

Primera pràctica de AS

Grupo 13

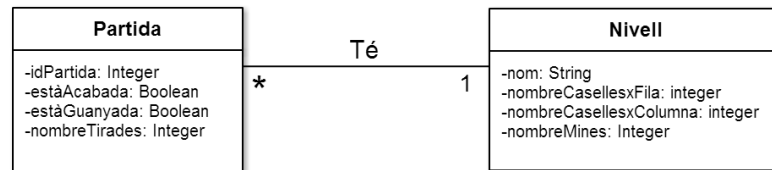
Qiaorui Xiang
ZhouYang Xue
Ignacio N. Lucero Ascencio
Liya Ye

Índex

1. Codi java amb hibernate.....	1
1.1 Esquema de les bases de dades.....	1
1.2 Codi Java.....	1
1.3 Programa de prova.....	2
2. Diagrames de seqüència de cas d'us de Jugar Partida.....	11
3. Diagrama de classes de la capa de domini.....	18
4. Justificació dels patrons de disseny.....	19

1. Codi java amb hibernate

1.1 Esquema de les bases de dades:



1.2 Codi Java:

Partida.class :

```
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;

@Entity
public class Partida {
    private int nombreTirades;
    private int idPartida;
    private boolean estaAcabada;
    private boolean estaGuanyada;
    private Nivel niv;

    @OneToOne()
    @JoinColumn(name = "NivelNom", nullable = false, updatable=false)
    public Nivel getNiv() {
        return niv;
    }

    public void setNiv(Nivel niv) {
        this.niv = niv;
    }

    @Id
    public int getIdPartida() {
        return idPartida;
    }

    public void setIdPartida(int idPartida) {
        this.idPartida = idPartida;
    }

    public int getNombreTirades() {
        return nombreTirades;
    }

    public void setNombreTirades(int nombreTirades) {
        this.nombreTirades = nombreTirades;
    }

    public boolean getEstaAcabada() {
        return estaAcabada;
    }

    public void setEstaAcabada(boolean estaAcabada) {
        this.estaAcabada = estaAcabada;
    }

    public boolean getEstaGuanyada() {
```

```

        return estaGuanyada;
    }
    public void setEstaGuanyada(boolean estaGuanyada) {
        this.estaGuanyada = estaGuanyada;
    }
}

```

Nivel.class

```

import javax.persistence.Entity;
import javax.persistence.Id;

```

```

@Entity
public class Nivel {
    private String nom;
    private int nombreCasellesxFila;
    private int nombreCasellesxColumna;
    private int nombreMines;

    @Id
    public String getNom() {
        return nom;
    }
    public void setNom(String nom) {
        this.nom = nom;
    }
    public int getNombreCasellesxFila() {
        return nombreCasellesxFila;
    }
    public void setNombreCasellesxFila(int nombreCasellesxFila) {
        this.nombreCasellesxFila = nombreCasellesxFila;
    }
    public int getNombreCasellesxColumna() {
        return nombreCasellesxColumna;
    }
    public void setNombreCasellesxColumna(int nombreCasellesxColumna) {
        this.nombreCasellesxColumna = nombreCasellesxColumna;
    }
    public int getNombreMines() {
        return nombreMines;
    }
    public void setNombreMines(int nombreMines) {
        this.nombreMines = nombreMines;
    }
}

```

1.3 Programa de prova:

```

import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.tool.hbm2ddl.SchemaExport;

public class PartidaNivelDriver {
    private static AnnotationConfiguration config;
    private static SessionFactory factory;

    protected static String title;
    protected static List<String> menu = new ArrayList<String>();
}

```

```

protected static String _msg_error_pre = "Error: ";

public static void main(String[] args) {
    // TODO Auto-generated method stub
    config = new AnnotationConfiguration();
    config.addAnnotatedClass(Partida.class);
    config.addAnnotatedClass(Nivel.class);
    config.configure("hibernate.cfg.xml");
    factory = config.buildSessionFactory();

    new SchemaExport(config).create(true, true);

    Nivel n = null;
    Partida p = null;
    //Generico del menu
    Scanner in = new Scanner(System.in);
    int opc = 0;
    int res = 0;
    String s1, s2;
    String[] argv;

    //Menu
    _menu();

    //Opciones
    do {
        //Lectura de datos
        argv = null;

        if (in.hasNext()) {
            argv = in.nextLine().split(" ");
        }

        if (argv == null) { //Terminamos el fichero
            opc = 0;
        } else if (argv.length > 0) {
            //Recoger la opcion del usuario
            opc = Integer.parseInt(argv[0]);
            //Accion
            switch(opc) {
                case 0:
                    break;
                case 1:
                    if (argv.length < 5) _msg_error_param_insuf();
                    else {
                        creatNivel(argv[1], Integer.parseInt(argv[2]),
                                Integer.parseInt(argv[3]), Integer.parseInt(argv[4]));
                        System.out.println("Nivel creado. \n");
                    }
                    break;
                case 2:
                    if (argv.length < 2) _msg_error_param_insuf();
                    else {
                        deleteNivel(argv[1]);
                        System.out.println("Nivel eliminado. \n");
                    }
                    break;
                case 3:
                    if (argv.length < 2) _msg_error_param_insuf();
                    else {
                        n = getNivel(argv[1]);
                        System.out.println("El nivel "+ n.getNom()+" tiene"+
                                n.getNombreCasellesxFila()+" casillas por fila,
                                "+n.getNombreCasellesxColumna()+" "+ "casillas por
                                columna y "+n.getNombreMines()+" minas. \n" );
                    }
            }
        }
    }
}

```

```

        break;
case 4:
    if (argv.length < 2) _msg_error_param_insuf();
    else {
        res = getnombreCasellesxFila(argv[1]);
        System.out.println("El nivel tiene "+res+" casillas por
        fila. \n");
    }
    break;
case 5:
    if (argv.length < 2) _msg_error_param_insuf();
    else {
        res = getnombreCasellesxColumna(argv[1]);
        System.out.println("El nivel tiene "+res+" casillas por
        columna. \n");
    }
    break;
case 6:
    if (argv.length < 2) _msg_error_param_insuf();
    else {
        res = getNombreMines(argv[1]);
        System.out.println("El nivel tiene "+res+" minas. \n");
    }
    break;
case 7:
    if (argv.length < 3) _msg_error_param_insuf();
    else {
        setnombreCasellesxFila(argv[1],
        Integer.parseInt(argv[2]));
        System.out.println("Casillas por fila puestas. \n");
    }
    break;
case 8:
    if (argv.length < 3) _msg_error_param_insuf();
    else {
        setnombreCasellesxColumna(argv[1],
        Integer.parseInt(argv[2]));

        System.out.println("Casillas por columna puestas. \n");
    }
    break;
case 9:
    if (argv.length < 3) _msg_error_param_insuf();
    else {
        setNombreMines(argv[1],Integer.parseInt(argv[2]));
        System.out.println("Numero de minas puesto. \n");
    }
    break;
case 10:
    if (argv.length < 5) _msg_error_param_insuf();
    else {
        creatPartida(Integer.parseInt(argv[1]),
        Integer.parseInt(argv[2]),
        Boolean.parseBoolean(argv[3]),
        Boolean.parseBoolean(argv[4]), getNivel(argv[5]));
        System.out.println("Partida creada. \n");
    }
    break;
case 11:
    if (argv.length < 2) _msg_error_param_insuf();
    else {
        deletePartida(Integer.parseInt(argv[1]));
        System.out.println("Partida eliminada. \n");
    }
    break;
case 12:

```

```

        if (argv.length < 2) _msg_error_param_insuf();
        else {
            p = getPartida(Integer.parseInt(argv[1]));
            if(p.getEstaAcabada()) s1 = " "; else s1 = "no";
            if(p.getEstaGuanyada()) s2 = " "; else s2 = "no";

            System.out.println("La partida "+p.getIdPartida()+"
            tiene "+p.getNombreTirades()+" tiradas, "
            + ""+s1+" esta acabada y "+s2+" esta ganada. \n");
        }
        break;
case 13:
        if (argv.length < 2) _msg_error_param_insuf();
        else {
            res = getNombreTirades(Integer.parseInt(argv[1]));
            System.out.println("La partida tiene "+res+" tiradas.
            \n");
        }
        break;
case 14:
        if (argv.length < 2) _msg_error_param_insuf();
        else {
            boolean b = getEstaAcabada(Integer.parseInt(argv[1]));
            if(b) s1 = " "; else s1 = "no";
            System.out.println("La partida "+s1+" esta acabada.
            \n");
        }
        break;
case 15:
        if (argv.length < 2) _msg_error_param_insuf();
        else {
            boolean b= getEstaGuanyada(Integer.parseInt(argv[1]));
            if(b) s2 = " "; else s2 = "no";
            System.out.println("La partida "+s2+" esta ganada. \n");
        }
        break;
case 16:
        if (argv.length < 3) _msg_error_param_insuf();
        else {
            setNombreTirades(Integer.parseInt(argv[1]),
            Integer.parseInt(argv[2]));
            System.out.println("Numero de tiradas puesto. \n");
        }
        break;
case 17:
        if (argv.length < 3) _msg_error_param_insuf();
        else {
            setEstaAcabada(Integer.parseInt(argv[1]),
            Boolean.parseBoolean(argv[2]));
            System.out.println("Estado de finalizacion cambiado.
            \n");
        }
        break;
case 18:
        if (argv.length < 3) _msg_error_param_insuf();
        else {
            setEstaGuanyada(Integer.parseInt(argv[1]),
            Boolean.parseBoolean(argv[2]));
            System.out.println("Estado de victoria cambiado. \n");
        }
        break;
case 19:
        if (argv.length < 2) _msg_error_param_insuf();
        else {
            n = getNivel(Integer.parseInt(argv[1]));

```



```

        System.out.println("El nivel "+ n.getNom()+" tiene
        "+n.getNombreCasellesxFila()+" casillas por fila,
        "+n.getNombreCasellesxColumna()+" "

+ "casillas por columna y "+n.getNombreMines()+" minas. \n" );

        }
        break;
    case 20:
        if (argv.length < 3) _msg_error_param_insuf();
        else {
            setNivel(Integer.parseInt(argv[1]), getNivel(argv[2]));
            System.out.println("Nivel de partida cambiado. \n");
        }
        break;
    default:
        System.out.println("Opcion Invalido");
        break;
    }
}

} while (opc != 0);
factory.close();
}

//Menu
//-----
public static void print_menu(){

    //Print title
    System.out.println("-----");
    System.out.println(title);
    System.out.println("----- \n\n");

    //MENU
    System.out.println("0. Exit");

    for(int i = 0; i < menu.size(); i++) {
        System.out.println((i + 1) + ". " + menu.get(i));
    }

    System.out.println("----- \n\n");
}

protected static void _msg_error_param_insuf() {
    System.out.println("Faltan parametros");
}

private static void _menu() {

    title = "Driver de Nivel y Partida";

    menu.add("createNivel(String nom, int nombreCasellesxFila, int
nombreCasellesxColumna, int nombreMines) : Nivel"); //1
    menu.add("deleteNivel(String nom) : void"); //2
    menu.add("getNivel(String nom) : Nivel"); //3
    menu.add("getnombreCasellesxFila(String nom) : int"); //4
    menu.add("getnombreCasellesxColumna(String nom) : int"); //5
    menu.add("getNombreMines(String nom) : int"); //6
    menu.add("setnombreCasellesxFila(String nom, int nombreCasellesxFila) : void"); //7
    menu.add("setnombreCasellesxColumna(String nom, int nombreCasellesxColumna) :
void"); //8
    menu.add("setNombreMines(String nom, int nombreMines) : void"); //9

    menu.add("createPartida(int id, int nombreTirades, boolean estaAcabada,

```

```

        boolean estaGuanyada, Nivel n(el nom) : void");//10
    menu.add("deletePartida(int id) : void");//11
    menu.add("getPartida(int id) : Partida");//12
    menu.add("getNombreTirades(int id) : int");//13
    menu.add("getEstaAcabada(int id) : boolean");//14
    menu.add("getEstaGuanyada(int id) : boolean");//15
    menu.add("getNivel(int id) : Nivel");//19
    menu.add("setNombreTirades(int id, int nombreTirades) : void");//16
    menu.add("setEstaAcabada(int id, boolean estaAcabada) : void");//17
    menu.add("setEstaGuanyada(int id, boolean estaGuanyada) : void");//18
    menu.add("setNivel(int id, Nivel n(el nom)) : void");//20
    print_menu();
}

public static void creatTable() {
    new SchemaExport(config).create(true, true);
}

// Tots els operacion sobre nivel

public static void creatNivel(String nom, int nombreCasellesxFila, int nombreCasellesxColumna,
    int nombreMines) {

    Session session = factory.getCurrentSession();
    session.beginTransaction();

    Nivel n = new Nivel();

    n.setNom(nom);
    n.setNombreCasellesxColumna(nombreCasellesxColumna);
    n.setNombreCasellesxFila(nombreCasellesxFila);
    n.setNombreMines(nombreMines);

    session.save(n);
    session.getTransaction().commit();
}

public static void deleteNivel(String nom) {
    Session session = factory.getCurrentSession();
    session.beginTransaction();

    Query q = session.createQuery("delete from Nivel where nom = '"+nom+"'");
    q.executeUpdate();
    session.getTransaction().commit();
}

public static Nivel getNivel(String nom) {
    Session session = factory.getCurrentSession();
    session.beginTransaction();

    Nivel n = (Nivel) session.createQuery("from Nivel where nom = '"+nom+"'").uniqueResult();
    session.getTransaction().commit();
    return n;
}

```

```

    public static int getnombreCasellesxFila(String nom) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();

        int n = (int) session.createQuery("Select nombreCasellesxFila from Nivel
                                         where nom = '"+nom+"'").uniqueResult();
        session.getTransaction().commit();
        return n;
    }
    public static int getnombreCasellesxColumna(String nom) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();

        int n = (int) session.createQuery("Select nombreCasellesxColumna from Nivel
                                         where nom = '"+nom+"'").uniqueResult();
        session.getTransaction().commit();
        return n;
    }
    public static int getNombreMines(String nom) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();

        int n = (int) session.createQuery("Select nombreMines from Nivel
                                         where nom = '"+nom+"'").uniqueResult();
        session.getTransaction().commit();
        return n;
    }

    public static void setnombreCasellesxFila(String nom, int nombreCasellesxFila) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        Query q = session.createQuery("Update Nivel set nombreCasellesxFila = '"+nombreCasellesxFila+"
                                     where nom = '"+nom+"'");

        q.executeUpdate();
        session.getTransaction().commit();
    }
    public static void setnombreCasellesxColumna(String nom, int nombreCasellesxColumna) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        Query q = session.createQuery("Update Nivel
                                     set nombreCasellesxColumna = '"+nombreCasellesxColumna+"
                                     where nom = '"+nom+"'");

        q.executeUpdate();
        session.getTransaction().commit();
    }
    public static void setNombreMines(String nom, int nombreMines) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        Query q = session.createQuery("Update Nivel set nombreMines = '"+nombreMines+"
                                     where nom = '"+nom+"'");

        q.executeUpdate();
        session.getTransaction().commit();
    }

    // Tots els operacion sobre partida
    public static void creatPartida(int id, int nombreTirades, boolean estaAcabada, boolean estaGuanyada, Nivel n)
{

```

```

        Session session = factory.getCurrentSession();
        session.beginTransaction();

        Partida a = new Partida();

        a.setIdPartida(id);
        a.setNombreTirades(nombreTirades);
        a.setEstaAcabada(estaAcabada);
        a.setEstaGuanyada(estaGuanyada);
        a.setNiv(n);
        session.save(a);
        session.getTransaction().commit();
    }

    public static void deletePartida(int id) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();

        Query q = session.createQuery("delete Partida where idPartida = "+id+"");
        q.executeUpdate();
        session.getTransaction().commit();
    }

    public static Partida getPartida(int id) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        Partida p = (Partida) session.createQuery("from Partida where id = "+id+"").uniqueResult();
        session.getTransaction().commit();
        return p;
    }

    public static int getNombreTirades(int id) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        int p = (int) session.createQuery("Select nombreTirades from Partida
                                         where id = "+id+"").uniqueResult();
        session.getTransaction().commit();
        return p;
    }

    public static boolean getEstaAcabada(int id) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        boolean p = (boolean) session.createQuery("Select estaAcabada from Partida
                                                where id = "+id+"").uniqueResult();
        session.getTransaction().commit();
        return p;
    }

    public static boolean getEstaGuanyada(int id) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        boolean p = (boolean) session.createQuery("Select estaGuanyada from Partida
                                                where id = "+id+"").uniqueResult();
        session.getTransaction().commit();
        return p;
    }

    public static Nivel getNivel(int id) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        String aux = (String) session.createQuery("Select nivelnom from Partida
                                                where id = "+id+"").uniqueResult();
    }

```

```

        session.getTransaction().commit();
        return getNivel(aux);
    }

    public static void setNombreTirades(int id, int nombreTirades) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        Query q = session.createQuery("Update Partida set nombreTirades = "+nombreTirades+"
                                     where id = "+id+"");
        q.executeUpdate();
        session.getTransaction().commit();
    }

    public static void setEstaAcabada(int id, boolean estaAcabada) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        Query q = session.createQuery("Update Partida set estaAcabada = "+estaAcabada+"
                                     where id = "+id+"");
        q.executeUpdate();
        session.getTransaction().commit();
    }

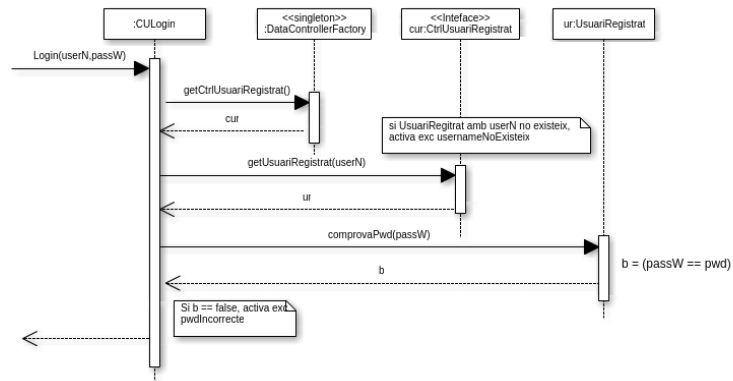
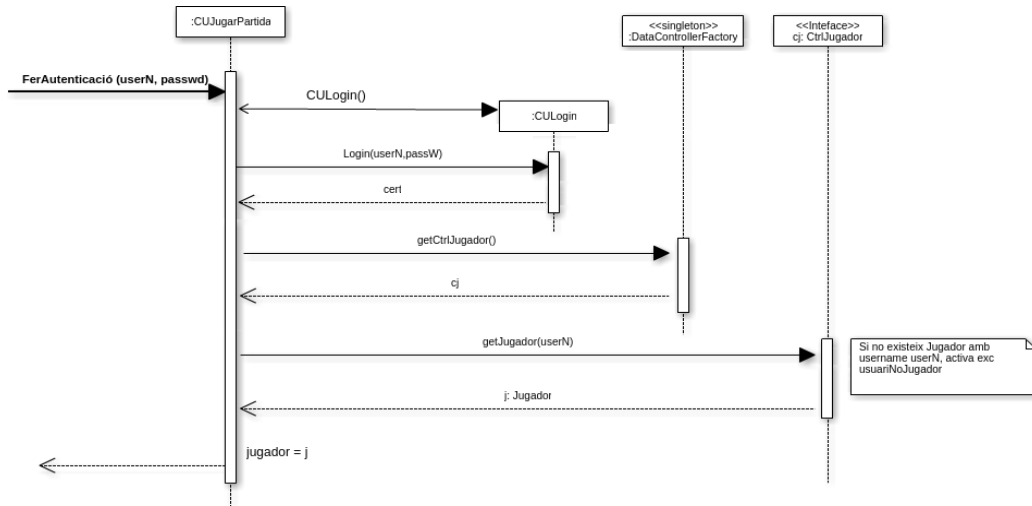
    public static void setEstaGuanyada(int id, boolean estaGuanyada) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        Query q = session.createQuery("Update Partida set estaGuanyada = "+estaGuanyada+"
                                     where id = "+id+"");
        q.executeUpdate();
        session.getTransaction().commit();
    }

    public static void setNivel(int id, Nivel n) {
        Session session = factory.getCurrentSession();
        session.beginTransaction();
        Query q = session.createQuery("Update Partida set nivelnom = "+n.getNom()+" where id = "+id+"");
        q.executeUpdate();
        session.getTransaction().commit();
    }
}

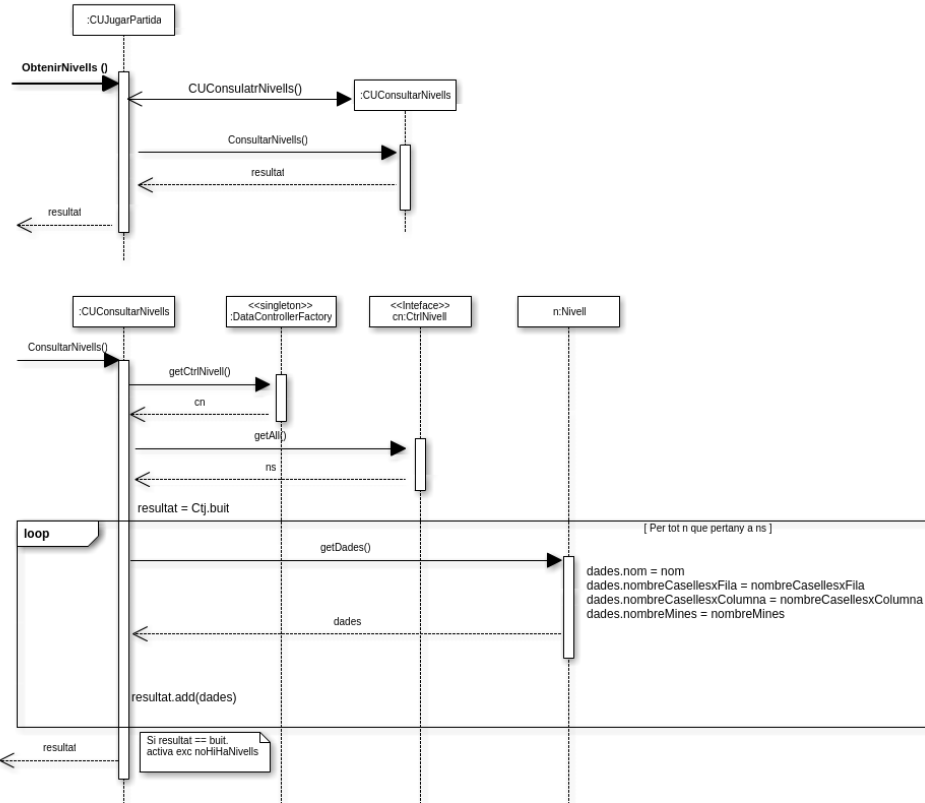
```

2. Diagrames de seqüència de cas d'us de Jugar Partida

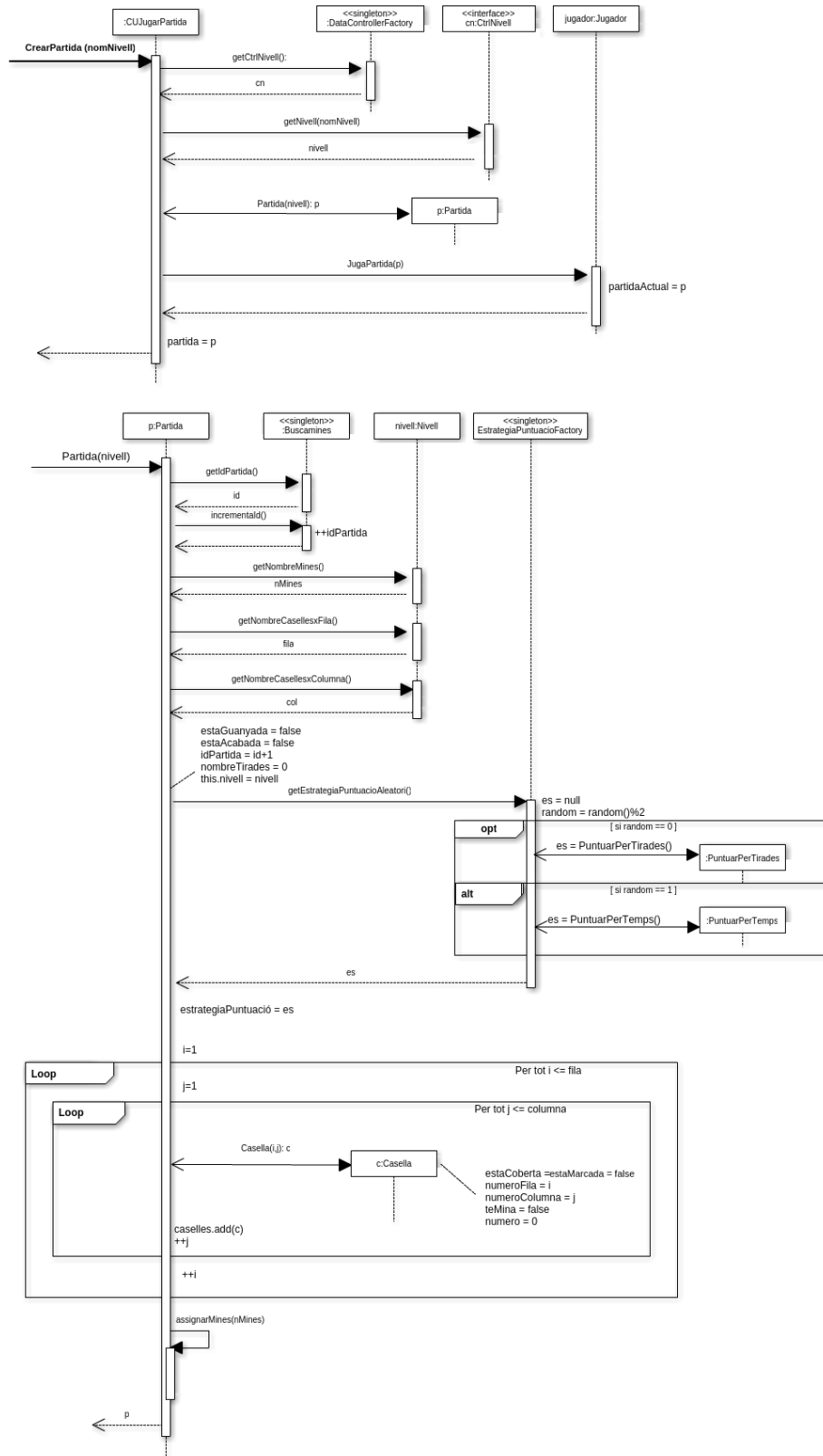
FerAutenticació

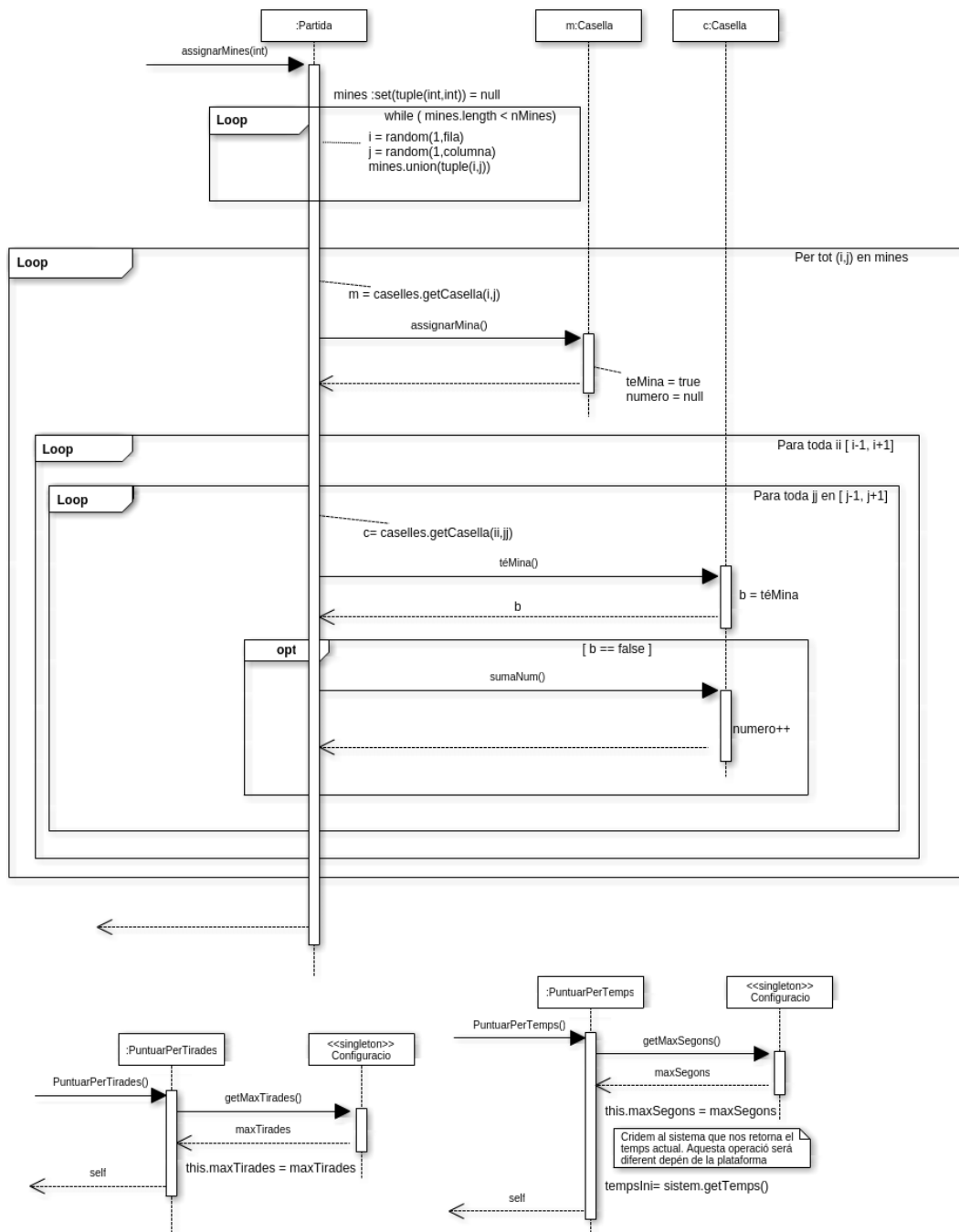


ObtenirNivells

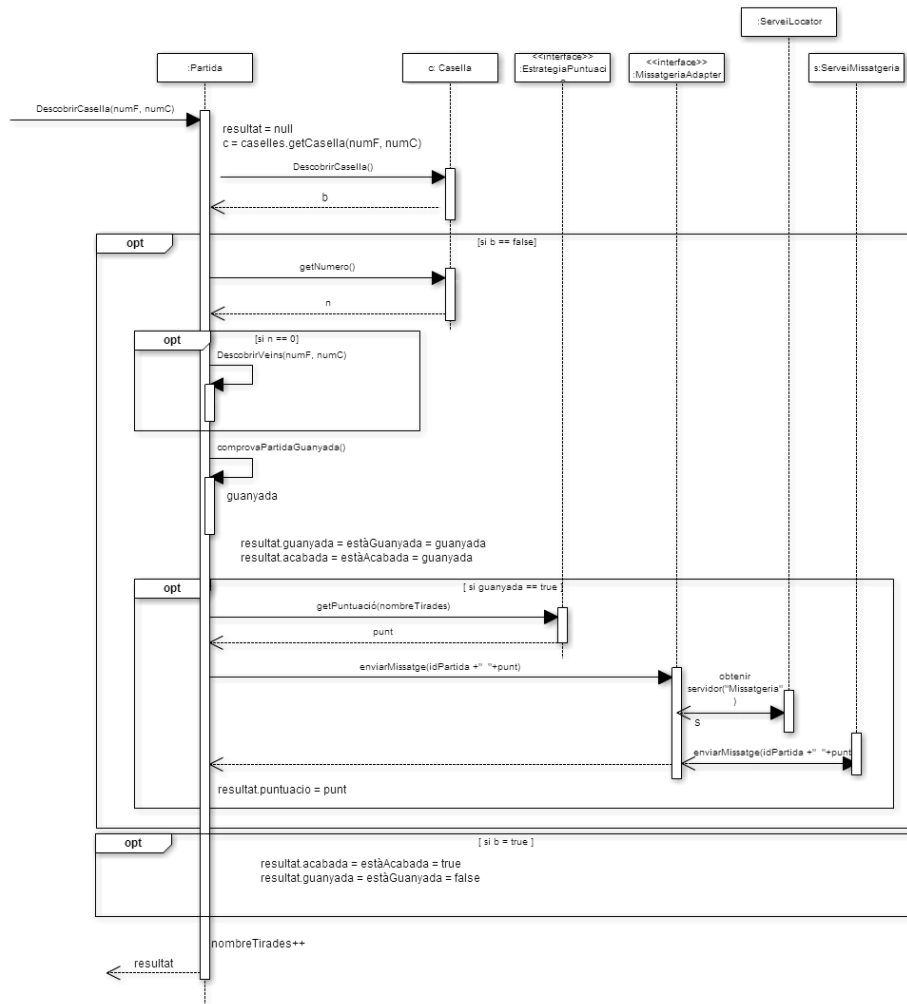
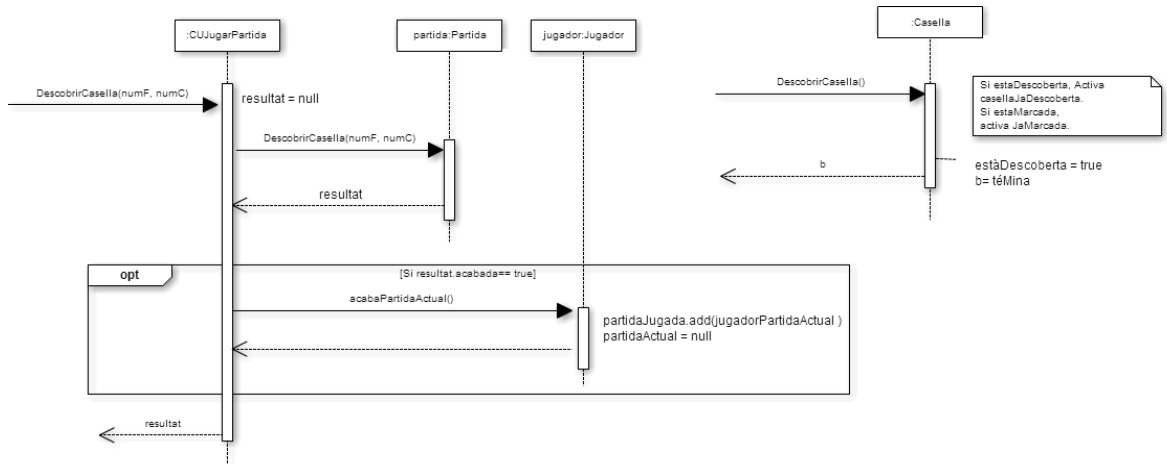


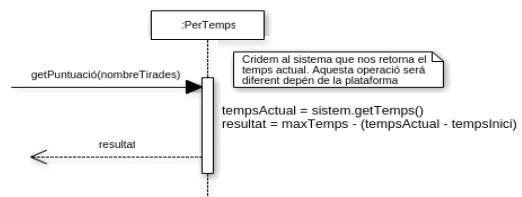
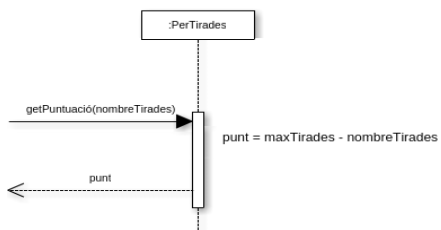
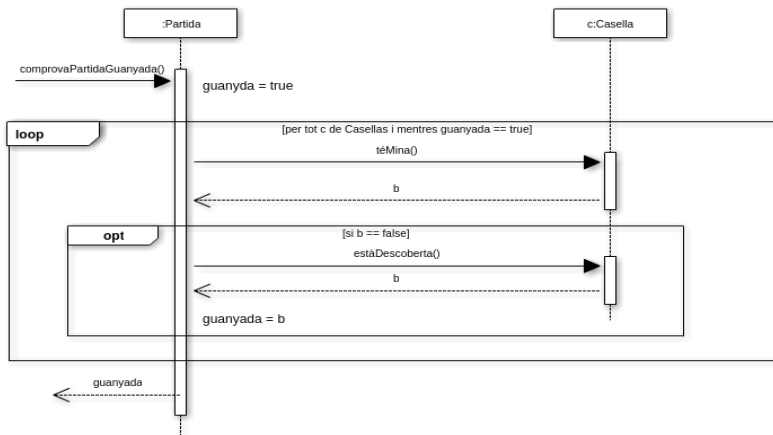
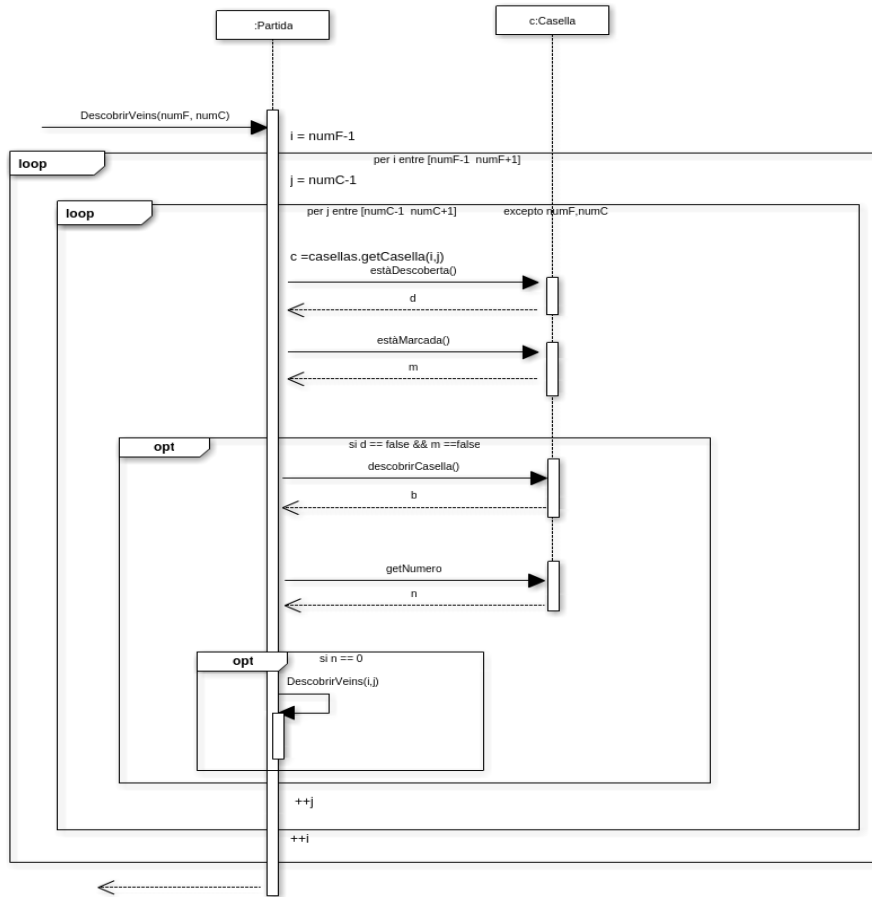
CrearPartida



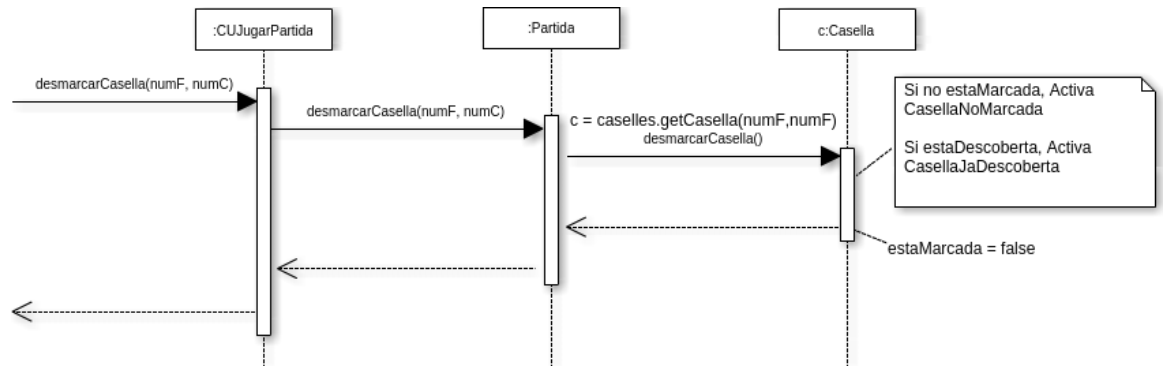


DescobrirCasella

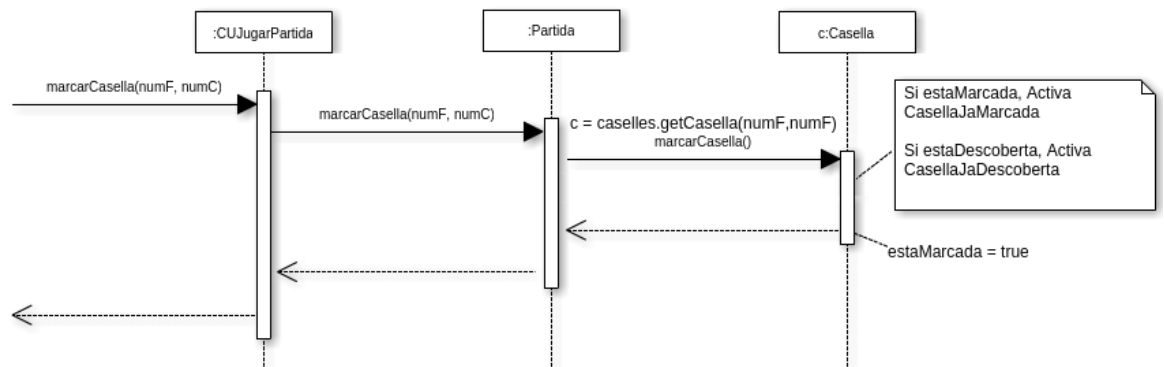




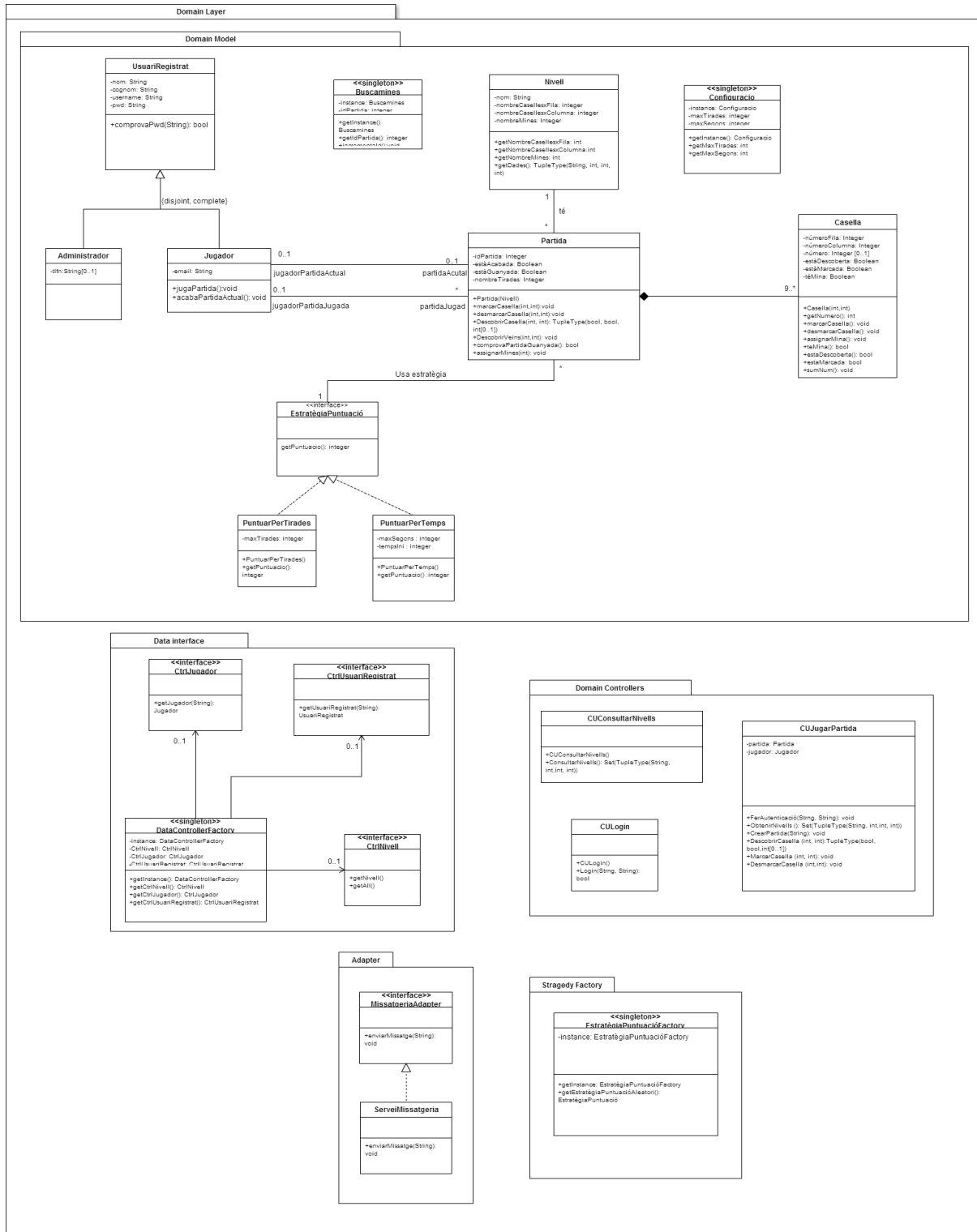
MarcarCasella



DesmarcarCasella



3. Diagrama de classes de la capa domini



4. Justificació dels patrons de disseny

Fer Autenticació

En aquesta operació fem servir el patró Factoria en la classe `DataControllerFactory` per a accedir als controladors de la capa de dades, en concret a `CtrlJugador`, `CtrlUsuariRegistrat` y `CtrlNivell`. `DataControllerFactory` encapsula les tasques de crear y obtenir aquests controladors de la capa de dades, sense especificar la classe concreta a la qual pertanyen. Això disminueix l'acoplament de les classes i capes del sistema: els controladors no s'acoplen a les classes que els necessiten, i no saben de quina classe són perquè implementen interfícies.

A més a més, `DataControllerFactory` és una classe Singleton, ja que només necessitem una instància d'aquesta classe i així tenim un accés controlat a aquest.

Crear Partida

Aquí tornem a fer servir el patró `DataControllerFactory`.

La classe `Buscamines` també segueix el patró Singleton, per les mateixes raons que `DataControllerFactory`.

També es fa servir el patró Estratègia en la operació de creació de Partida: les partides es poden puntuar segons una estratègia de tirades o una estratègia de temps. D'aquesta manera a cada partida se li pot assignar una estratègia de puntuació diferent de manera còmoda i dinàmica, fent el sistema escalable i canviaable. A més es fa servir el patró Factoria i Singleton novament en la classe `EstrategiaPuntuacióFactory`, que es comporta de forma similar a `DataControllerFactory`.

Tant si es puntua per tirades com si es puntua per temps, hem d'accedir a una classe Configuració que es Singleton perquè guarda propietats que són globals per a tot el sistema.

Descobrir Casella

En aquesta operació fem servir el patró Adaptador en la classe `MissatgeriaAdapter` per connectar-nos al servei de missatgeria i enviar un mail al jugador. Així, convertim la interfície del servei de missatgeria perquè la Partida es pugui comunicar amb aquest servei, ja que normalment no comparteixen la mateixa interfície. A més, estem fent el sistema més canviaable, ja que si en un futur vulguéssim treballar amb un altre sistema de missatgeria només hauríem de canviar el adaptador però no la classe Partida. Estem evitant un acoplament entre la classe Partida i la classe del servei de missatgeria.

Per altra banda, fem servir el patró Factoria en la classe `ServiceFactory` per tal de facilitar la introducció de nous serveis al sistema. D'aquesta manera, per a buscar un servei, es crida a la Factoria que el buscarà i en retornarà l'adaptador.