadísticas de jugadores.

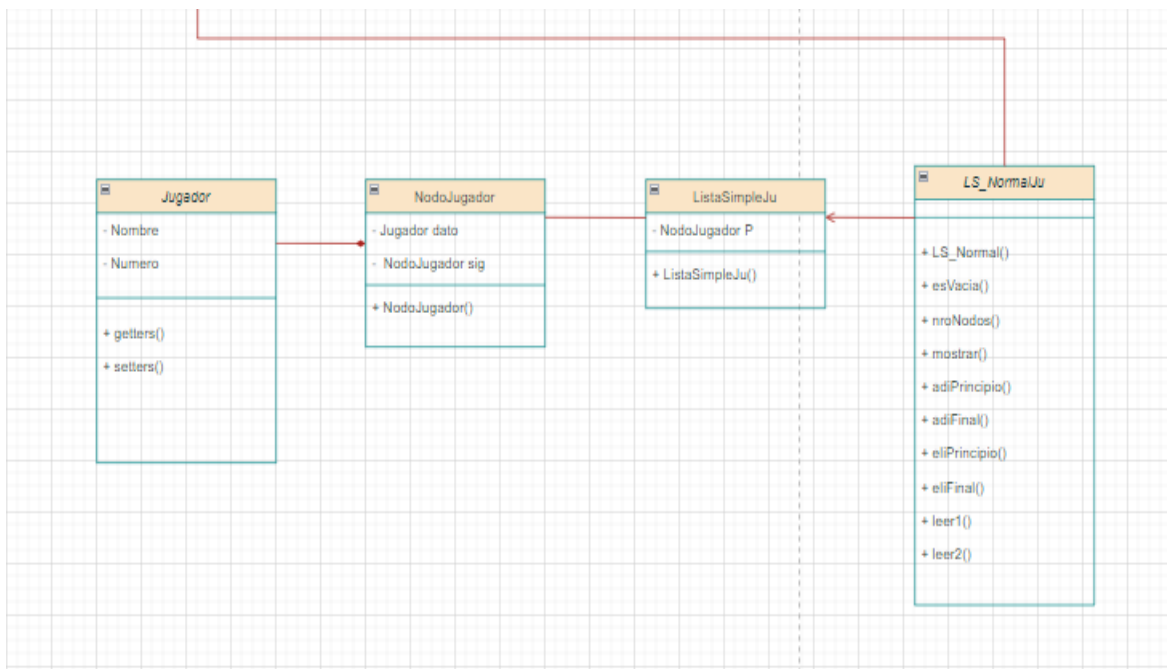
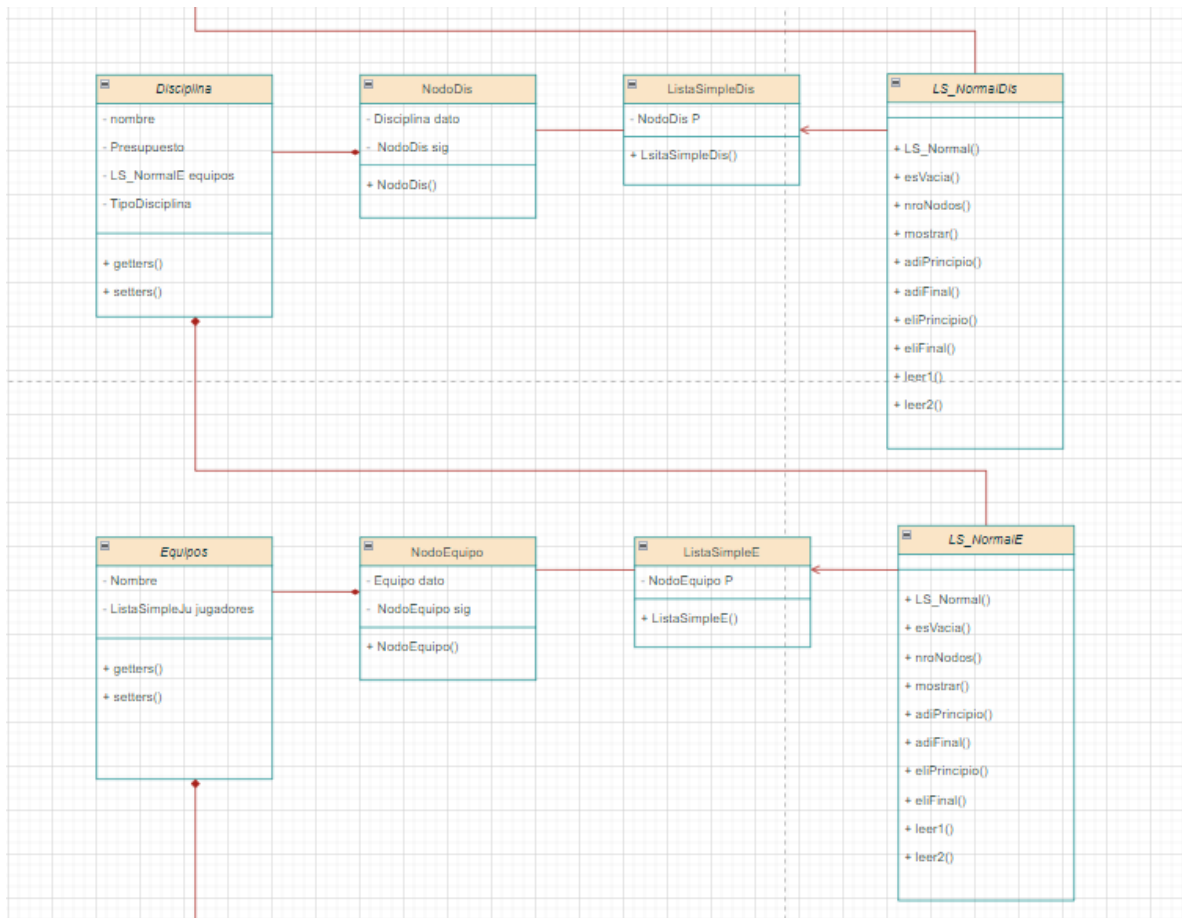
- **Asignación eficiente de recursos:** Optimizar la distribución de canchas, materiales, personal y presupuesto entre los distintos proyectos y disciplinas.
- **Gestión de instalaciones deportivas:** Coordinar el uso y mantenimiento de instalaciones deportivas municipales.

### Lista de cinco problemas a resolver:

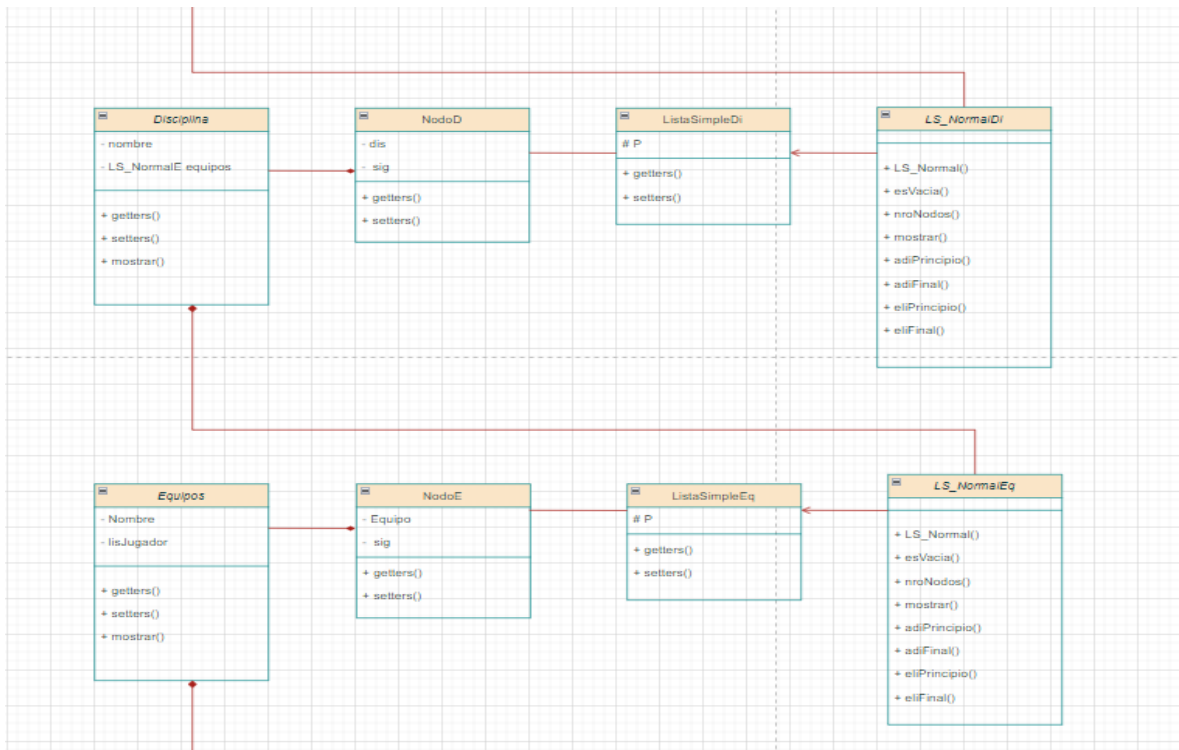
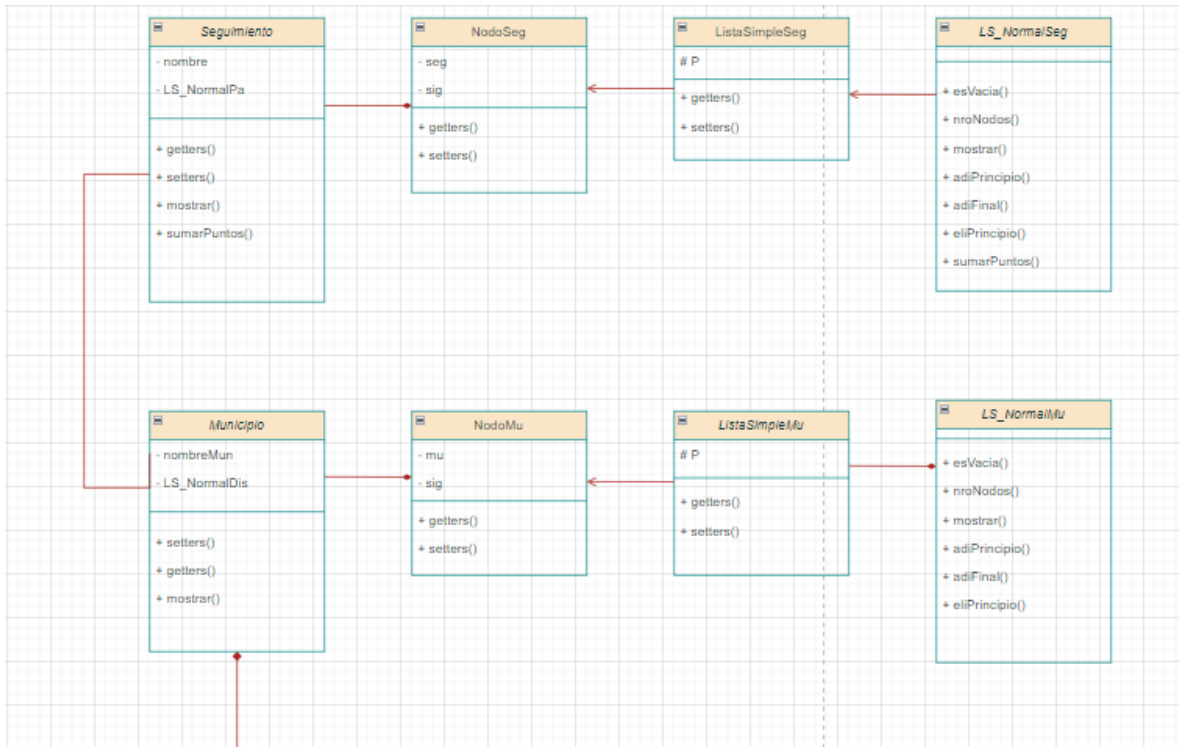
1. Crear una sola lista de equipos de una disciplina x de todos los municipios
2. Ordenar y mostrar los equipos de acuerdo a sus resultados
3. Dar como ganador al equipo que tenga mayor puntuación de cada disciplina
4. Verificar si un jugador x pertenece a dos equipos de la misma disciplina a nivel municipio.
5. Verificar que todos los equipos no tengan más de x jugadores

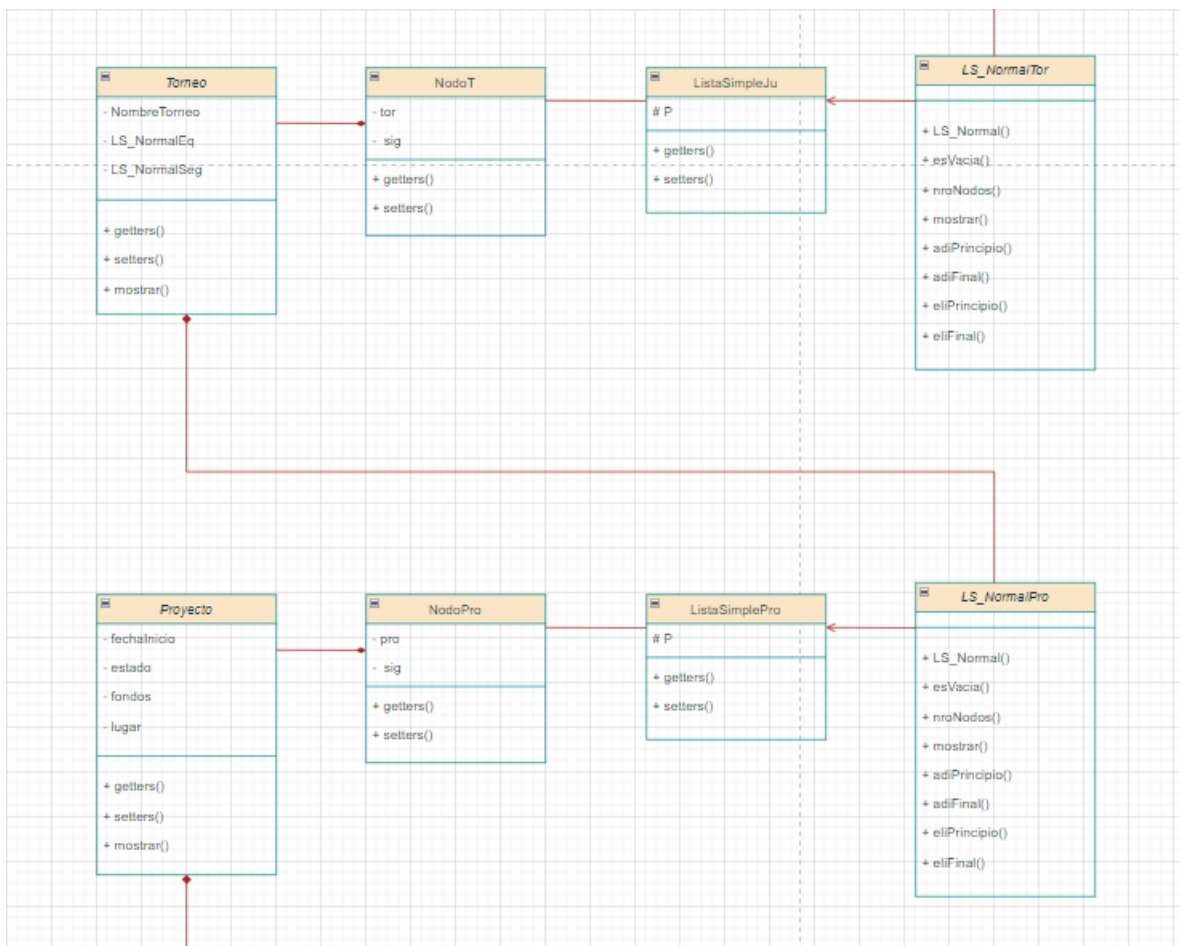
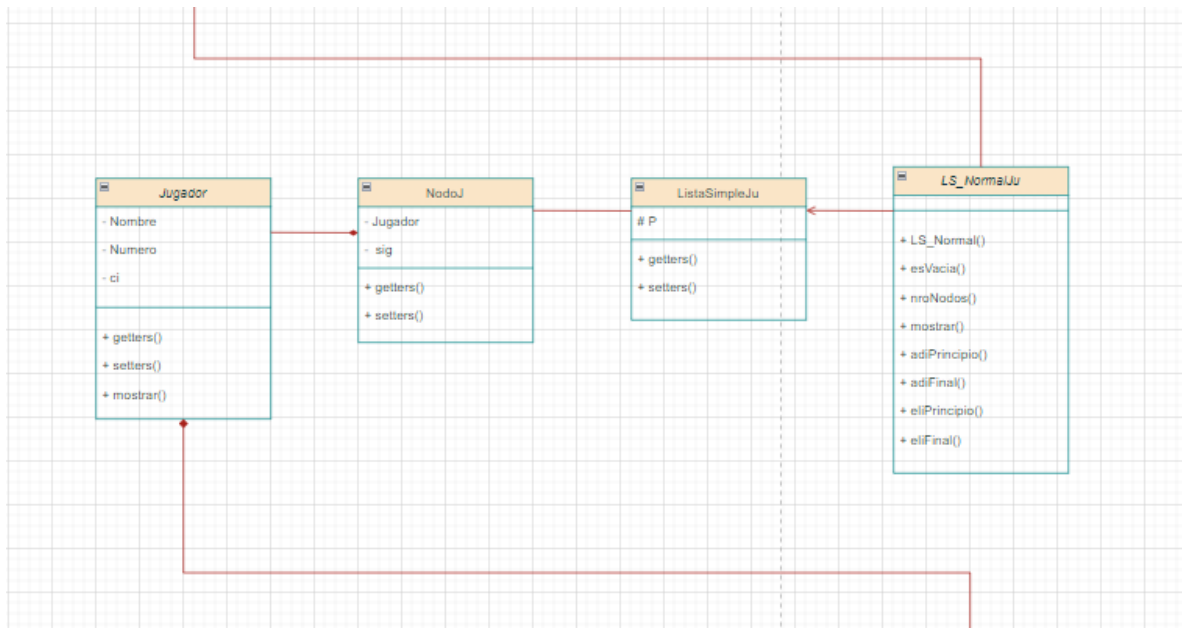
## DIAGRAMA DE CLASES

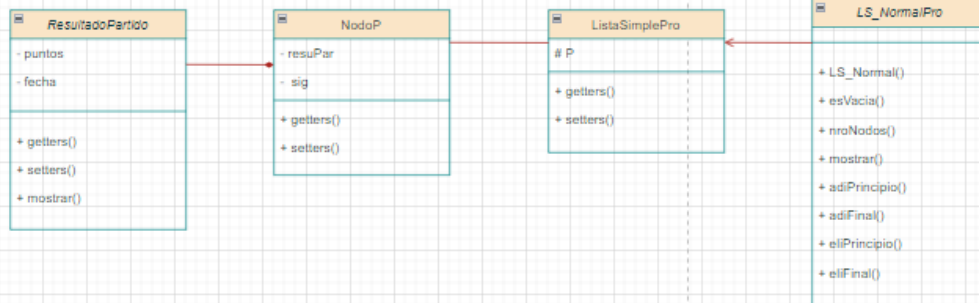




## DIAGRAMA DE CLASES FINAL







## PSEUDOCODIGO

### ➤ Problema 1

```
323 //1
324 static LS_NormalEq agruparEquipos(LS_NormalMu liMu, String disciplina) {
325     LS_NormalEq listaFinal = new LS_NormalEq(); // Lista donde se guardarán los equipos agrupados
326     NodoMu NodoMu = liMu.getP(); // Nodo inicial de la lista de municipios
327
328     while (NodoMu != null) {
329         Municipio municipio = NodoMu.getMu(); // Obtén el municipio actual
330         LS_NormalDi lisDis = municipio.getLisDis(); // Obtén las disciplinas del municipio
331         NodoD nodoDisciplina = lisDis.getP(); // Nodo inicial de las disciplinas
332
333         while (nodoDisciplina != null) {
334             Disciplina disciplinaActual = nodoDisciplina.getDis(); // Obtén la disciplina actual
335             if (disciplinaActual.getNombre().equalsIgnoreCase(disciplina)) { // Compara el nombre de la disciplina
336                 LS_NormalEq listaEquipos = disciplinaActual.getEquipos(); // Obtén los equipos de esta disciplina
337                 NodoE nodoEquipo = listaEquipos.getP(); // Nodo inicial de los equipos
338
339                 while (nodoEquipo != null) {
340                     Equipo equipo = nodoEquipo.getEq(); // Obtén el equipo actual
341                     listaFinal.adiFinal(equipo); // Añádelo a la lista final
342                     nodoEquipo = nodoEquipo.getSig(); // Avanza al siguiente equipo
343                 }
344                 nodoDisciplina = nodoDisciplina.getSig(); // Avanza a la siguiente disciplina
345             }
346             NodoMu = NodoMu.getSig(); // Avanza al siguiente municipio
347         }
348     }
349
350     return listaFinal;
351 }
```

### ➤ Problema 2

```
353 //2
354
355 static void mostrarTopDeMayorAMenor(LS_NormalTor A){
356     NodoT R = A.getP();
357     while(R != null){
358         LS_NormalSeg seg = R.getTor().getLisSeg();
359         mostrarTopOrdenado(seg);
360
361         R=R.getSig();
362     }
363 }
364
365 static void mostrarTopOrdenado(LS_NormalSeg A) {
366     // Crear una copia de la lista original
367     LS_NormalSeg listaOrdenada = copiarLista(A);
368
369     // Ordenar la lista copiada
370     ordenarListaPorPuntos(listaOrdenada);
371
372     // Mostrar los equipos en orden
373     NodoSeg R = listaOrdenada.getP();
374     while (R != null) {
375         System.out.println("Equipo: " + R.getSeg().getNombre() + " - Puntos: " + puntosEq(R));
376         R = R.getSig();
377     }
378 }
```



```

379
380 // Método para copiar una lista
381 static LS_NormalSeg copiarLista(LS_NormalSeg A) {
382     LS_NormalSeg aux = new LS_NormalSeg();
383     NodoSeg R = A.getP();
384
385     while (R != null) {
386         // Create a new node with the same data as R
387         NodoSeg S = new NodoSeg();
388         S.setSeg(R.getSeg()); // Assuming NodoSeg has a setDato method
389         aux.adifinal(S.getSeg());
390
391         R = R.getSig(); // Move to the next node
392     }
393
394     return aux;
395 }
396
397 // Método para ordenar una lista por puntos de mayor a menor (burbuja)
398 static void ordenarListaPorPuntos(LS_NormalSeg lista) {
399     if (lista.getP() == null || lista.getP().getSig() == null) {
400         return; // Lista vacía o con un solo elemento, ya está ordenada
401     }
402
403     boolean cambiado;
404     do {
405         NodoSeg R = lista.getP();
406         NodoSeg ant = null;
407         NodoSeg sig = R.getSig();
408         cambiado = false;
409
410         while (sig != null) {
411             if (puntosEq(R) < puntosEq(sig)) {
412                 // Intercambiar nodos
413                 if (ant != null) {
414                     ant.setSig(sig);
415                 } else {
416                     lista.setP(sig); // Actualizar la cabeza de la lista
417                 }
418                 R.setSig(sig.getSig());
419                 sig.setSig(R);
420
421                 cambiado = true;
422                 R = sig;
423             } else {
424                 ant = R;
425                 R = sig;
426             }
427             sig = R.getSig();
428         }
429     } while (cambiado);
430 }

```

### ➤ Problema 3

```
431 //3
432 static void mostrarCampeonesDeLosTorneosYDis (LS_NormalTor lisTor ) {
433     NodoT R = lisTor.getP();
434     while (R != null) {
435         System.out.println("CAMPEON DEL TORNEO " + R.getTor().getNombreTorneo());
436
437         mostrarTop(R.getTor().getLisSeg());
438
439         R = R.getSig();
440     }
441 }
442
443 static void mostrarTop(LS_NormalSeg A) {
444     NodoSeg R = A.getP();
445
446     while (R != null) {
447         int mayor = mayorPts(A);
448         if (puntosEq(R) == mayor) {
449             System.out.println(R.getSeg().getNombre());
450         }
451         R = R.getSig();
452     }
453 }
454
455 static int mayorPts(LS_NormalSeg A) {
456     NodoSeg R = A.getP();
457     int mayor = 0;
458     while (R != null) {
459         if (puntosEq(R) >= mayor) {
460             mayor = puntosEq(R);
461         }
462
463         R = R.getSig();
464     }
465     return mayor;
466 }
467
468 static int puntosEq(NodoSeg a) {
469     LS_NormalPa pa = a.getSeg().getLisResPa();
470     NodoP r = pa.getP();
471     int suma = 0;
472     while (r != null) {
473         suma = suma + r.getResuPar().getPuntos();
474         r = r.getSig();
475     }
476     return suma;
477 }
478
479 }
480
```

## ➤ Problema 4

```
482 //4
483 static boolean verificarJugadorEnDosEquipos(LS_NormalMu listaMunicipios, String ciJugador) {
484     NodoMu nodoMunicipio = listaMunicipios.getP();
485
486     while (nodoMunicipio != null) {
487         Municipio municipio = nodoMunicipio.getMu();
488         LS_NormalDi listaDisciplinas = municipio.getLisDis();
489         NodoD nodoDisciplina = listaDisciplinas.getP();
490
491         while (nodoDisciplina != null) {
492             Disciplina disciplina = nodoDisciplina.getDis();
493             LS_NormalEq listaEquipos = disciplina.getEquipos();
494             NodoE nodoEquipo = listaEquipos.getP();
495
496             int contadorApariciones = 0;
497
498             while (nodoEquipo != null) {
499                 Equipo equipo = nodoEquipo.getEq();
500                 LS_NormalJu listaJugadores = equipo.getLisJugador();
501                 NodoJ nodoJugador = listaJugadores.getP();
502
503                 while (nodoJugador != null) {
504                     Jugador jugador = nodoJugador.getJu();
505                     if (jugador.getCi().equals(ciJugador)) {
506                         contadorApariciones++;
507                         // Si aparece en más de un equipo, devolvemos true
508                         if (contadorApariciones > 1) {
509                             return true;
510                         }
511                     }
512                     nodoJugador = nodoJugador.getSig();
513                 }
514                 nodoEquipo = nodoEquipo.getSig();
515             }
516             nodoDisciplina = nodoDisciplina.getSig();
517         }
518         nodoMunicipio = nodoMunicipio.getSig();
519     }
520
521     return false;
522 }
```

## ➤ Problema 5

```
524 //5
525 public static void verificarEquiposNoExcedenJugadores(LS_NormalMu liMu, int maxJugadores) {
526     NodoMu nodoMunicipio = liMu.getP(); // Primer municipio
527     boolean todosValidos = true; // Bandera para verificar si todos los equipos cumplen con el limite
528
529     while (nodoMunicipio != null) {
530         System.out.println("Municipio: " + nodoMunicipio.getMu().getNombreMun());
531         LS_NormalDi listaDisciplinas = nodoMunicipio.getMu().getLisDis();
532         NodoD nodoDisciplina = listaDisciplinas.getP(); // Primer disciplina
533
534         while (nodoDisciplina != null) {
535             System.out.println("    Disciplina: " + nodoDisciplina.getDis().getNombre());
536             LS_NormalEq listaEquipos = nodoDisciplina.getDis().getEquipos();
537             NodoE nodoEquipo = listaEquipos.getP(); // Primer equipo
538
539             while (nodoEquipo != null) {
540                 LS_NormalJu listaJugadores = nodoEquipo.getEq().getLisJugador();
541                 int contadorJugadores = contarJugadores(listaJugadores);
542
543                 if (contadorJugadores > maxJugadores) {
544                     System.out.println("        El equipo " + nodoEquipo.getEq().getNombre() + " tiene "
545                                         + contadorJugadores + " jugadores. Excede el limite de " + maxJugadores + "!");
546                     todosValidos = false; // Marca que no todos los equipos cumplen
547                 } else {
548                     System.out.println("        El equipo " + nodoEquipo.getEq().getNombre() + " tiene "
549                                         + contadorJugadores + " jugadores. Cumple con el limite.");
550                 }
551                 nodoEquipo = nodoEquipo.getSig(); // Siguiente equipo
552             }
553             nodoDisciplina = nodoDisciplina.getSig(); // Siguiente disciplina
554         }
555         nodoMunicipio = nodoMunicipio.getSig(); // Siguiente municipio
556     }
557
558     if (todosValidos) {
559         System.out.println("\nTodos los equipos cumplen con el limite de " + maxJugadores + " jugadores.");
560     } else {
561         System.out.println("\nAlgunos equipos exceden el limite de " + maxJugadores + " jugadores.");
562     }
563 }
564
565 // Método auxiliar para contar jugadores en un equipo
566 public static int contarJugadores(LS_NormalJu listaJugadores) {
567     NodoJ nodoJugador = listaJugadores.getP(); // Primer jugador
568     int contador = 0;
569
570     while (nodoJugador != null) {
571         contador++;
572         nodoJugador = nodoJugador.getSig(); // Siguiente jugador
573     }
574
575     return contador;
576 }
577
578 }
579
```

## **EJECUCIÓN DEL PROGRAMA**

```
Output - InterfazProyecto (run) ×
run:
EJERCICIO 1
Nombre Del Torneo: Torneo de Informatica
-----
Nombre del equipo: Real Sociedad
Nombre del jugador: brayan
Numero del jugador: 7
Ci del jugador: 123
Nombre del jugador: alex
Numero del jugador: 10
Ci del jugador: 122
Nombre del jugador: pepe
Numero del jugador: 3
Ci del jugador: 133
-----
Nombre del equipo: Vengadores
Nombre del jugador: brandon
Numero del jugador: 7
Ci del jugador: 333
Nombre del jugador: daniel
Numero del jugador: 11
Ci del jugador: 444
Nombre del jugador: luis
Numero del jugador: 5
Ci del jugador: 555
-----
Nombre del equipo: Real Sociedad
Nombre del jugador: brayan
Numero del jugador: 7
Ci del jugador: 123
Nombre del jugador: alex
Numero del jugador: 10
Ci del jugador: 122
Nombre del jugador: pepe
Numero del jugador: 3
Ci del jugador: 133
-----
```

```
-----
Nombre del equipo: Vengadores
Nombre del jugador: brandon
Numero del jugador: 7
Ci del jugador: 333
Nombre del jugador: daniel
Numero del jugador: 11
Ci del jugador: 444
Nombre del jugador: luis
Numero del jugador: 5
Ci del jugador: 555
-----

Nombre del equipo: Real Sociedad
Nombre del jugador: brayan
Numero del jugador: 7
Ci del jugador: 123
Nombre del jugador: alex
Numero del jugador: 10
Ci del jugador: 122
Nombre del jugador: pepe
Numero del jugador: 3
Ci del jugador: 133
-----

Nombre del equipo: Vengadores
Nombre del jugador: brandon
Numero del jugador: 7
Ci del jugador: 333
Nombre del jugador: daniel
Numero del jugador: 11
Ci del jugador: 444
Nombre del jugador: luis
Numero del jugador: 5
Ci del jugador: 555
Nombre del equipo: Real Sociendad
-----

Resultados Partidos:
Puntos del partido... 3
Fecha del partido... 1 de enero
-----
```

```
-----
Resultados Partidos:
Puntos del partido... 3
Fecha del partido... 2 de enero
-----

Resultados Partidos:
Puntos del partido... 3
Fecha del partido... 3 de enero
Nombre del equipo: Real Socienda
-----

Resultados Partidos:
Puntos del partido... 0
Fecha del partido... 1 de enero
-----

Resultados Partidos:
Puntos del partido... 0
Fecha del partido... 2 de enero
-----

Resultados Partidos:
Puntos del partido... 0
Fecha del partido... 3 de enero
Nombre del equipo: Vengadores
-----

Resultados Partidos:
Puntos del partido... 0
Fecha del partido... 1 de enero
-----

Resultados Partidos:
Puntos del partido... 0
Fecha del partido... 2 de enero
-----

Resultados Partidos:
Puntos del partido... 0
Fecha del partido... 3 de enero
Nombre del equipo: Real Socienda
-----

Resultados Partidos:
Puntos del partido... 0
Fecha del partido... 1 de enero
-----
```

-----  
Resultados Partidos:  
Puntos del partido... 0  
Fecha del partido... 2 de enero  
-----

Resultados Partidos:  
Puntos del partido... 0  
Fecha del partido... 3 de enero  
Nombre del equipo: Vengadores  
-----

Resultados Partidos:  
Puntos del partido... 0  
Fecha del partido... 1 de enero  
-----

Resultados Partidos:  
Puntos del partido... 0  
Fecha del partido... 2 de enero  
-----

Resultados Partidos:  
Puntos del partido... 0  
Fecha del partido... 3 de enero

#### EJERCICIO 2

Equipo: Vengadores - Puntos: 9  
Equipo: Real Sociendad - Puntos: 4  
Equipo: Real Sociendad - Puntos: 0  
Equipo: Vengadores - Puntos: 0  
Equipo: Real Sociendad - Puntos: 0  
Equipo: Vengadores - Puntos: 0

#### EJERCICIO 3

CAMPEON DEL TORNEO Torneo de Informatica  
Vengadores

#### Ejercici 4

El jugador con CI 123 NO pertenece a dos equipos de la misma disciplina.



## EJERCICIO 5

Verificando que los equipos no tengan mss de 3 jugadores...

Municipio: El Alto

Disciplina: FUTBOL

El equipo Real Sociedad tiene 4 jugadores. Excede el limite de 3!

El equipo Vengadores tiene 3 jugadores. Cumple con el limite.

Disciplina: WALLY

El equipo Vengadores tiene 3 jugadores. Cumple con el limite.

El equipo Alpha tiene 3 jugadores. Cumple con el limite.

Municipio: La Paz

Disciplina: FUTBOL

El equipo Real Sociedad tiene 4 jugadores. Excede el limite de 3!

El equipo Vengadores tiene 3 jugadores. Cumple con el limite.

Disciplina: WALLY

El equipo Vengadores tiene 3 jugadores. Cumple con el limite.

El equipo Alpha tiene 3 jugadores. Cumple con el limite.

Municipio: Viacha

Disciplina: FUTBOL

El equipo Real Sociedad tiene 4 jugadores. Excede el limite de 3!

El equipo Vengadores tiene 3 jugadores. Cumple con el limite.

Disciplina: WALLY

El equipo Vengadores tiene 3 jugadores. Cumple con el limite.

El equipo Alpha tiene 3 jugadores. Cumple con el limite.

Algunos equipos exceden el limite de 3 jugadores.

BUILD SUCCESSFUL (total time: 0 seconds)