

Buscamines

Arquitectura de Software

1er Lliurament

QT14-15

Grup 13

**Hector Gutierrez
Sergi Jimenez
Alex Morral
Albert Navarro
Riogemm Panopio**

1.- Es demana que utilitzeu Hibernate (amb anotacions) per definir una part de l'esquema conceptual (les classes Partida i Nivell amb els seus atributs, l'associació entre elles i els getters i els setters de cada classe) i la part corresponent de l'esquema de la BD. En concret, es demana el codi en Java de les classes, l'esquema de la BD i un programa de prova que utilitzi les operacions definides a les classes (poseu el codi en el fitxer pdf).

Codi java de les classes:

```
package buscaminas;

import java.util.ArrayList;
import java.util.Collection;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

/**
 *
 * @author albert
 */

@Entity
@Table(name="Nivell")

public class Nivell {
    @Id private String nom;
    private int nombreCasellesxFila;
    private int nombreCasellesxColumna;
    private int nombreMines;

    @ManyToOne private Collection<Partida> partidas;

    public Nivell(String nom, int nombreCasellesxFila, int nombreCasellesxColumna, int nombreMines ) {
        this.nom = nom;
        this.nombreCasellesxFila = nombreCasellesxFila;
        this.nombreCasellesxColumna = nombreCasellesxColumna;
        this.nombreMines = nombreMines;
    }

    /**
     * @return the nom
     */
    public String getNom() {
        return nom;
    }

    /**
     * @param nom the nom to set
     */
    public void setNom(String nom) {
        this.nom = nom;
    }
}
```

```

/**
 * @return the nombreCasellesxFila
 */
public int getNombreCasellesxFila() {
    return nombreCasellesxFila;
}

/**
 * @param nombreCasellesxFila the nombreCasellesxFila to set
 */
public void setNombreCasellesxFila(int nombreCasellesxFila) {
    this.nombreCasellesxFila = nombreCasellesxFila;
}

/**
 * @return the nombreCasellesxColumna
 */
public int getNombreCasellesxColumna() {
    return nombreCasellesxColumna;
}

/**
 * @param nombreCasellesxColumna the nombreCasellesxColumna to set
 */
public void setNombreCasellesxColumna(int nombreCasellesxColumna) {
    this.nombreCasellesxColumna = nombreCasellesxColumna;
}

/**
 * @return the nombreMines
 */
public int getNombreMines() {
    return nombreMines;
}

/**
 * @param nombreMines the nombreMines to set
 */
public void setNombreMines(int nombreMines) {
    this.nombreMines = nombreMines;
}
}

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package buscaminas;

```

```

import java.util.Collection;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;
import javax.persistence.Table;

```

```

/**
 *
 * @author albert
 */
@Entity
@Table(name="Partida")

public class Partida {
    @Id private int idPartida;
    private boolean estaAcabada;
    private boolean estaGuanyada;
    private int nombreTirades;

    @OneToMany private Nivell nivell;

    public Partida(int idPartida) {
        this.idPartida=idPartida;
        this.estaAcabada=false;
        this.estaGuanyada=false;
        this.nombreTirades=0;
    }

    /**
     * @return the idPartida
     */
    public int getIdPartida() {
        return idPartida;
    }

    /**
     * @param idPartida the idPartida to set
     */
    public void setIdPartida(int idPartida) {
        this.idPartida = idPartida;
    }

    /**
     * @return the estaAcabada
     */
    public boolean isEstaAcabada() {
        return estaAcabada;
    }

    /**
     * @param estaAcabada the estaAcabada to set
     */
    public void setEstaAcabada(boolean estaAcabada) {
        this.estaAcabada = estaAcabada;
    }

    /**
     * @return the estaGuanyada
     */
    public boolean isEstaGuanyada() {

```

```

        return estaGuanyada;
    }

    /**
     * @param estaGuanyada the estaGuanyada to set
     */
    public void setEstaGuanyada(boolean estaGuanyada) {
        this.estaGuanyada = estaGuanyada;
    }

    /**
     * @return the nombreTirades
     */
    public int getNombreTirades() {
        return nombreTirades;
    }

    /**
     * @param nombreTirades the nombreTirades to set
     */
    public void setNombreTirades(int nombreTirades) {
        this.nombreTirades = nombreTirades;
    }
}

```

Codi del test per les classes:

```

package buscaminas;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

/**
 *
 * @author albert
 */
public class NivellTest {

    public NivellTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }
}

```

```

@Before
public void setUp() {
}

@After
public void tearDown() {
}

/**
 * Test of getNom method, of class Nivell.
 */
@Test
public void testGetNom() {
    System.out.println("getNom");
    Nivell instance = new Nivell("",0,0,0);
    String expectedResult = "";
    String result = instance.getNom();
    assertEquals(expResult, result);
}

/**
 * Test of setNom method, of class Nivell.
 */
@Test
public void testSetNom() {
    System.out.println("setNom");
    String nom = "";
    Nivell instance = new Nivell("",0,0,0);
    instance.setNom(nom);
    assertEquals(nom, instance.getNom());
}

/**
 * Test of getNombreCasellesxFila method, of class Nivell.
 */
@Test
public void testGetNombreCasellesxFila() {
    System.out.println("getNombreCasellesxFila");
    Nivell instance = new Nivell("",0,0,0);
    int expectedResult = 0;
    int result = instance.getNombreCasellesxFila();
    assertEquals(expResult, result);
}

/**
 * Test of setNombreCasellesxFila method, of class Nivell.
 */
@Test
public void testSetNombreCasellesxFila() {
    System.out.println("setNombreCasellesxFila");
    int nombreCasellesxFila = 0;
    Nivell instance = new Nivell("",0,0,0);
    instance.setNombreCasellesxFila(nombreCasellesxFila);
    assertEquals(nombreCasellesxFila, instance.getNombreCasellesxFila());
}

/**

```

* Test of getNombreCasellesxColumna method, of class Nivell.

*/

@Test

```
public void testGetNombreCasellesxColumna() {  
    System.out.println("getNombreCasellesxColumna");  
    Nivell instance = new Nivell("",0,0,0);  
    int expResult = 0;  
    int result = instance.getNombreCasellesxColumna();  
    assertEquals(expResult, result);  
}
```

/**

* Test of setNombreCasellesxColumna method, of class Nivell.

*/

@Test

```
public void testSetNombreCasellesxColumna() {  
    System.out.println("setNombreCasellesxColumna");  
    int nombreCasellesxColumna = 0;  
    Nivell instance = new Nivell("",0,0,0);  
    instance.setNombreCasellesxColumna(nombreCasellesxColumna);  
    assertEquals(nombreCasellesxColumna, instance.getNombreCasellesxColumna());  
}
```

/**

* Test of getNombreMines method, of class Nivell.

*/

@Test

```
public void testGetNombreMines() {  
    System.out.println("getNombreMines");  
    Nivell instance = new Nivell("",0,0,0);  
    int expResult = 0;  
    int result = instance.getNombreMines();  
    assertEquals(expResult, result);  
}
```

/**

* Test of setNombreMines method, of class Nivell.

*/

@Test

```
public void testSetNombreMines() {  
    System.out.println("setNombreMines");  
    int nombreMines = 0;  
    Nivell instance = new Nivell("",0,0,0);  
    instance.setNombreMines(nombreMines);  
    assertEquals(nombreMines, instance.getNombreMines());  
}
```

}

}

```

package buscaminas;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

/**
 *
 * @author albert
 */
public class PartidaTest {

    public PartidaTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    /**
     * Test of getldPartida method, of class Partida.
     */
    @Test
    public void testGetldPartida() {
        System.out.println("getldPartida");
        Partida instance = new Partida(0);
        int expResult = 0;
        int result = instance.getldPartida();
        assertEquals(expResult, result);
    }

    /**
     * Test of setldPartida method, of class Partida.
     */
    @Test
    public void testSetldPartida() {
        System.out.println("setldPartida");
        int idPartida = 0;
        Partida instance = new Partida(0);
        instance.setldPartida(idPartida);
        assertEquals(idPartida, instance.getldPartida());
    }
}

```



```

/**
 * Test of isEstaAcabada method, of class Partida.
 */
@Test
public void testIsEstaAcabada() {
    System.out.println("isEstaAcabada");
    Partida instance = new Partida(0);
    boolean expResult = false;
    boolean result = instance.isEstaAcabada();
    assertEquals(expResult, result);
}

/**
 * Test of setEstaAcabada method, of class Partida.
 */
@Test
public void testSetEstaAcabada() {
    System.out.println("setEstaAcabada");
    boolean estaAcabada = false;
    Partida instance = new Partida(0);
    instance.setEstaAcabada(estaAcabada);
    assertEquals(estaAcabada, instance.isEstaAcabada());
}

/**
 * Test of isEstaGuanyada method, of class Partida.
 */
@Test
public void testIsEstaGuanyada() {
    System.out.println("isEstaGuanyada");
    Partida instance = new Partida(0);
    boolean expResult = false;
    boolean result = instance.isEstaGuanyada();
    assertEquals(expResult, result);
}

/**
 * Test of setEstaGuanyada method, of class Partida.
 */
@Test
public void testSetEstaGuanyada() {
    System.out.println("setEstaGuanyada");
    boolean estaGuanyada = false;
    Partida instance = new Partida(0);
    instance.setEstaGuanyada(estaGuanyada);
    assertEquals(estaGuanyada, instance.isEstaGuanyada());
}

```

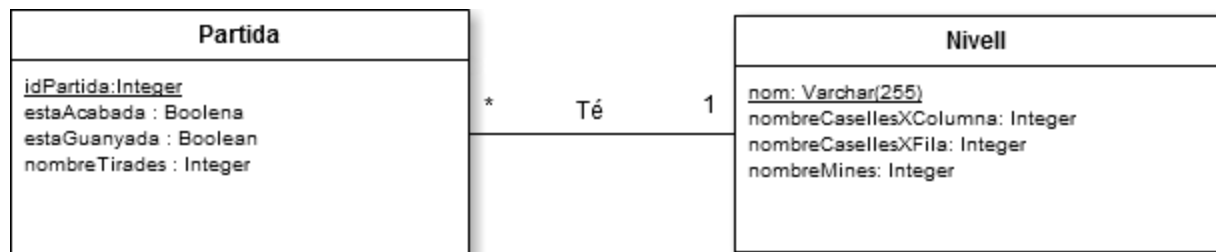
```

/**
 * Test of getNombreTirades method, of class Partida.
 */
@Test
public void testGetNombreTirades() {
    System.out.println("getNombreTirades");
    Partida instance = new Partida(0);
    int expectedResult = 0;
    int result = instance.getNombreTirades();
    assertEquals(expectedResult, result);
}

/**
 * Test of setNombreTirades method, of class Partida.
 */
@Test
public void testSetNombreTirades() {
    System.out.println("setNombreTirades");
    int nombreTirades = 0;
    Partida instance = new Partida(0);
    instance.setNombreTirades(nombreTirades);
    assertEquals(nombreTirades, instance.getNombreTirades());
}
}

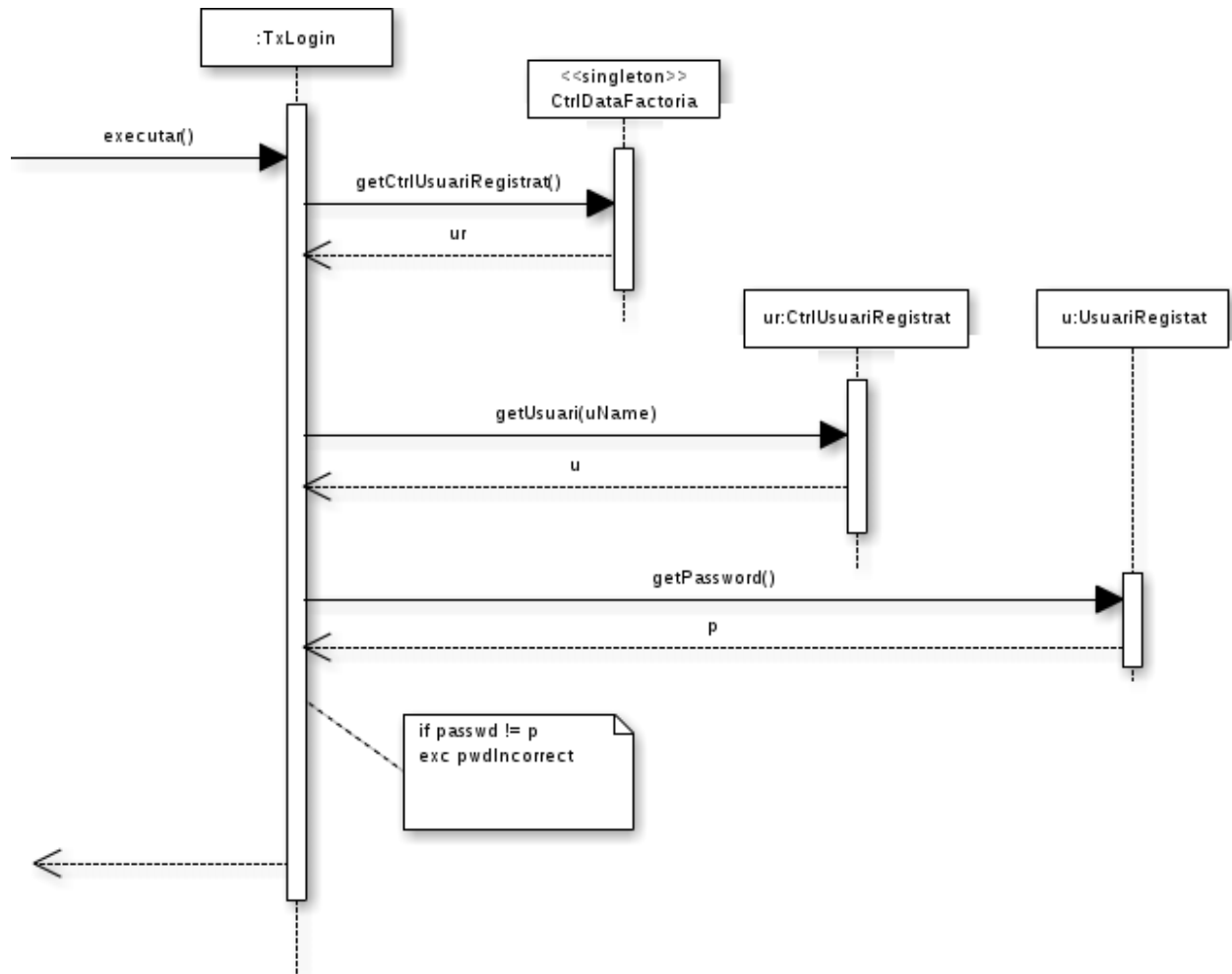
```

Esquema de la base de dades:

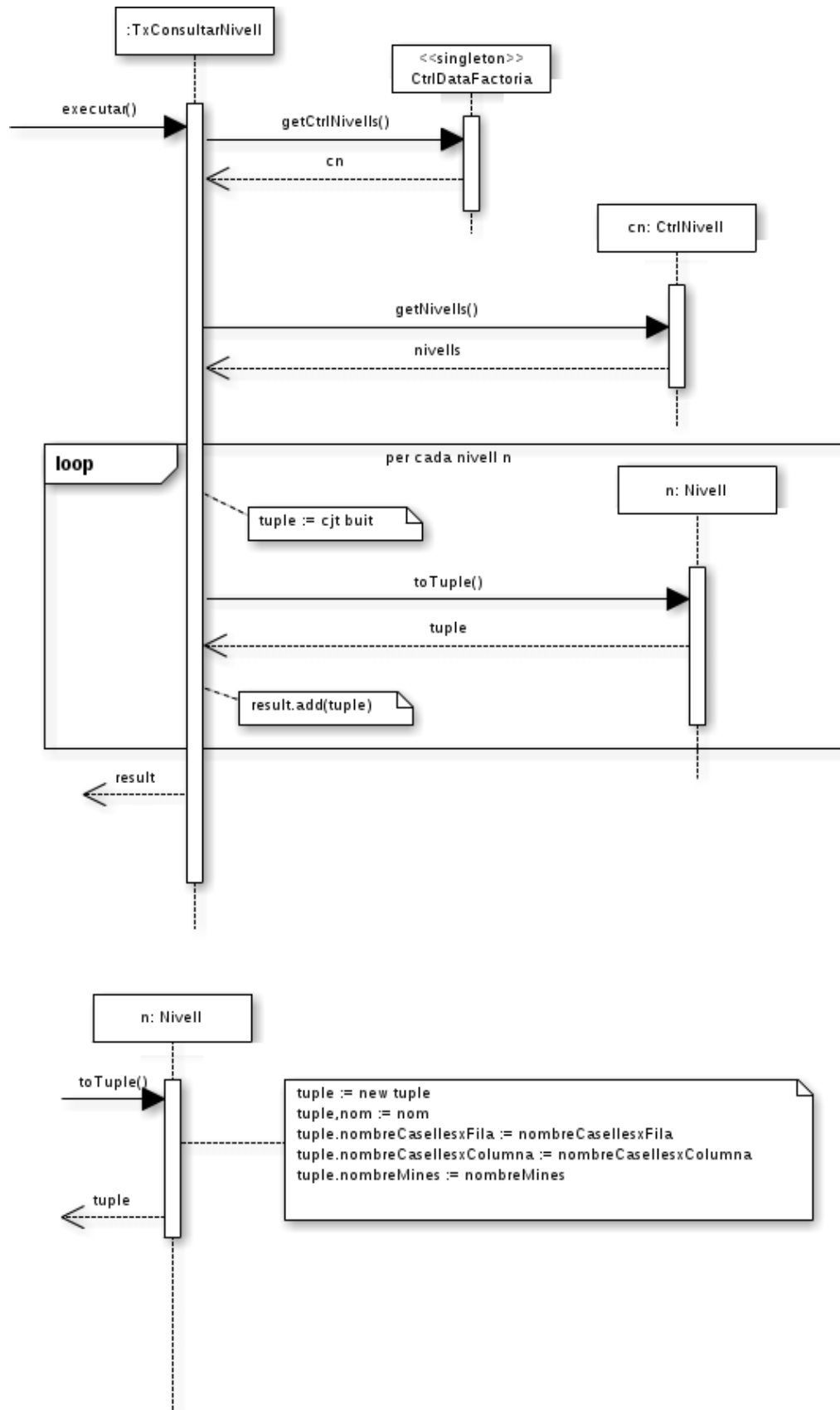


2.- Diagrames de seqüència de les operacions de la capa de domini corresponents al cas d'ús Jugar partida.

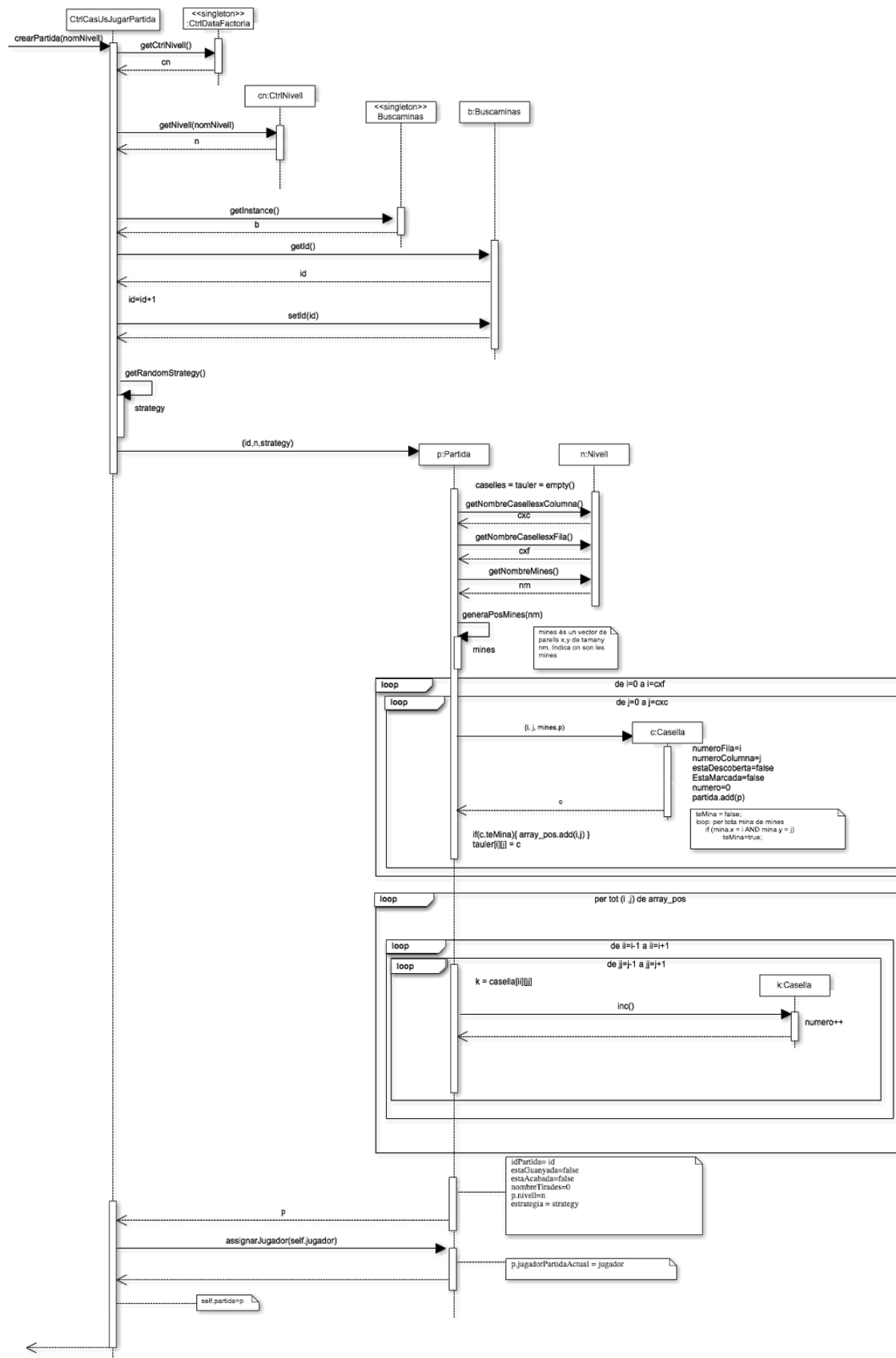
Cas D'Ús Login:



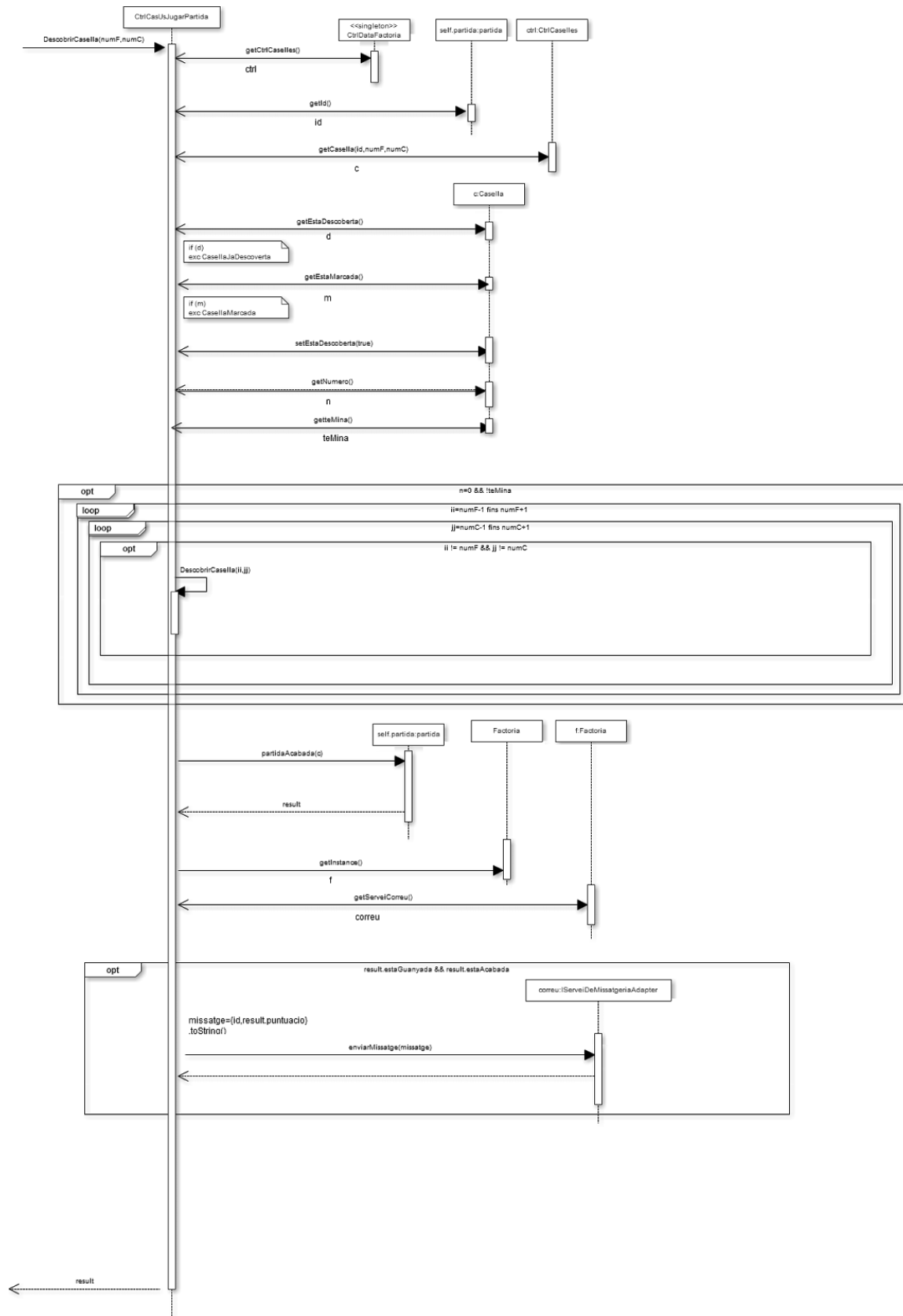
Cas D'Ús Consultar Nivells:



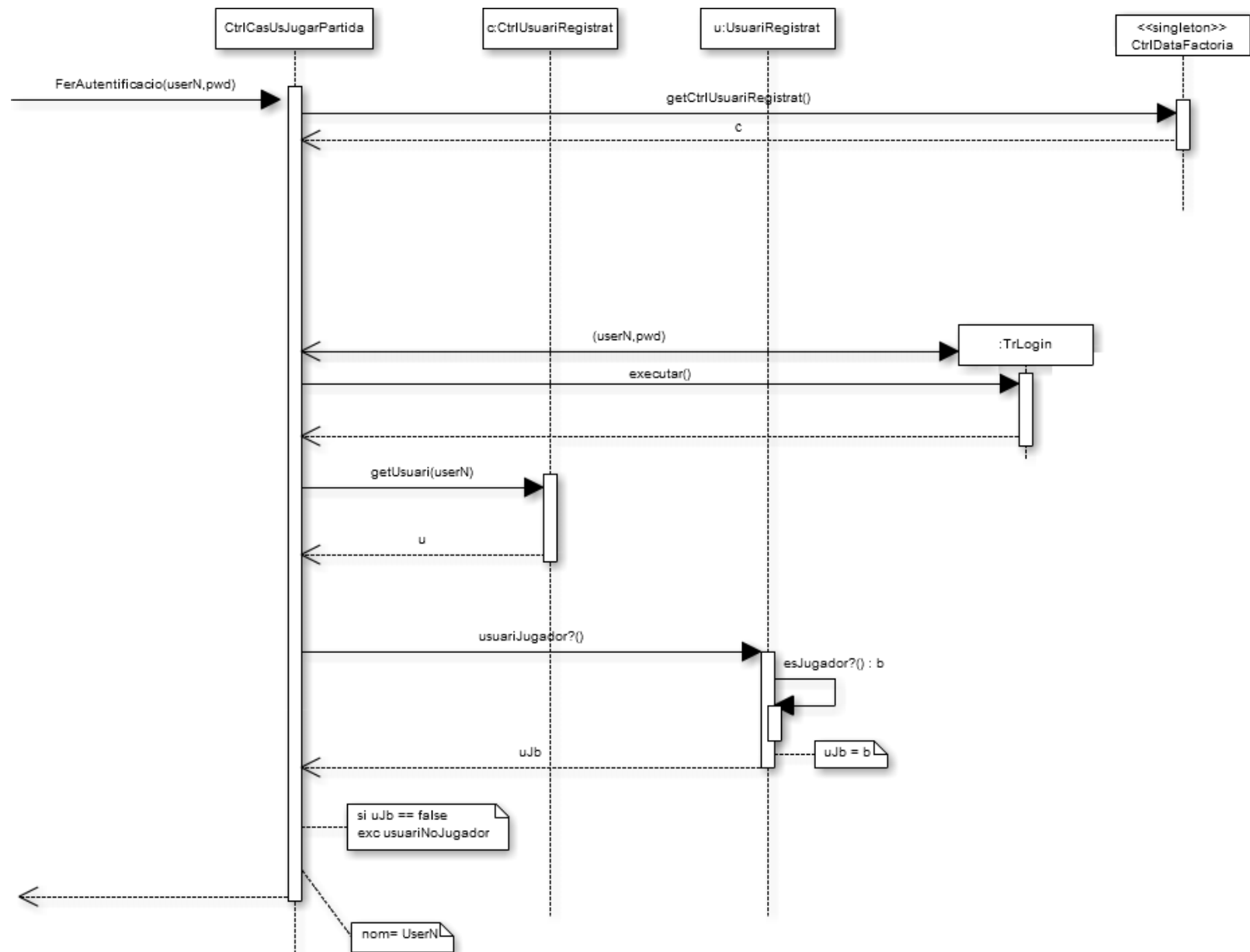
Cas D'Ús Jugar Partida:



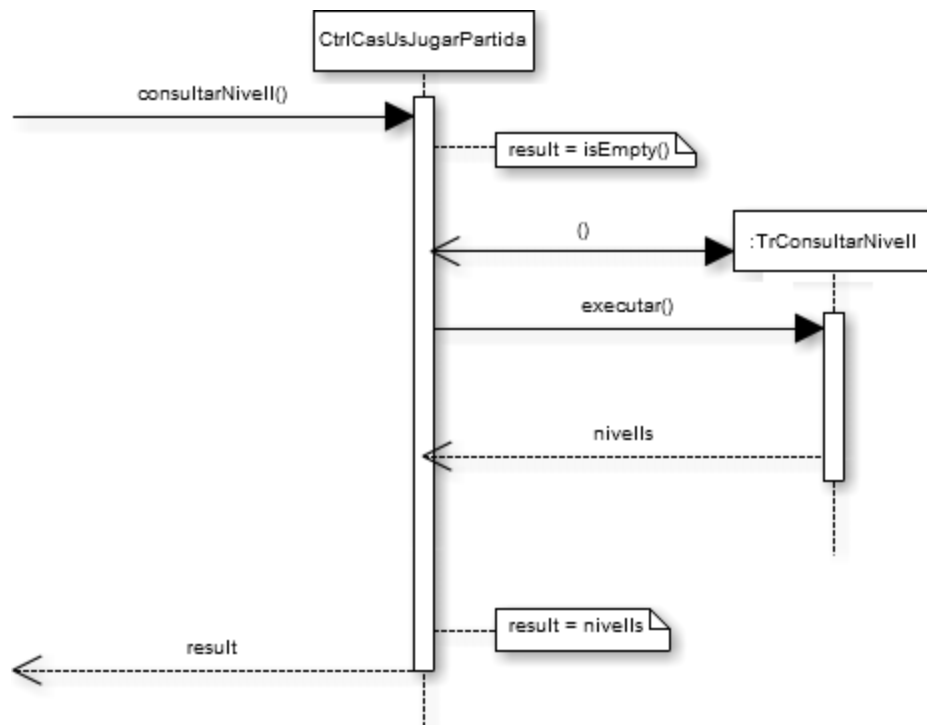
DescobrirCasella:



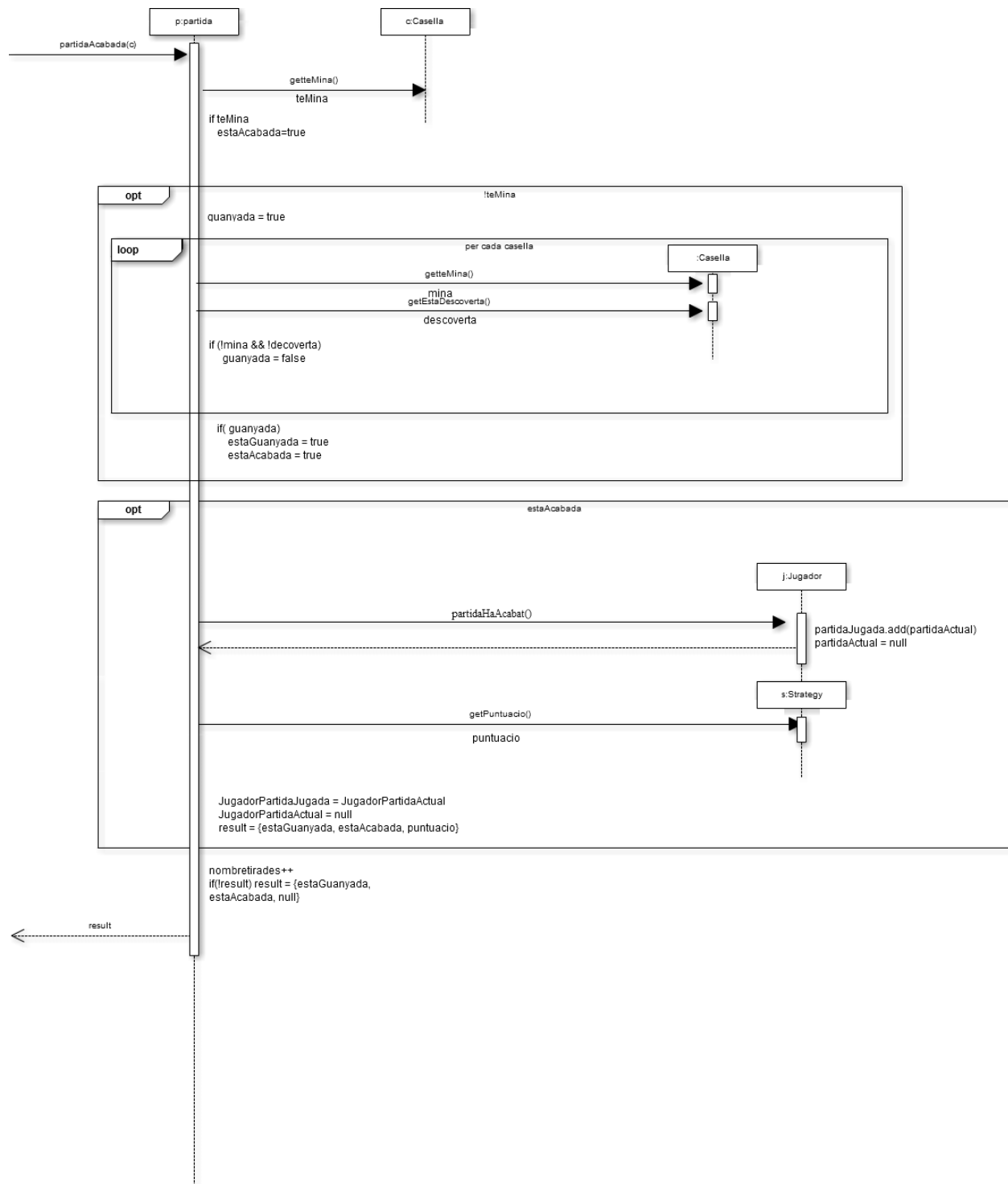
FerAutenticació:



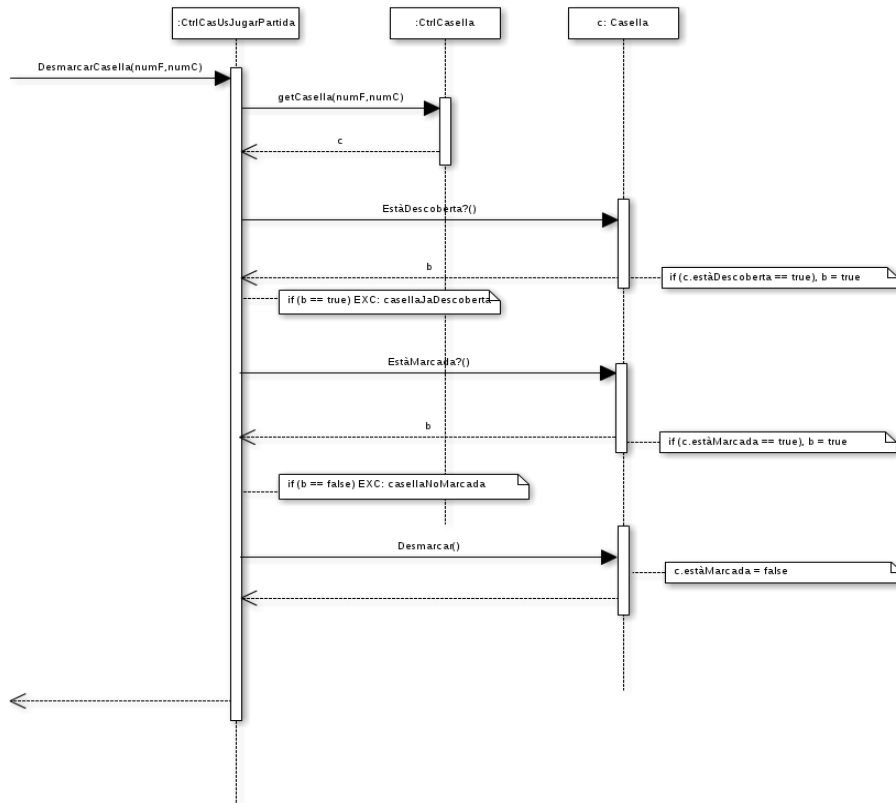
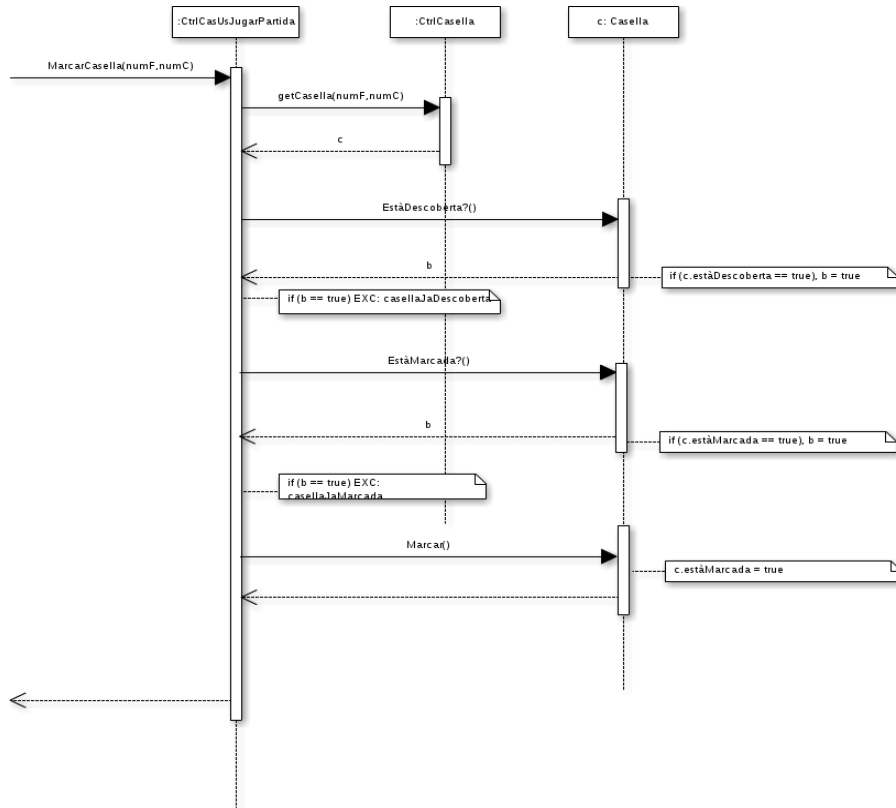
ObtenirNivells:



partidaAcabada:



MarcarCasella i DesmarcarCasella:



3.- Diagrama de classes de la capa de domini. Heu de fer servir un controlador de cas d'ús per guardar l'estat del cas d'ús.

Diagrama de classes de Domain Model i Adapters:

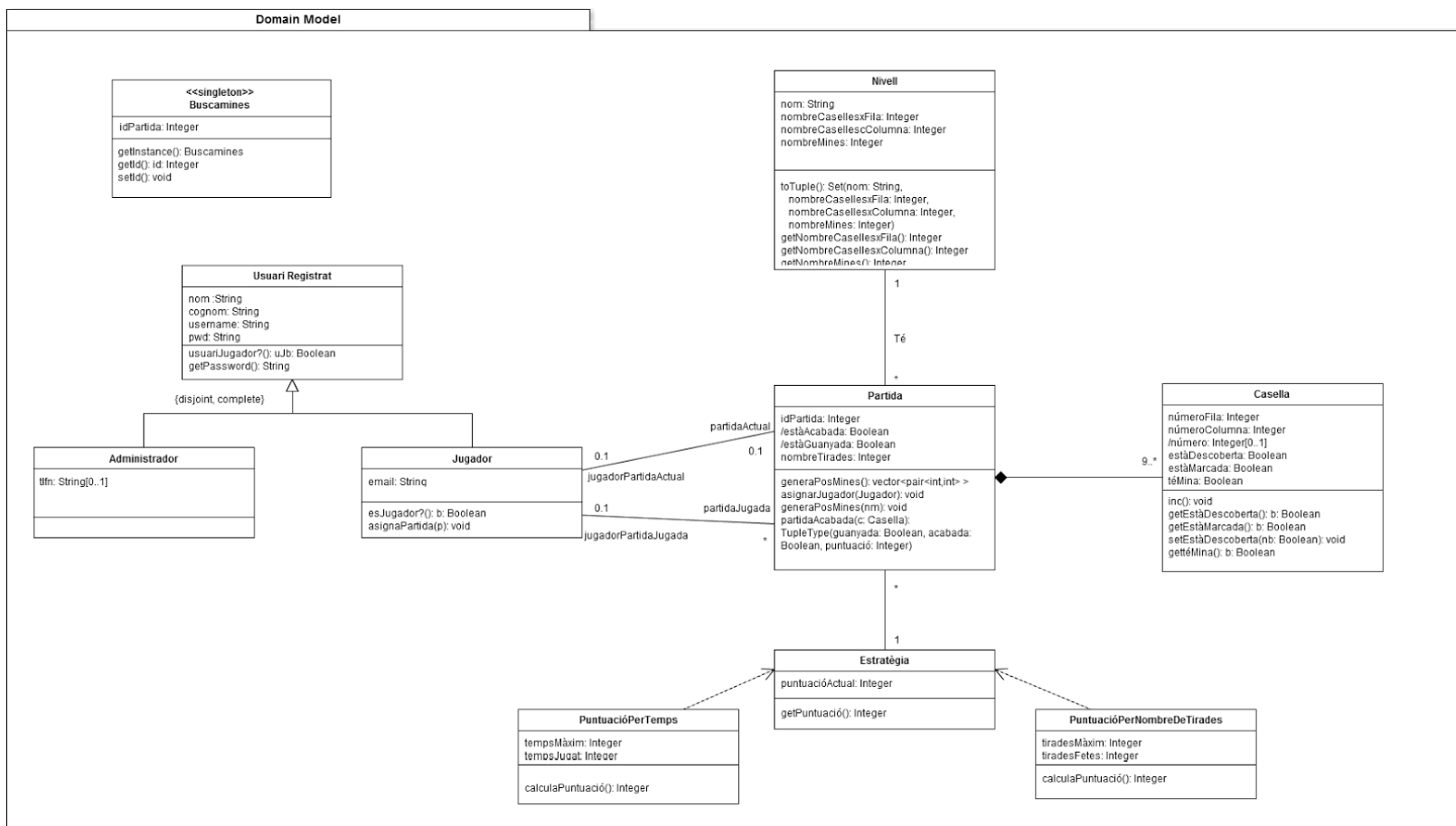
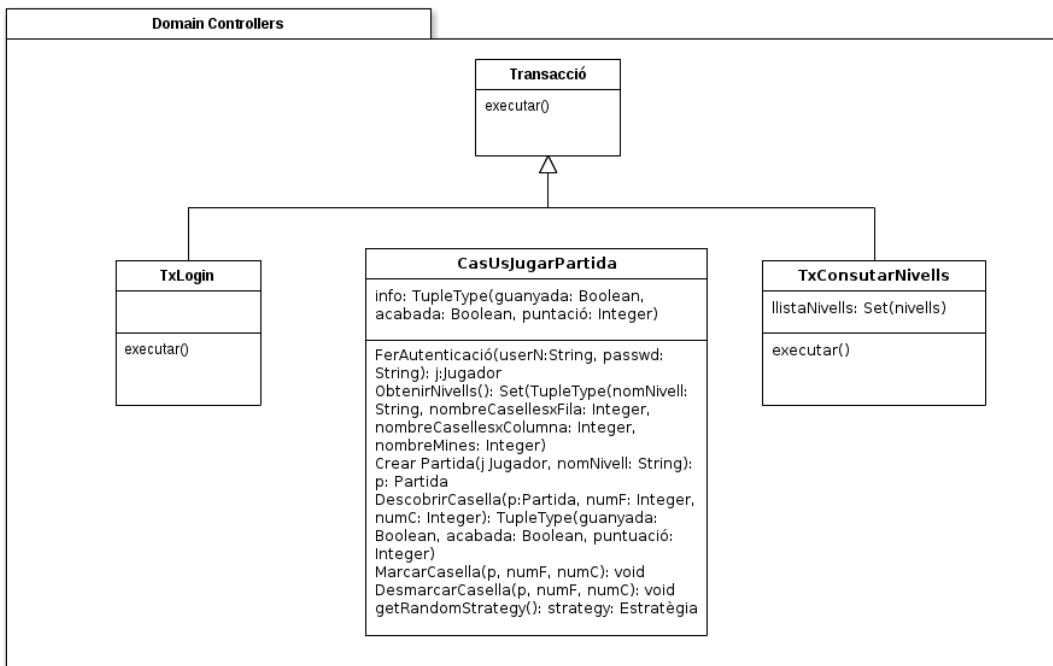
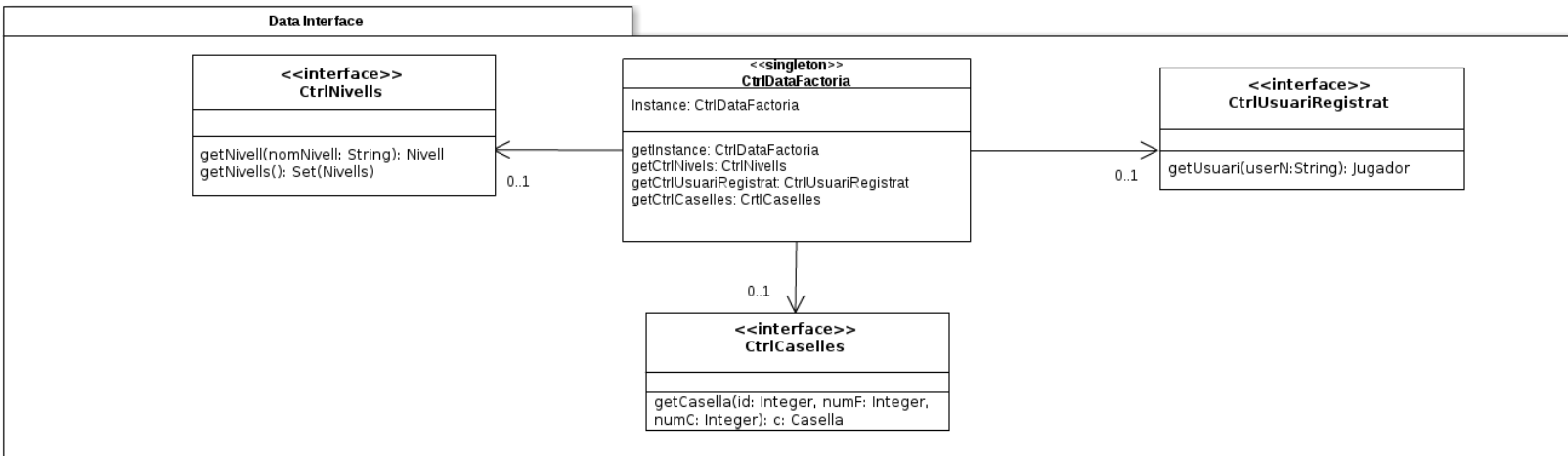


Diagrama de classes de Data Interface i Domain Controllers:



4.- (Competència transversal) Justifiqueu de forma raonada els patrons de disseny que heu aplicat en aquesta pràctica.

A lo largo de la práctica nos hemos encontrado con la decisión de implementar diferentes patrones de diseño, para mejorar la cambiabilidad, la reusabilidad, y la efectividad de nuestro sistema entre los cuales podemos encontrar (*singleton*, *adapter*, *abstract factory*,...).

A continuación se detalla el uso de los diferentes patrones usados hasta el momento

- Patrón de creación **Singleton**

Este patrón se manifiesta en varios de los objetos, de hecho en aquellos a los que hemos querido otorgar un carácter de singularidad y exclusividad, es decir, en aquellos que hemos querido garantizar que sólo existe una instancia, asegurándonos así de que todos los objetos que hagan uso de esa clase utilizarán esa misma instancia.

Este patrón lo hemos requerido para aquellas clases que deben encargarse de gestionar algún recurso, ya sea externo (conexiones, la Factoria..) o bien interno (configuración y estado global, Buscaminas).

- Patrón **Abstract Factory**

Usado para mantener la independencia entre la creación y el uso de los controladores, así nos aseguramos de que nuestro sistema es más cambiable.

- Patrón **Adapter**

Hemos usado este patrón para la interfaz *IServeiDeMisatgeriaAdapter*, para así poder interaccionar con el servicio de mensajería externo. Es aconsejable usar este tipo de patrón ya que el sistema no tiene la obligación de conocer a fondo el mecanismo del servicio externo, lanzando así peticiones mas genéricas que luego el *adapter* se encargará de adaptar, como bien dice el nombre, para que el servicio externo las interprete y ejecute la acción correspondiente.

- Patrón **Template Method**

Para el control de los diferentes tipos de estrategias y el cálculo de las puntuaciones hemos usado el patrón *template method*, ya que todas las subclases de estrategia tienen sus particularidades a la hora de hacer los cálculos pero en gran parte, se comparte el código ya que tienen el mismo objetivo, calcular la puntuación final de una partida finalizada con *estaAcabada a true* y *estaGuanyada a true*. De otra forma no llega a realizar el cálculo de puntuación.