2048

abril 1

Capa de dominio

Victor Dubé
Joan Oliva
Miguel Angel Aula
Fco. Javier Gárate

Grau AS - FIB

Contenido

1. Ejercicio 1 3
   a. Prólogo 3
   b. Tablas de la base de datos 4
      i. Tabla casella 4
      ii. Tabla partidas 4
   c. Clases Dominio 5
      i. Partidas 5
      ii. CasellaId 7
      iii. Casella 9
   d. Hibernate Configuration Files 10
      i. Hibernate.cfg.xml 10
      ii. Hibernate.reveng.xml 10
      iii. Casella.hbm.xml 10
      iv. Partidas.hbm.xml 11
      v. Hibernate.Util.java 11
   e. Testing tool 13
      i. Menu.java 13
      ii. NuevaPartidaDialog.java 17
      iii. CasillasInPartida.java 22
2. Ejercicio 2 28
   a. Caso de uso Login 28
      i. Contrato login 28
   b. Caso de uso Jugar partida 28
      i. Contrato FerAutenticacio 28
      ii. Contrato CrearPartida 29
      iii. Contrato FerMoviment 31
      iv. Contrato ObtenirRanking 44
   c. Caso de uso Consultar Ranking 45
   d. Utilitats comunes entre contractes i casos d'us 49
3. Ejercicio 3 54
4. Ejercicio 4 55
   a. Patró adaptador 55
   b. Patró estratègia i patró plantilla 55

c. Patró singleton 55
d. Patró iterador 56

a.Ejercicio 1

a. Prólogo

Este documento contiene el código que se ejecuta en el video enlazado a continuación.

https://drive.google.com/file/d/0BxCEkST1S7jLOWdMRldrRlZsUWs/view?usp=sharing

El sistema de testing está formado por una base de datos POSTGRESQL, que debido al gran número de fallos y problemas que hemos tenido, probablemente no sea la plataforma a utilizar en el futuro, unas clases de dominio en el package as.entity, unos archivos de configuración de hibernate, y unas clases de interfaz para las pruebas.

Como son pruebas muy especificas de un sistema parcial, estas clases de testing se encargan de hacer consultas a la base de datos y de mostrar un menú que nos permite, crear una nueva partida, o ver/editar las casillas de una partida existente.

b. Tablas de la base de datos

i. Tabla casella

```
CREATE TABLE casella
(
idpartida integer NOT NULL,
numerofila integer NOT NULL,
numerocolumna integer NOT NULL,
numero integer,
CONSTRAINT "Casella_pkey" PRIMARY KEY (idpartida, numerofila, numerocolumna),
  CONSTRAINT "Casella_idPartida_fkey" FOREIGN KEY (idpartida)
REFERENCES partidas (idpartida) MATCH SIMPLE
     ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

idpartida [PK] integer	numerofila [PK] integer	numerocolumna [PK] integer	numero integer

ii. Tabla partidas

```
CREATE TABLE partidas
(
idpartida integer NOT NULL,
estaacabada boolean,
estajugada boolean,
puntuacion integer,
  CONSTRAINT "PK" PRIMARY KEY (idpartida)
)
```

idpartida [PK] integer	estaacabada boolean	estajugada boolean	puntuacion integer

c. Clases Dominio

i. Partidas

```
package as.entity;
import java.util.HashSet;
import java.util.Set;
import org.hibernate.Session;

/**
 * Partidas generated by hbm2java
*/
public class Partidas  implements java.io.Serializable {
private int idPartida;
private Boolean estaAcabada;
private Boolean estaJugada;
private Integer puntuacion;
private Set casellas = new HashSet(16);

public Partidas() {
   }

public Partidas(int idPartida) {
       this.idPartida = idPartida;

      this.estaAcabada = false;
      this.estaJugada = false;
      this.puntuacion = 0;
for (int i = 0; i<4; ++i){
for (int j = 0; j<4; ++j){
this.casellas.add(new Casella(i,j,0,this));
}

      }
   }
public Partidas(int idPartida, Boolean estaAcabada, Boolean estaJugada, Integer puntuacion,
Set casellas) {
      this.idPartida = idPartida;
      this.estaAcabada = estaAcabada;
      this.estaJugada = estaJugada;
      this.puntuacion = puntuacion;
this.casellas = casellas;
   }

public int getIdPartida() {
return this.idPartida;
   }

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

```
public Boolean getEstaAcabada() {
return this.estaAcabada;
}

public void setEstaAcabada(Boolean estaAcabada) {
      this.estaAcabada = estaAcabada;
   }
public Boolean getEstaJugada() {
return this.estaJugada;
   }

public void setEstaJugada(Boolean estaJugada) {
      this.estaJugada = estaJugada;
   }
public Integer getPuntuacion() {
return this.puntuacion;
   }

public void setPuntuacion(Integer puntuacion) {
this.puntuacion = puntuacion;
   }
public Set getCasellas() {
return this.casellas;
   }

public void setCasellas(Set casellas) {
      this.casellas = casellas;
   }

public void persiste(Session sesion) {
sesion.beginTransaction();
sesion.persist(this);
for (Object c : casellas) {
sesion.persist((Casella)c);
}
sesion.getTransaction().commit();

   }
}
```

ii. CasellaId

```
package as.entity;

/**
 * CasellaId generated by hbm2java
 */
public class CasellaId  implements java.io.Serializable {
private int idPartida;
private int numeroFila;
private int numeroColumna;

public CasellaId() {
   }

public CasellaId(int idPartida, int numeroFila, int numeroColumna) {
this.idPartida = idPartida;
      this.numeroFila = numeroFila;
      this.numeroColumna = numeroColumna;
   }

public int getIdPartida() {
return this.idPartida;
   }

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

public void setNumeroFila(int numeroFila) {
       this.numeroFila = numeroFila;
   }
public int getNumeroColumna() {
return this.numeroColumna;
   }

public void setNumeroColumna(int numeroColumna) {
       this.numeroColumna = numeroColumna;
   }
public boolean equals(Object other) {
if ( (this == other ) ) return true;
                     if ( (other == null ) ) return false;
                     if ( !(other instanceof CasellaId) ) return false;
                      CasellaId castOther = ( CasellaId ) other;

                     return (this.getIdPartida()==castOther.getIdPartida())
&& (this.getNumeroFila()==castOther.getNumeroFila())
&& (this.getNumeroColumna()==castOther.getNumeroColumna());
  }
```

```
public int hashCode() {
int result = 17;

result = 37 * result + this.getIdPartida();
result = 37 * result + this.getNumeroFila();
result = 37 * result + this.getNumeroColumna();
return result;
}


}
```

iii. Casella

```
package as.entity;

/**
 * Casella generated by hbm2java
 */
public class Casella  implements java.io.Serializable {
private CasellaId id;
private Partidas partidas;
private Integer numero;

public Casella() {
   }
public Casella(CasellaId id, Partidas partidas) {
this.id = id;
this.partidas = partidas;
   }
public Casella(int fila, int columna, int punt, Partidas partidas) {
       this.id = new CasellaId(partidas.getIdPartida(),fila,columna);
       this.partidas = partidas;
       this.numero = punt;
   }
public Casella(CasellaId id, Partidas partidas, Integer numero) {
      this.id = id;
      this.partidas = partidas;
this.numero = numero;
   }

public CasellaId getId() {
return this.id;
   }

public void setId(CasellaId id) {
this.id = id;
   }
public Partidas getPartidas() {
return this.partidas;
   }

public void setPartidas(Partidas partidas) {
       this.partidas = partidas;
   }
public Integer getNumero() {
return this.numero;
}

public void setNumero(Integer numero) {
       this.numero = numero;
   }
   }
```

d. Hibernate Configuration Files

i. Hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD
3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
<property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
<property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/AS1</property>
<property name="hibernate.connection.username">usras</property>
<property name="hibernate.connection.password">usrASusrAS</property>
<property name="hibernate.show_sql">true</property>
<property
name="hibernate.query.factory_class">org.hibernate.hql.internal.classic.ClassicQueryTranslat
orFactory</property>
<mapping resource="as/entity/Casella.hbm.xml"/>
<mapping resource="as/entity/Partidas.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

ii. Hibernate.reveng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-reverse-engineering PUBLIC "-//Hibernate/Hibernate Reverse
Engineering DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-reverse-engineering-
3.0.dtd">
<hibernate-reverse-engineering>
<schema-selection match-catalog="AS1" match-schema="public"/>
<table-filter match-name="partidas"/>
<table-filter match-name="casella"/>
</hibernate-reverse-engineering>
```

iii. Casella.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated 07-abr-2015 16:04:59 by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
<class name="as.entity.Casella" table="casella" schema="public" optimistic-lock="version">
<composite-id name="id" class="as.entity.CasellaId">
<key-property name="idPartida" type="int">
<column name="idpartida" />
</key-property>
<key-property name="numeroFila" type="int">
<column name="numerofila" />
</key-property>
<key-property name="numeroColumna" type="int">
<column name="numerocolumna" />
```

```
</key-property>
</composite-id>
<many-to-one name="partidas" class="as.entity.Partidas" update="false" insert="false"
fetch="select">
<column name="idpartida" not-null="true" />
</many-to-one>
<property name="numero" type="java.lang.Integer">
<column name="numero" />
</property>
</class>
</hibernate-mapping>
```

iv. Partidas.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated 07-abr-2015 16:04:59 by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
<class name="as.entity.Partidas" table="partidas" schema="public" optimistic-lock="version">
<id name="idPartida" type="int">
<column name="idpartida" />
<generator class="assigned" />
</id>
<property name="estaAcabada" type="java.lang.Boolean">
<column name="estaacabada" />
</property>
<property name="estaJugada" type="java.lang.Boolean">
<column name="estajugada" />
</property>
<property name="puntuacion" type="java.lang.Integer">
<column name="puntuacion" />
</property>
<set name="casellas" table="casella" inverse="true" lazy="true" fetch="select">
<key>
<column name="idpartida" not-null="true" />
</key>
<one-to-many class="as.entity.Casella" />
</set>
</class>
</hibernate-mapping>
```

v. Hibernate.Util.java

```
/*
 * 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 as.util;

import org.hibernate.cfg.AnnotationConfiguration;
```

```
import org.hibernate.SessionFactory;

/**
* Hibernate Utility class with a convenient method to get Session Factory
 * object.
 *
 * @author fjgarate
 */
public class HibernateUtil {

private static final SessionFactory sessionFactory;

static {
try {
          // Create the SessionFactory from standard (hibernate.cfg.xml)
          // config file.
sessionFactory = new AnnotationConfiguration().configure().buildSessionFactory();
       } catch (Throwable ex) {
          // Log the exception.
System.err.println("Initial SessionFactory creation failed." + ex);
throw new ExceptionInInitializerError(ex);
       }
   }

public static SessionFactory getSessionFactory() {
return sessionFactory;
}
}
```

e. Testing tool

i. Menu.java

```
/*
 * 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 as.gui;

import as.entity.Partidas;
import as.util.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.criterion.Order;

/**
 *
 * @author fjgarate
 */
public class Menu extends javax.swing.JFrame {

    /**
     * Creates new form Menu
     */
public Menu() {
initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
@SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
menuBar = new javax.swing.JMenuBar();
fileMenu = new javax.swing.JMenu();
openMenuItem = new javax.swing.JMenuItem();
saveMenuItem = new javax.swing.JMenuItem();
saveAsMenuItem = new javax.swing.JMenuItem();
exitMenuItem = new javax.swing.JMenuItem();
editMenu = new javax.swing.JMenu();
cutMenuItem = new javax.swing.JMenuItem();
copyMenuItem = new javax.swing.JMenuItem();
pasteMenuItem = new javax.swing.JMenuItem();
deleteMenuItem = new javax.swing.JMenuItem();
helpMenu = new javax.swing.JMenu();
```

```
contentsMenuItem = new javax.swing.JMenuItem();
aboutMenuItem = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jButton1.setText("Find Partida");
jButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton1ActionPerformed(evt);
        }
    });

jButton2.setText("New Partida");
jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
        }
    });

fileMenu.setMnemonic('f');
fileMenu.setText("File");

openMenuItem.setMnemonic('o');
openMenuItem.setText("Open");
fileMenu.add(openMenuItem);

saveMenuItem.setMnemonic('s');
saveMenuItem.setText("Save");
fileMenu.add(saveMenuItem);

saveAsMenuItem.setMnemonic('a');
saveAsMenuItem.setText("Save As ...");
saveAsMenuItem.setDisplayedMnemonicIndex(5);
fileMenu.add(saveAsMenuItem);

exitMenuItem.setMnemonic('x');
exitMenuItem.setText("Exit");
exitMenuItem.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
exitMenuItemActionPerformed(evt);
        }
    });
fileMenu.add(exitMenuItem);

menuBar.add(fileMenu);

editMenu.setMnemonic('e');
editMenu.setText("Edit");

cutMenuItem.setMnemonic('t');
cutMenuItem.setText("Cut");
editMenu.add(cutMenuItem);
```

```
copyMenuItem.setMnemonic('y');
copyMenuItem.setText("Copy");
editMenu.add(copyMenuItem);

pasteMenuItem.setMnemonic('p');
pasteMenuItem.setText("Paste");
editMenu.add(pasteMenuItem);

deleteMenuItem.setMnemonic('d');
deleteMenuItem.setText("Delete");
editMenu.add(deleteMenuItem);

menuBar.add(editMenu);

helpMenu.setMnemonic('h');
helpMenu.setText("Help");

contentsMenuItem.setMnemonic('c');
contentsMenuItem.setText("Contents");
helpMenu.add(contentsMenuItem);

aboutMenuItem.setMnemonic('a');
aboutMenuItem.setText("About");
helpMenu.add(aboutMenuItem);

menuBar.add(helpMenu);

setJMenuBar(menuBar);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap(254, Short.MAX_VALUE)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(21, 21, 21))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
               .addGap(20, 20, 20)
               .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
javax.swing.GroupLayout.PREFERRED_SIZE)
               .addContainerGap(255, Short.MAX_VALUE)))
     );
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
```

```
            .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 257,
Short.MAX_VALUE)
            .addContainerGap())
         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
               .addContainerGap()
               .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE, 257,
Short.MAX_VALUE)
               .addContainerGap()))
      );

pack();
   }// </editor-fold>

private void exitMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);
   }

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// Find partidas
      CasillasInPartida cip = new CasillasInPartida();
cip.setVisible(true);
   }

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      // New Partida
      Session session = HibernateUtil.getSessionFactory().openSession();
// Coger el id máximo
      Partidas newest = (Partidas) session.createCriteria(Partidas.class)
.addOrder(Order.desc("idPartida"))
   .setMaxResults(1)
   .uniqueResult();
int nextid = 1;
if (newest != null) nextid = newest.getIdPartida() +1;
Partidas p = new Partidas(nextid);
p.persiste(session);
session.close();
NuevaPartidaDialog diag = new NuevaPartidaDialog(this,true,nextid);
diag.setVisible(true);
   }

   /**
    * @param args the command line arguments
    */
public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */
try {
```

```
for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName());
break;
            }
          }
      } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
      } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
      } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
      }
      //</editor-fold>

      /* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new Menu().setVisible(true);
         }
      });
   }

   // Variables declaration - do not modify
private javax.swing.JMenuItem aboutMenuItem;
private javax.swing.JMenuItem contentsMenuItem;
private javax.swing.JMenuItem copyMenuItem;
private javax.swing.JMenuItem cutMenuItem;
private javax.swing.JMenuItem deleteMenuItem;
private javax.swing.JMenu editMenu;
private javax.swing.JMenuItem exitMenuItem;
private javax.swing.JMenu fileMenu;
private javax.swing.JMenu helpMenu;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JMenuBar menuBar;
private javax.swing.JMenuItem openMenuItem;
private javax.swing.JMenuItem pasteMenuItem;
private javax.swing.JMenuItem saveAsMenuItem;
private javax.swing.JMenuItem saveMenuItem;
   }
```

ii. NuevaPartidaDialog.java

```
/*
 * 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 as.gui;

import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import javax.swing.AbstractAction;
import javax.swing.ActionMap;
import javax.swing.InputMap;
import javax.swing.JComponent;
import javax.swing.KeyStroke;

/**
 *
 * @author fjgarate
*/
public class NuevaPartidaDialog extends javax.swing.JDialog {

/**
    * A return status code - returned if Cancel button has been pressed
    */
public static final int RET_CANCEL = 0;
   /**
    * A return status code - returned if OK button has been pressed
    */
public static final int RET_OK = 1;

   /**
    * Creates new form NuevaPartidaDialog
    */
public NuevaPartidaDialog(java.awt.Frame parent, boolean modal, Integer nueid) {
super(parent, modal);
initComponents();
this.newpid.setText(nueid.toString());

      // Close the dialog when Esc is pressed
      String cancelName = "cancel";
      InputMap inputMap =
getRootPane().getInputMap(JComponent.WHEN_ANCESTOR_OF_FOCUSED_COMPONENT);
inputMap.put(KeyStroke.getKeyStroke(KeyEvent.VK_ESCAPE, 0), cancelName);
      ActionMap actionMap = getRootPane().getActionMap();
actionMap.put(cancelName, new AbstractAction() {
public void actionPerformed(ActionEvent e) {
doClose(RET_CANCEL);
         }
      });
   }

   /**
    * @return the return status of this dialog - one of RET_OK or RET_CANCEL
    */
```

```
public int getReturnStatus() {
return returnStatus;
   }

   /**
    * This method is called from within the constructor to initialize the form.
    * WARNING: Do NOT modify this code. The content of this method is always
    * regenerated by the Form Editor.
    */
@SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

okButton = new javax.swing.JButton();
      jLabel1 = new javax.swing.JLabel();
newpid = new javax.swing.JLabel();

addWindowListener(new java.awt.event.WindowAdapter() {
public void windowClosing(java.awt.event.WindowEvent evt) {
closeDialog(evt);
         }
      });

okButton.setText("OK");
okButton.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
okButtonActionPerformed(evt);
         }
      });

jLabel1.setText("New Partida id:");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)
            .addGap(31, 31, 31)
            .addComponent(newpid)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap(26, Short.MAX_VALUE)
            .addComponent(okButton, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(98, 98, 98))
      );
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(20, 20, 20)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
               .addComponent(jLabel1)
               .addComponent(newpid))
            .addGap(33, 33, 33)
            .addComponent(okButton)
            .addContainerGap(25, Short.MAX_VALUE))
      );

getRootPane().setDefaultButton(okButton);

pack();
   }// </editor-fold>

private void okButtonActionPerformed(java.awt.event.ActionEvent evt) {
doClose(RET_OK);
   }

   /**
    * Closes the dialog
    */
private void closeDialog(java.awt.event.WindowEvent evt) {
doClose(RET_CANCEL);
   }

private void doClose(int retStatus) {
returnStatus = retStatus;
setVisible(false);
dispose();
   }

   /**
    * @param args the command line arguments
    */
public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */
try {
for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName());
break;
            }
          }
      } catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(NuevaPartidaDialog.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
      } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(NuevaPartidaDialog.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
      } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(NuevaPartidaDialog.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(NuevaPartidaDialog.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
      }
      //</editor-fold>

      /* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
            NuevaPartidaDialog dialog = new NuevaPartidaDialog(new javax.swing.JFrame(),
true,0);
dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
public void windowClosing(java.awt.event.WindowEvent e) {
System.exit(0);
                }
            });
dialog.setVisible(true);
}
      });
   }

   // Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel newpid;
private javax.swing.JButton okButton;
   // End of variables declaration

private int returnStatus = RET_CANCEL;
}
```

iii. CasillasInPartida.java

```
/*
 * 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 as.gui;

import as.entity.Casella;
import as.entity.Partidas;
import as.util.HibernateUtil;
import java.util.ArrayList;
import java.util.List;
import java.util.Vector;
import javax.swing.RowSorter;
import javax.swing.SortOrder;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;
import org.hibernate.HibernateException;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;

/**
 *
 * @author fjgarate
 */
public class CasillasInPartida extends javax.swing.JFrame {

    /**
     * Creates new form ventana
     */
public CasillasInPartida() {
initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
@SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
idPartidaText = new javax.swing.JTextField();
        jScrollPane1 = new javax.swing.JScrollPane();
resultTable = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("id Partida:");

resultTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
new String [] {
            "Title 1", "Title 2", "Title 3", "Title 4"
        }
    ));
jScrollPane1.setViewportView(resultTable);

jButton1.setText("Buscar");
jButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton1ActionPerformed(evt);
        }
    });

jLabel2.setText("Caselles trobades");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 697,
Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1)
                    .addGap(18, 18, 18)
                    .addComponent(idPartidaText, javax.swing.GroupLayout.PREFERRED_SIZE, 112,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jButton1)
                    .addGap(0, 0, Short.MAX_VALUE)))
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addGap(341, 341, 341)
            .addComponent(jLabel2)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(idPartidaText, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton1))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 15,
Short.MAX_VALUE)
            .addComponent(jLabel2)
            .addGap(18, 18, 18)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 265,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
      );

pack();
   }// </editor-fold>
private static String PARTIDAS_QUERY_BASED_ON_IDPARTIDA="from Partidas p where
p.idPartida = ";
private static String CASELLES_QUERY_BASED_ON_IDPARTIDA="from Casella c where
c.id.idPartida = ";

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
if (idPartidaText.getText() != "") {
executeHQLQuery(PARTIDAS_QUERY_BASED_ON_IDPARTIDA + idPartidaText.getText());
      }
   }

   /**
    * @param args the command line arguments
    */
public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */
try {
for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName());
break;
            }
          }
      } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(CasillasInPartida.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
      } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(CasillasInPartida.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(CasillasInPartida.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(CasillasInPartida.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new CasillasInPartida().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
private javax.swing.JTextField idPartidaText;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable resultTable;
    // End of variables declaration

private Partidas p;
private void executeHQLQuery(String hql) {
try {
            Session session = HibernateUtil.getSessionFactory().openSession();
session.beginTransaction();
            Query q = session.createQuery(hql);
            List resultList = q.list();
displayResult(resultList);
session.getTransaction().commit();
session.close();
        } catch (HibernateException he) {
he.printStackTrace();
        }
    }

private void displayResult(List resultList) {
        Vector<String> tableHeaders = new Vector<String>();
        Vector tableData = new Vector();
tableHeaders.add("Numero Fila");
tableHeaders.add("Numero Columna");
tableHeaders.add("Numero");

for(Object o : resultList) {
```

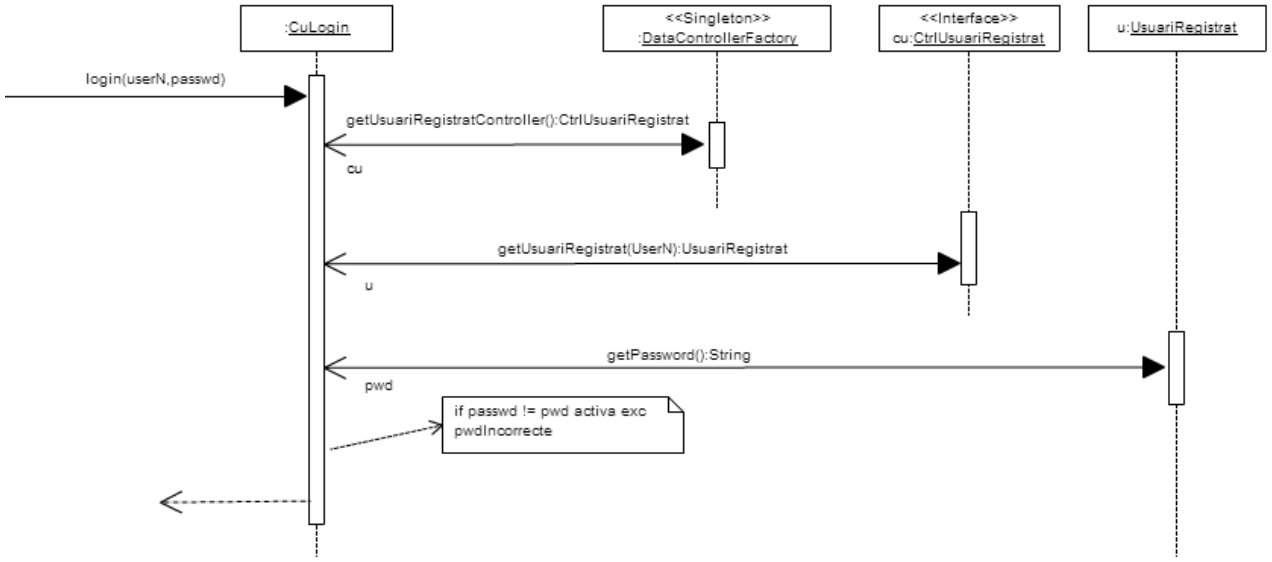
```
        p = (Partidas)o;
for(Object ocs : p.getCasellas()) {
            Casella cas = (Casella)ocs;
            Vector<Object> oneRow = new Vector<Object>();
oneRow.add(cas.getId().getNumeroFila());
oneRow.add(cas.getId().getNumeroColumna());
oneRow.add(cas.getNumero());

tableData.add(oneRow);
        }
      }

      DefaultTableModel model = new DefaultTableModel(tableData, tableHeaders)
      {
public boolean isCellEditable(int row, int column)
        {
if (column != 2) return false;
return true;
        }
        @Override
public void setValueAt(Object value, int row, int col) {
            ArrayList<Casella> cs = new ArrayList<Casella>(p.getCasellas());
int v = Integer.parseInt((String)value);
int numfila = (int)this.getValueAt(row, 0);
int numcolu = (int)this.getValueAt(row, 1);

for (Casella c : cs){
if (numfila == c.getId().getNumeroFila() && numcolu == c.getId().getNumeroColumna()) {

c.setNumero(v);
                Session session = HibernateUtil.getSessionFactory().openSession();
try {

                   Transaction tx = session.beginTransaction();

                   session.update(c);

tx.commit();

                }

catch (HibernateException e) {

e.printStackTrace();

session.getTransaction().rollback();

                } finally {
session.close();
//jButton1ActionPerformed(null);
                }
```

```
break;
                }
             }
fireTableCellUpdated(row, col);

jButton1ActionPerformed(null);
          }

            // return true;

       };
resultTable.setModel(model);
       TableRowSorter<TableModel> sorter = new TableRowSorter<>(resultTable.getModel());
resultTable.setRowSorter(sorter);
       List<RowSorter.SortKey> sortKeys = new ArrayList<>();

int columnIndexToSort = 0;
sortKeys.add(new RowSorter.SortKey(columnIndexToSort, SortOrder.ASCENDING));
sortKeys.add(new RowSorter.SortKey(columnIndexToSort+1, SortOrder.ASCENDING));

sorter.setSortKeys(sortKeys);
sorter.sort();
}
}
```

b. Ejercicio 2

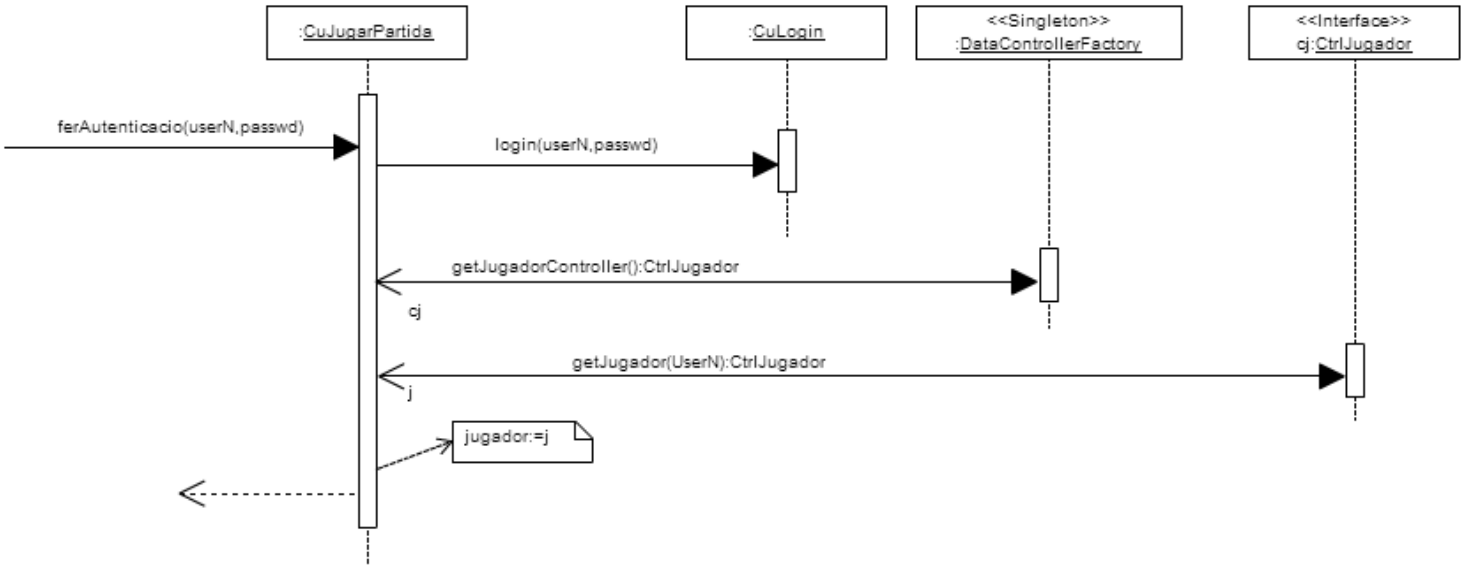
a. Caso de uso Login

i. Contrato login

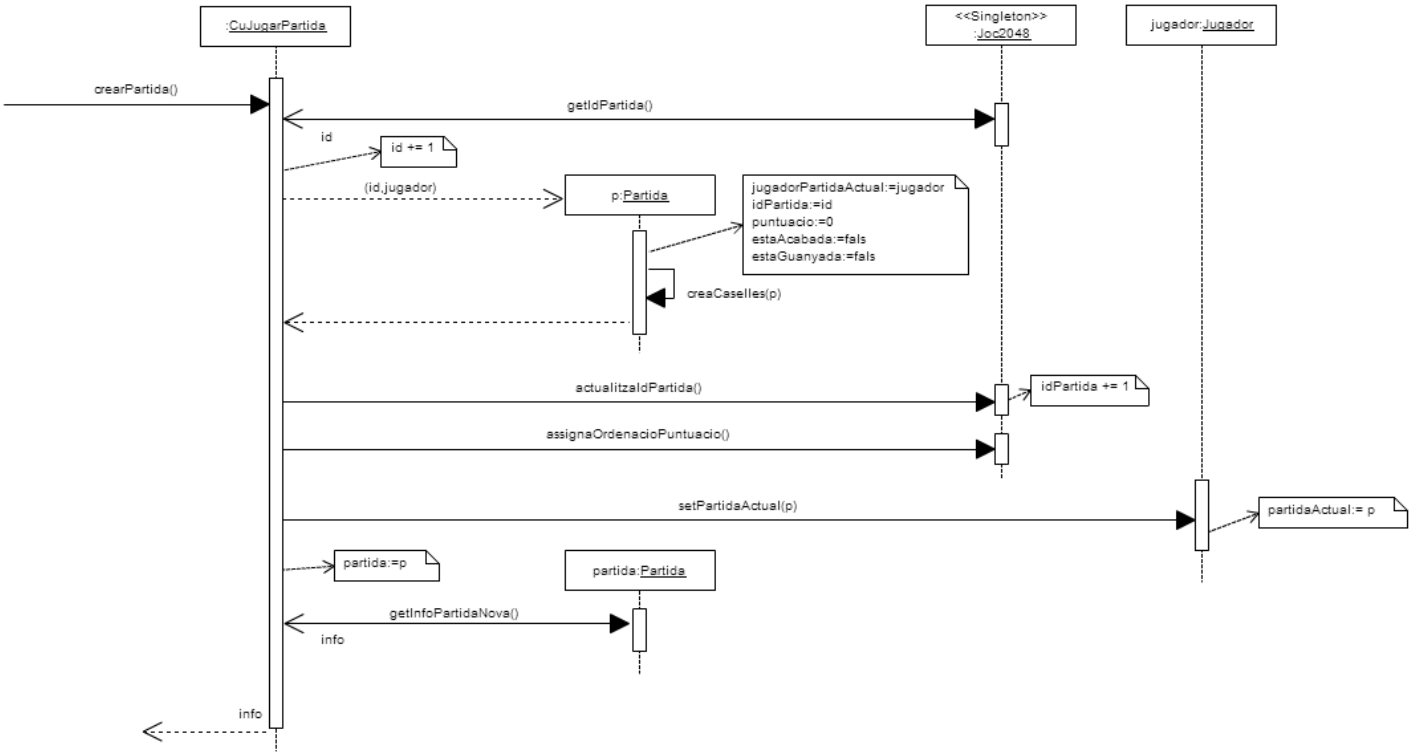


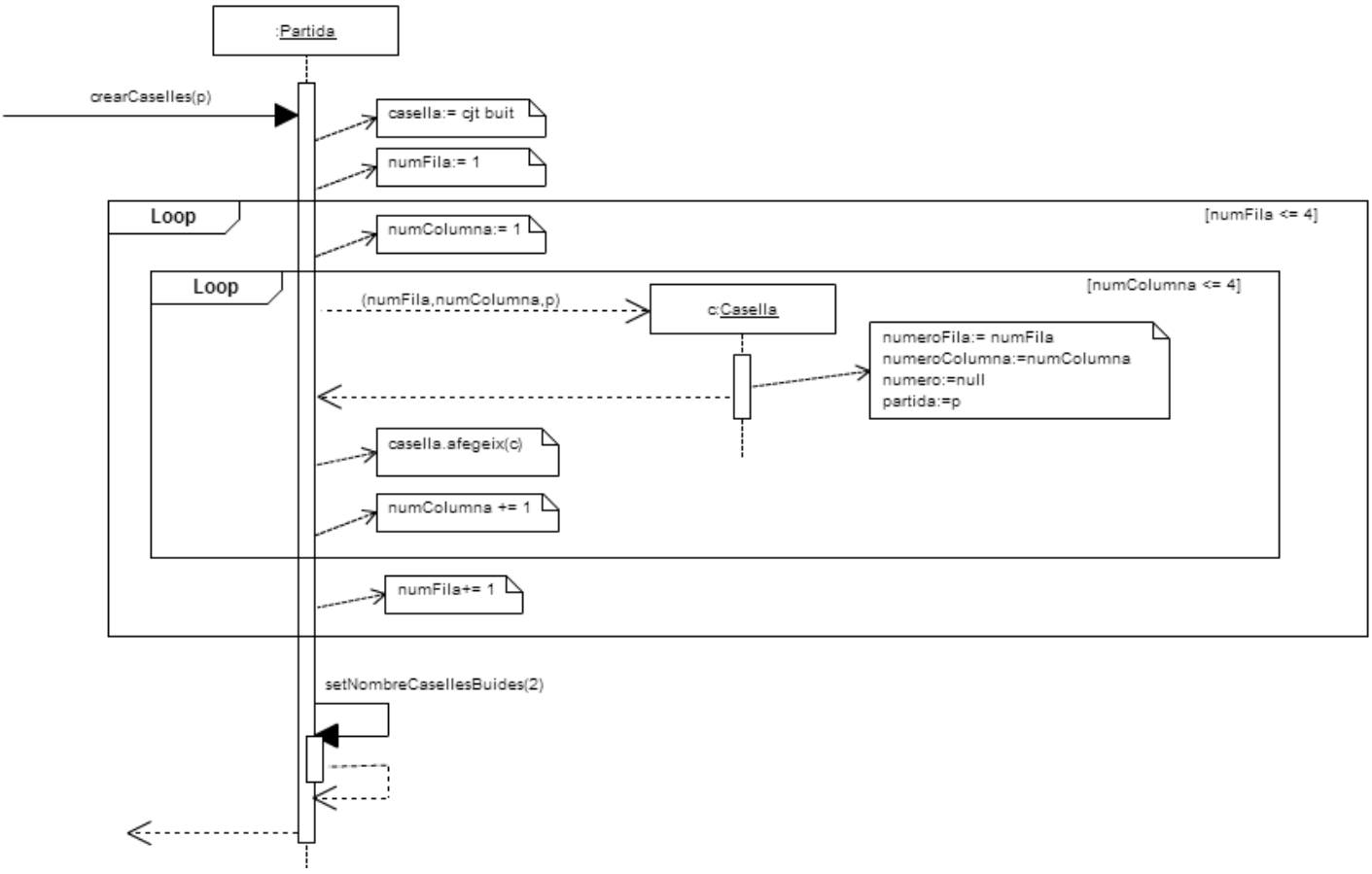
b. Caso de uso Jugar partida

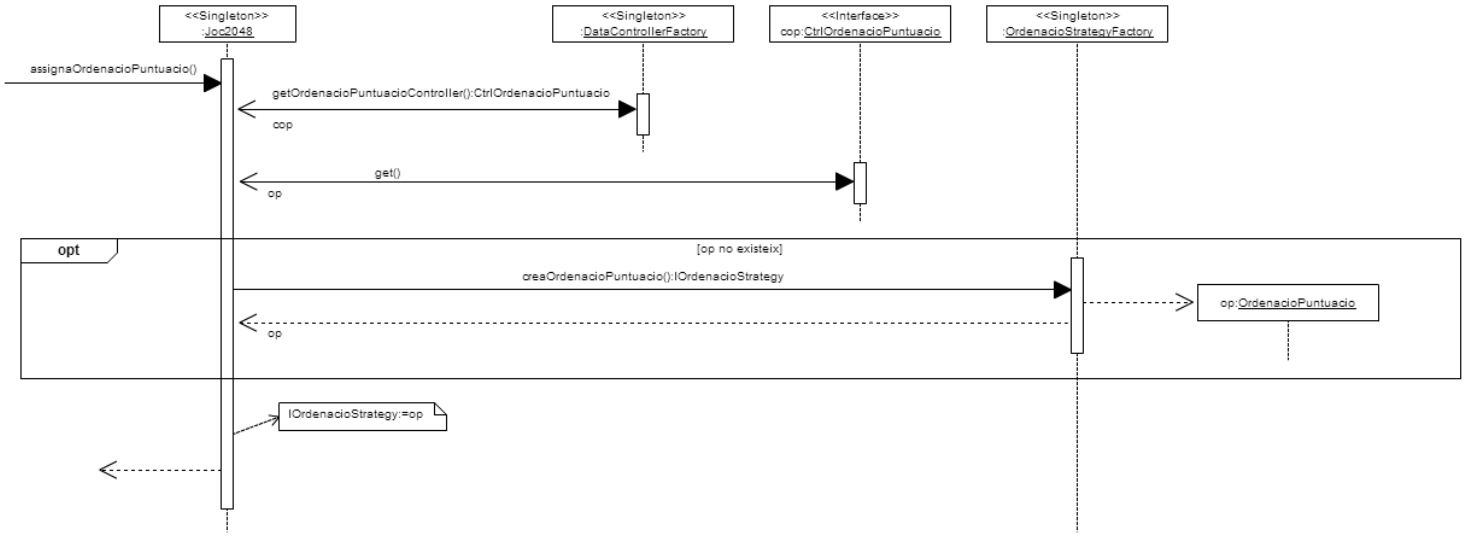
i. Contrato FerAutenticacio

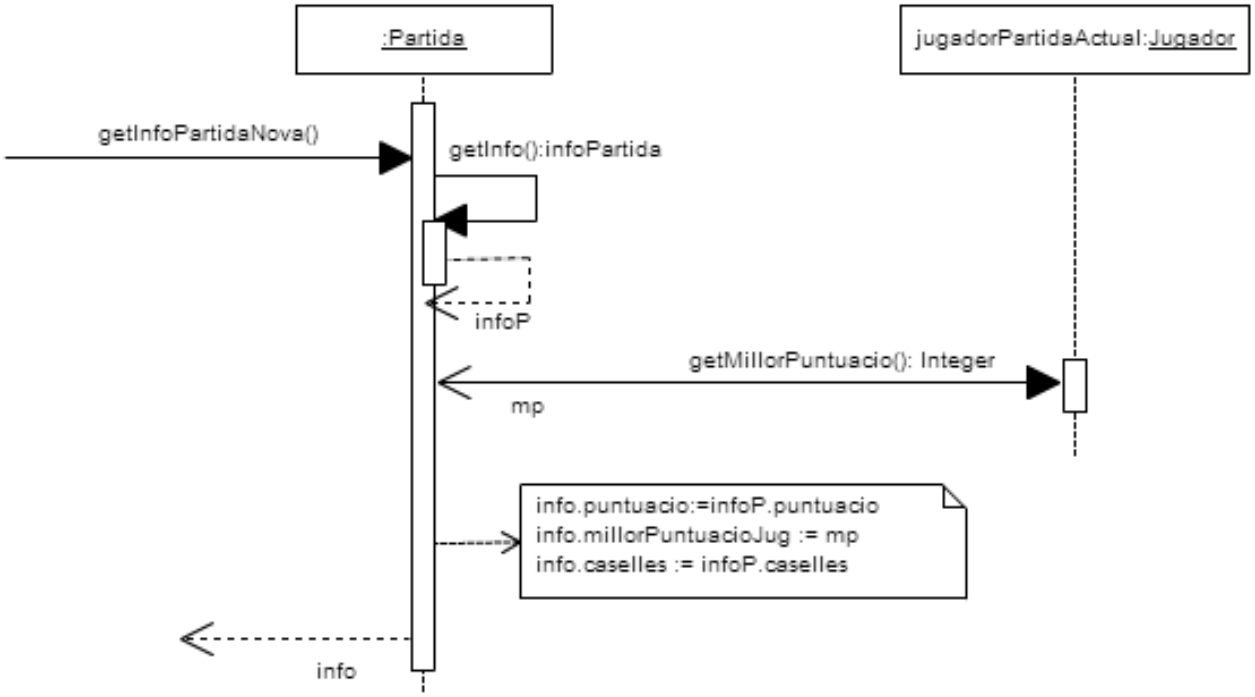


ii. Contrato CrearPartida

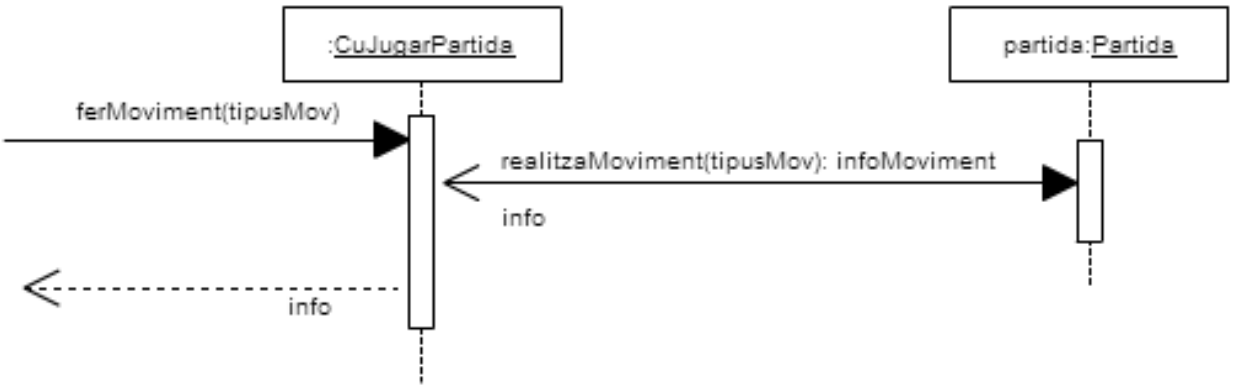


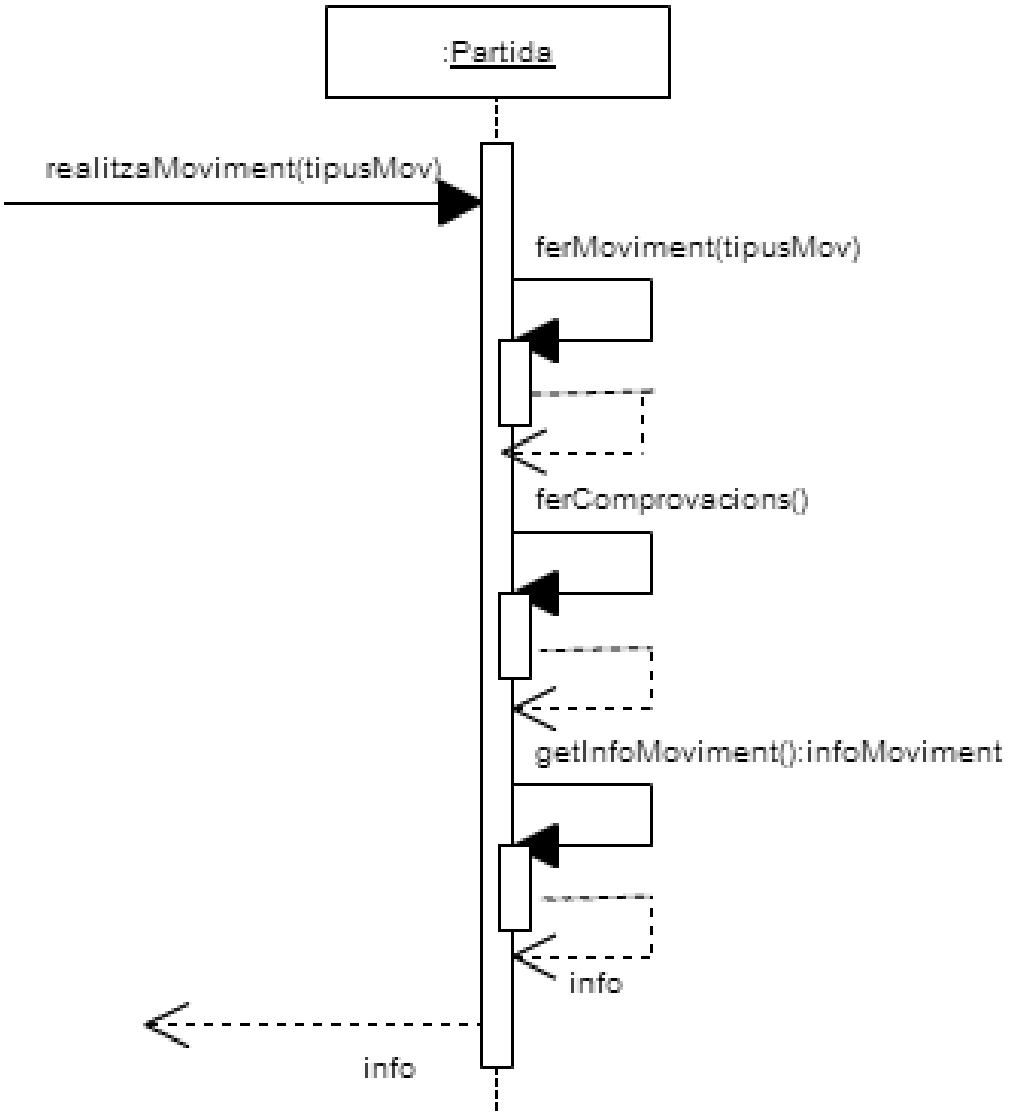


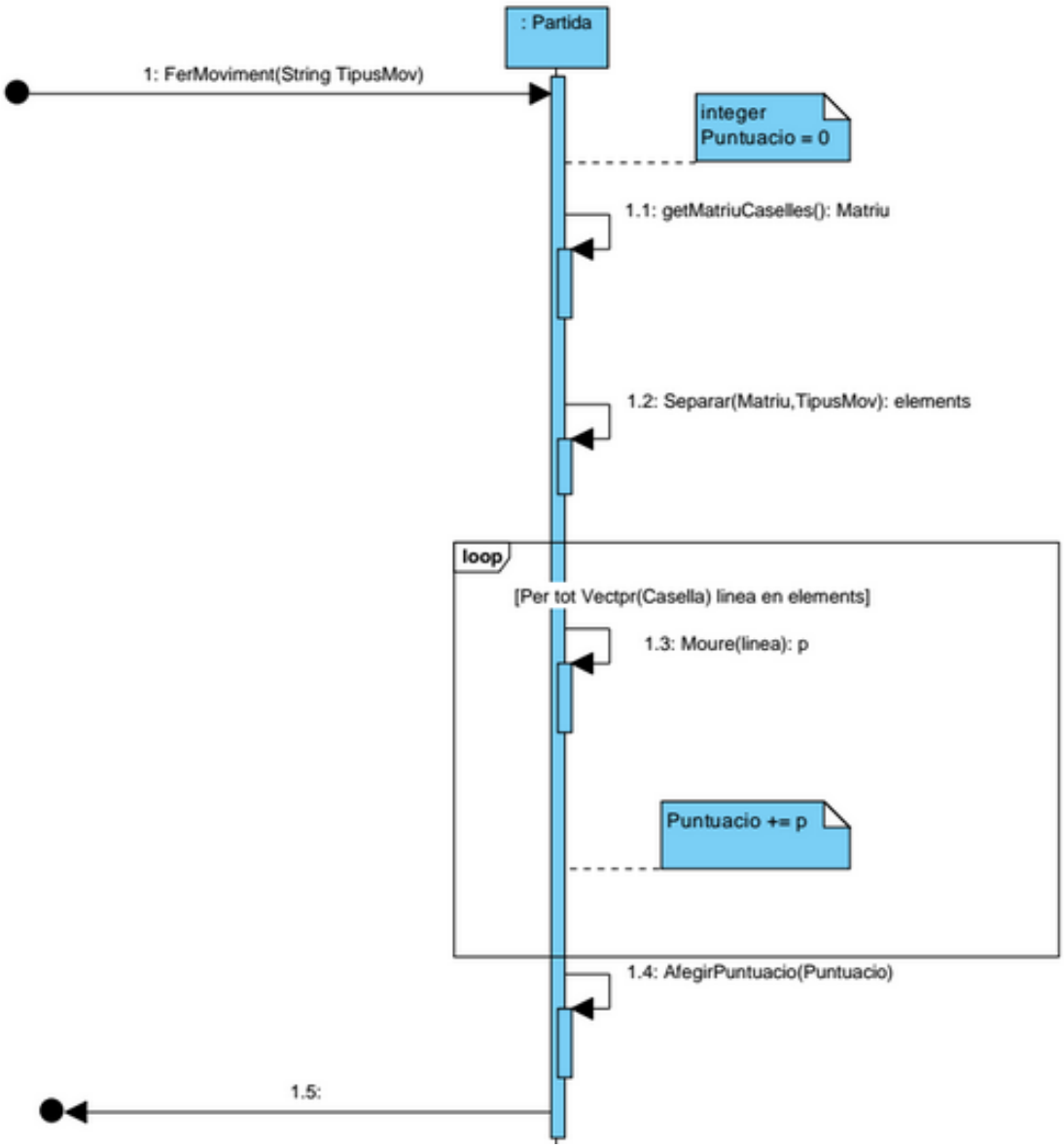

<<Singleton>>
:Joc2048
<<Singleton>>
:DataControllerFactory
<<Interface>>
cop:CtrlOrdenacioPuntuacio
<<Singleton>>
:OrdenacioStrategyFactory
assignaOrdenacioPuntuacio()
getOrdenacioPuntuacioController():CtrlOrdenacioPuntuacio
cop
get()
op
opt
[op no existeix]
creaOrdenacioPuntuacio():IOrdenacioStrategy
op:OrdenacioPuntuacio
op
IOrdenacioStrategy:=op

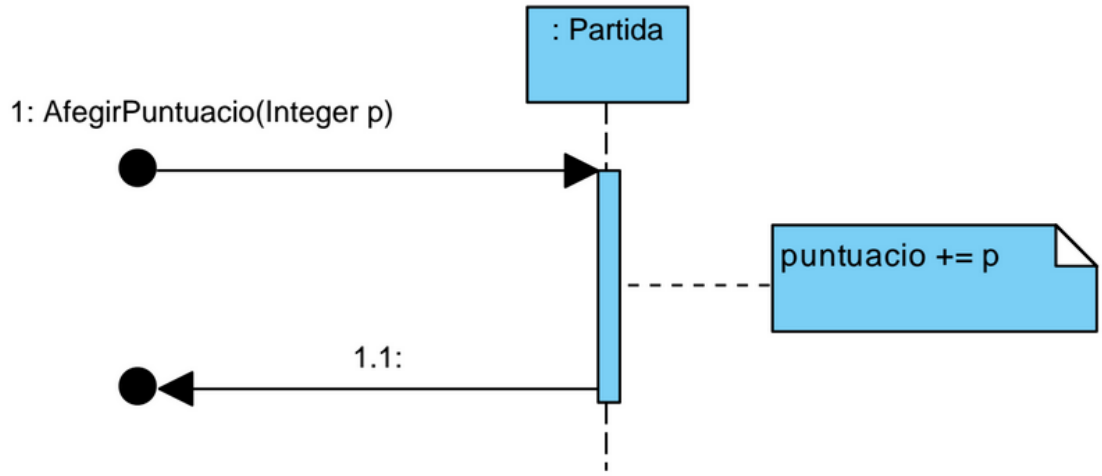

:Partida
jugadorPartidaActual:Jugador
getInfoPartidaNova()
getInfo():infoPartida
infoP
getMillorPuntuacio(): Integer
mp
info.puntuacio:=infoP.puntuacio
info.millorPuntuacioJug := mp
info.caselles := infoP.caselles
info

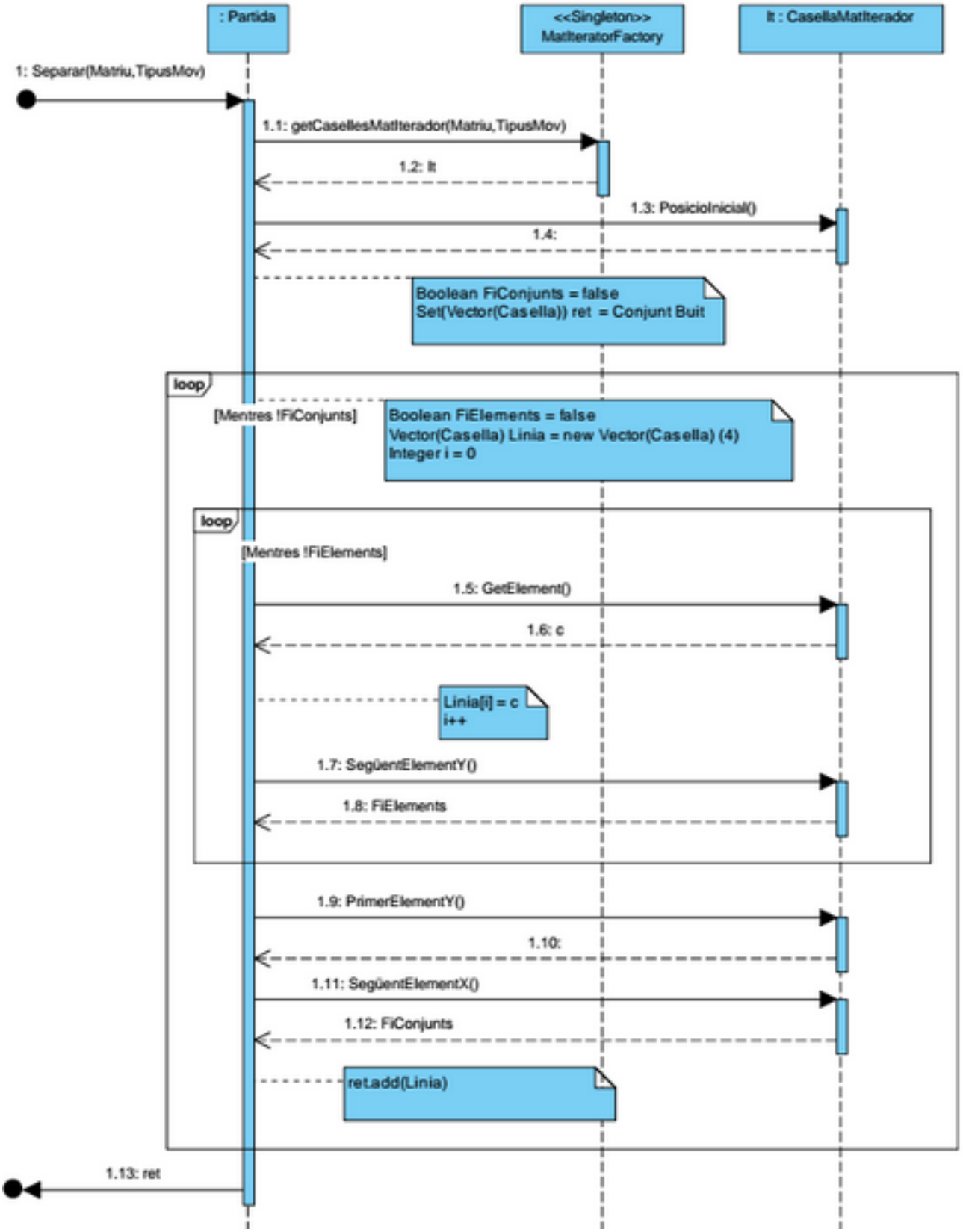
iii. Contrato FerMoviment

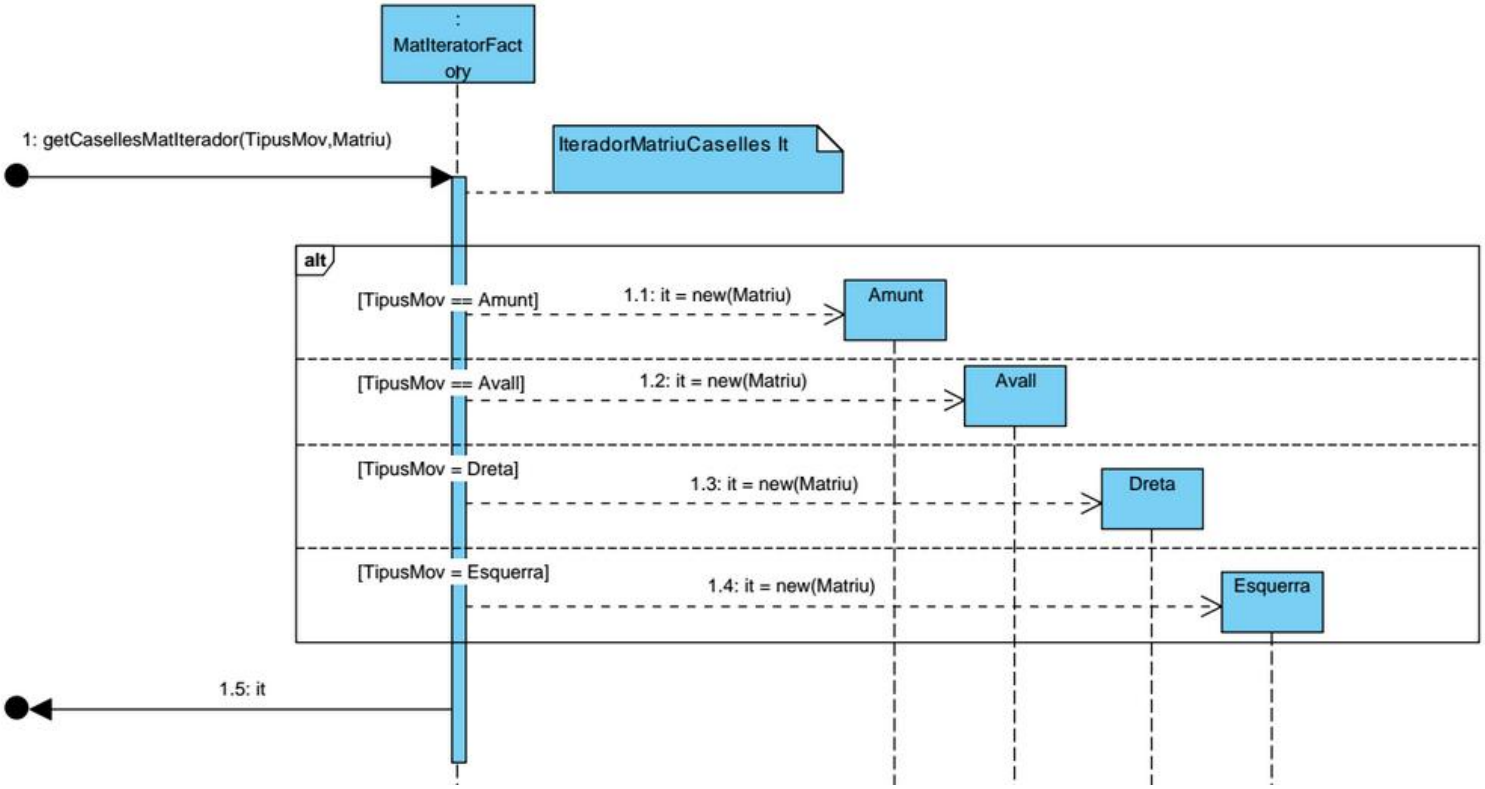


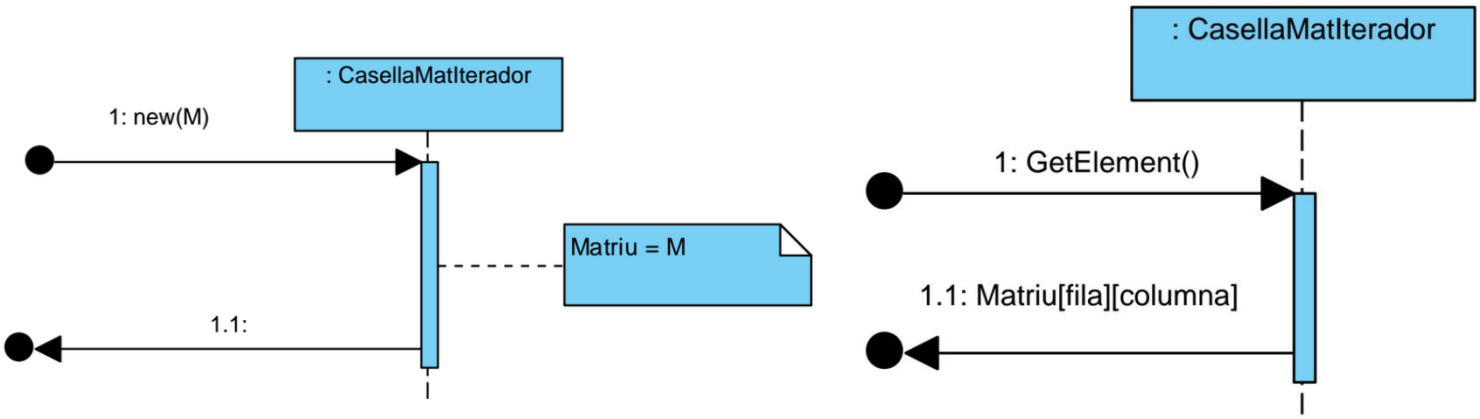


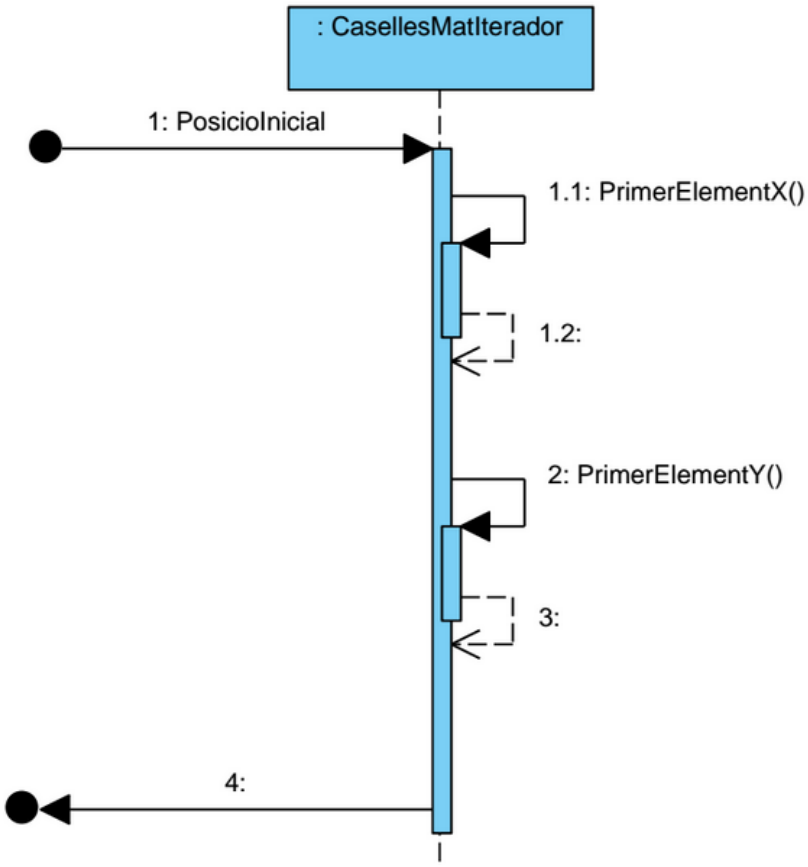

: Partida
1: FerMoviment(String TipusMov)
integer
Puntuacio = 0
1.1: getMatriuCaselles(): Matriu
1.2: Separar(Matriu,TipusMov): elements
loop
[Per tot Vectpr(Casella) linea en elements]
1.3: Moure(linea): p
Puntuacio += p
1.4: AfegirPuntuacio(Puntuacio)
1.5:

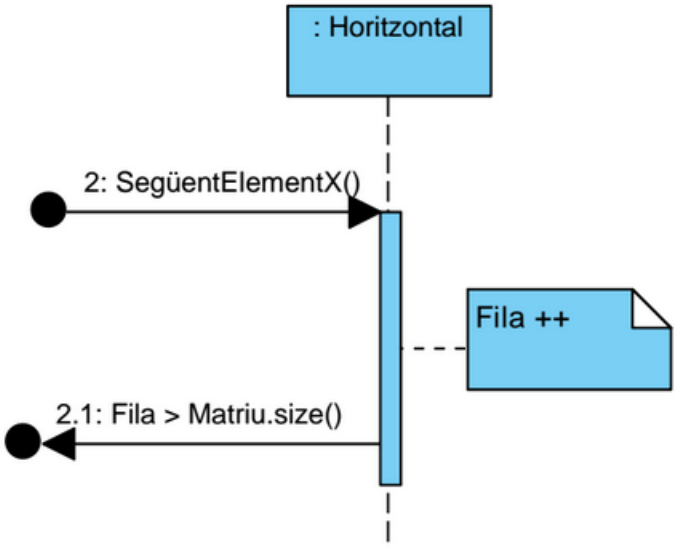

: Partida
1: AfegirPuntuacio(Integer p)
puntuacio += p
1.1:

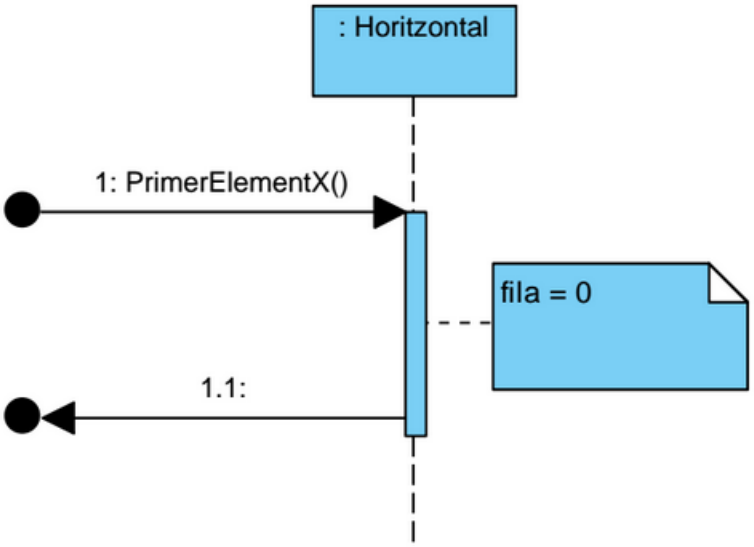

: Partida
<<Singleton>>
MatIteratorFactory
It : CasellaMatIterador
1: Separar(Matriu,TipusMov)
1.1: getCasellesMatIterador(Matriu,TipusMov)
1.2: It
1.3: PosicioInicial()
1.4:
Boolean FiConjunts = false
Set(Vector(Casella)) ret = Conjunt Buit
loop
[Mentres !FiConjunts]
Boolean FiElements = false
Vector(Casella) Linia = new Vector(Casella) (4)
Integer i = 0
loop
[Mentres !FiElements]
1.5: GetElement()
1.6: c
Linia[i] = c
i++
1.7: SegüentElementY()
1.8: FiElements
1.9: PrimerElementY()
1.10:
1.11: SegüentElementX()
1.12: FiConjunts
ret.add(Linia)
1.13: ret

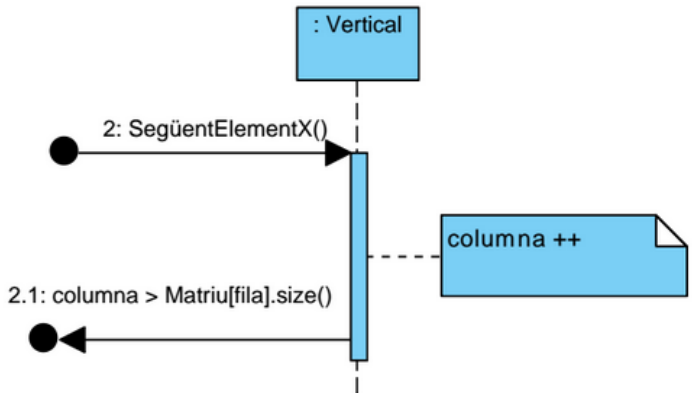

: MatIteratorFactory
1: getCasellesMatIterador(TipusMov,Matriu)
IteradorMatriuCaselles It
alt
[TipusMov == Amunt]
1.1: it = new(Matriu)
Amunt
[TipusMov == Avall]
1.2: it = new(Matriu)
Avall
[TipusMov = Dreta]
1.3: it = new(Matriu)
Dreta
[TipusMov = Esquerra]
1.4: it = new(Matriu)
Esquerra
1.5: it

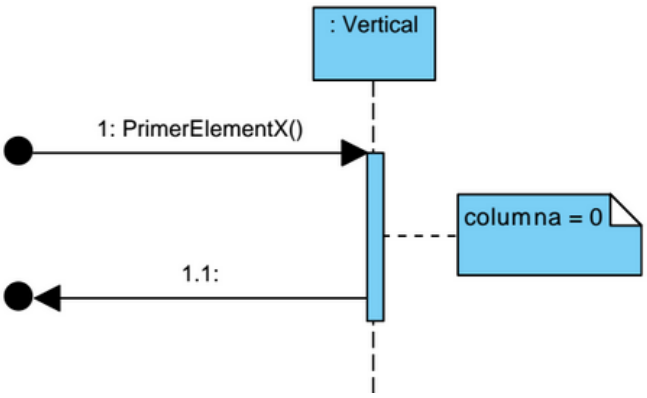

: CasellaMatIterador
1: new(M)
Matriu = M
1.1:
: CasellaMatIterador
1: GetElement()
1.1: Matriu[fila][columna]

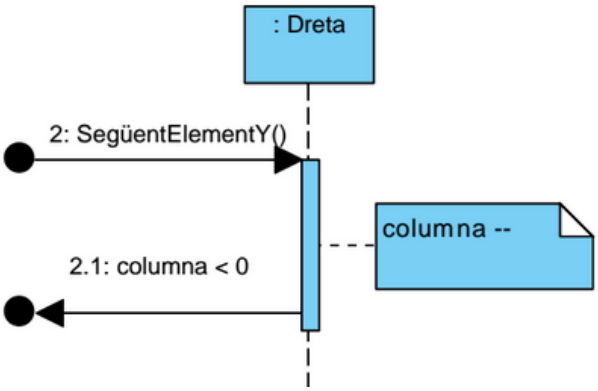

: CasellesMatIterador
1: PosicioInicial
1.1: PrimerElementX()
1.2:
2: PrimerElementY()
3:
4:

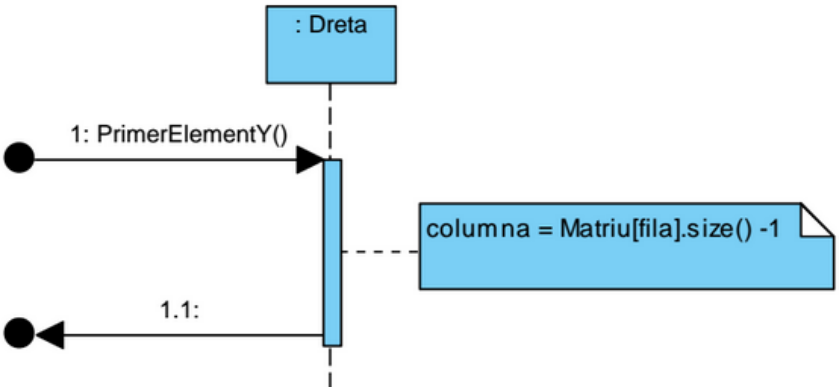

: Horitzontal
2: SegüentElementX()
Fila ++
2.1: Fila > Matriu.size()

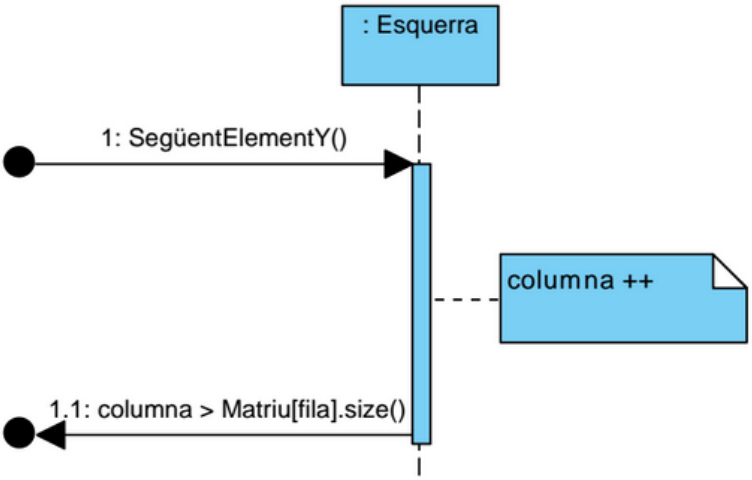

: Horitzontal
1: PrimerElementX()
fila = 0
1.1:

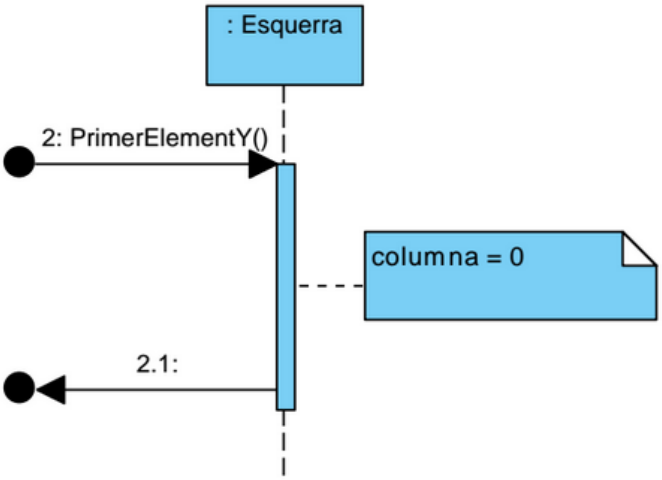

: Vertical
2: SegüentElementX()
columna ++
2.1: columna > Matriu[fila].size()

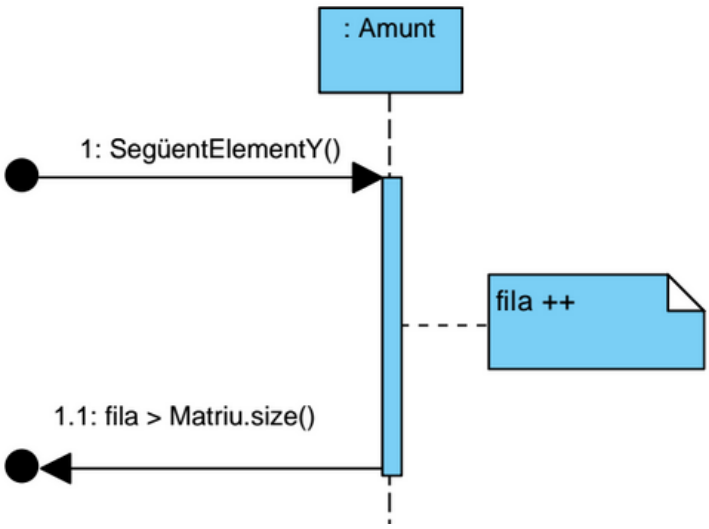

: Vertical
1: PrimerElementX()
columna = 0
1.1:

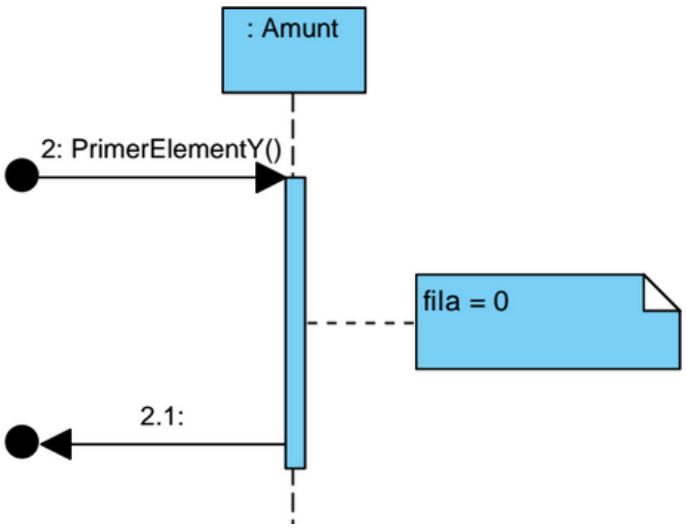

: Dreta
2: SegüentElementY()
columna --
2.1: columna < 0

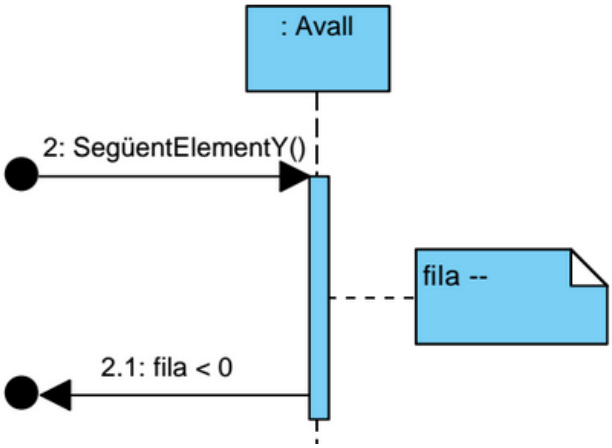

: Dreta
1: PrimerElementY()
columna = Matriu[fila].size() -1
1.1:

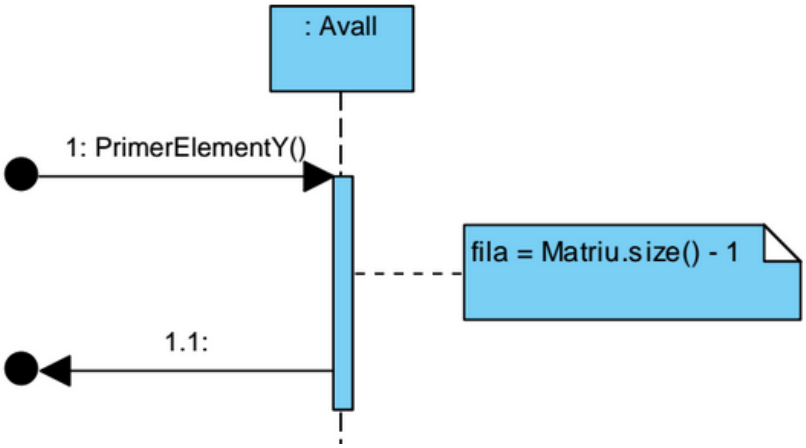

: Esquerra
1: SegüentElementY()
columna ++
1.1: columna > Matriu[fila].size()

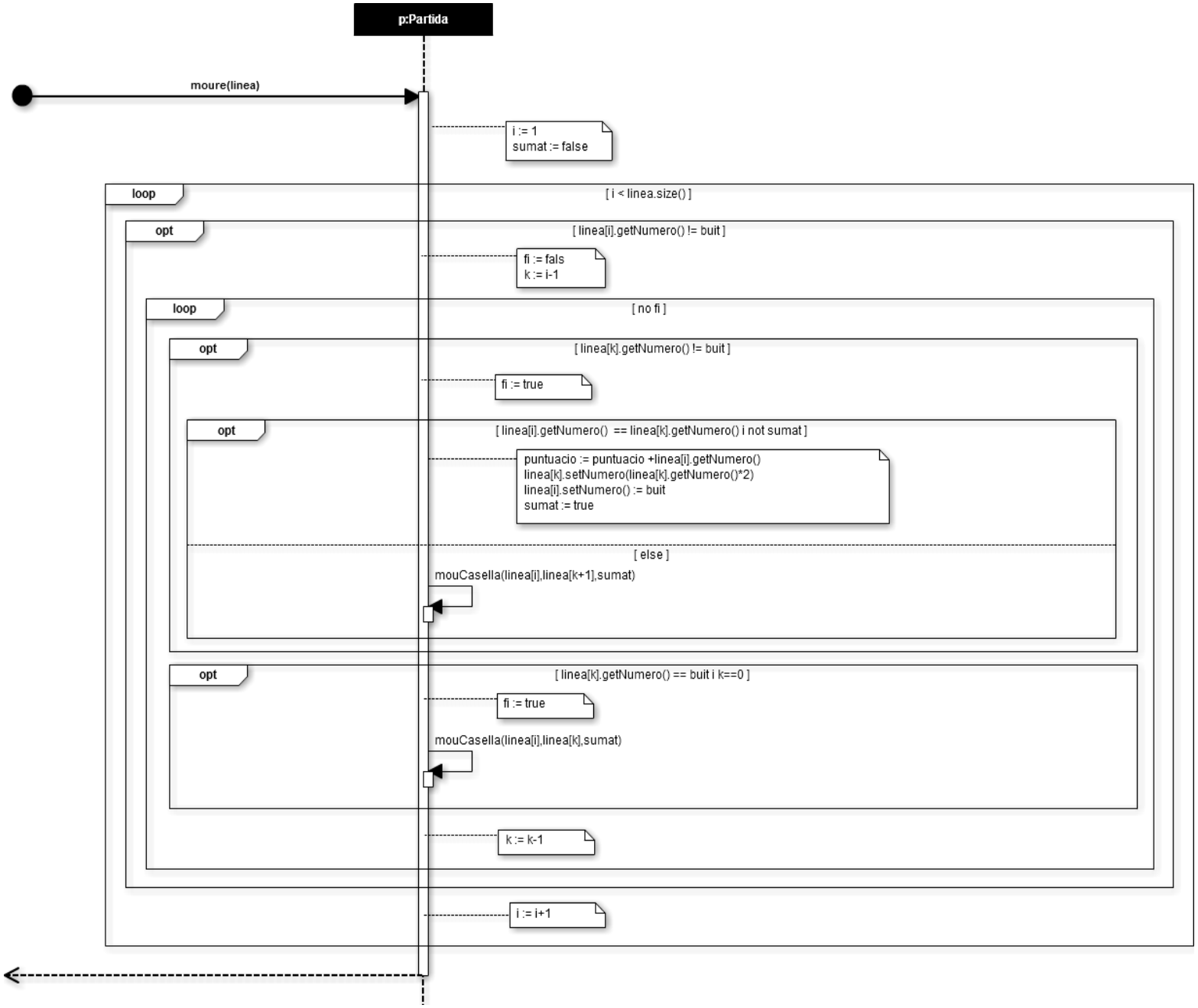

: Esquerra
2: PrimerElementY()
columna = 0
2.1:

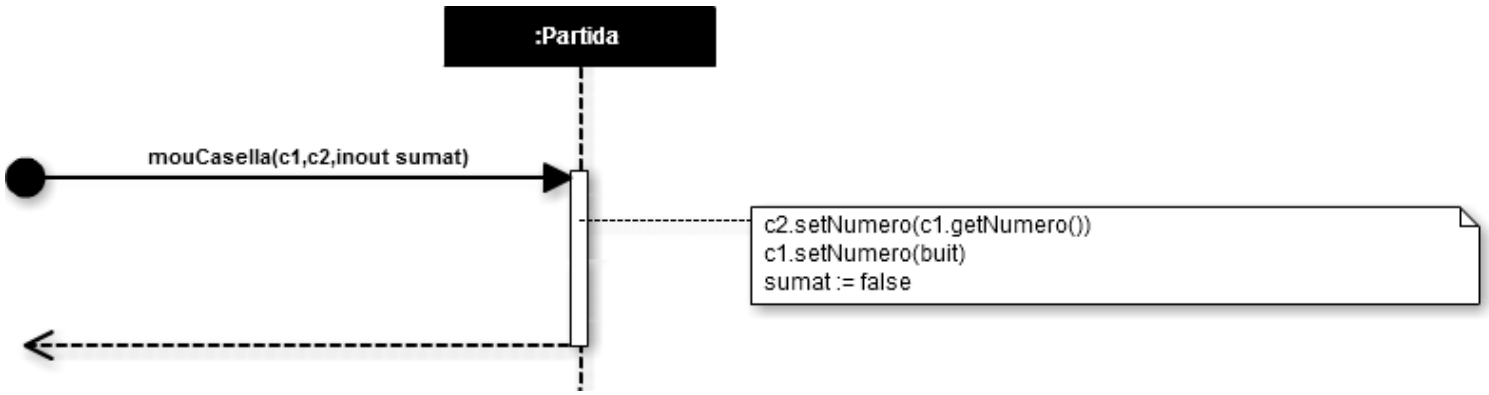

: Amunt
1: SegüentElementY()
fila ++
1.1: fila > Matriu.size()

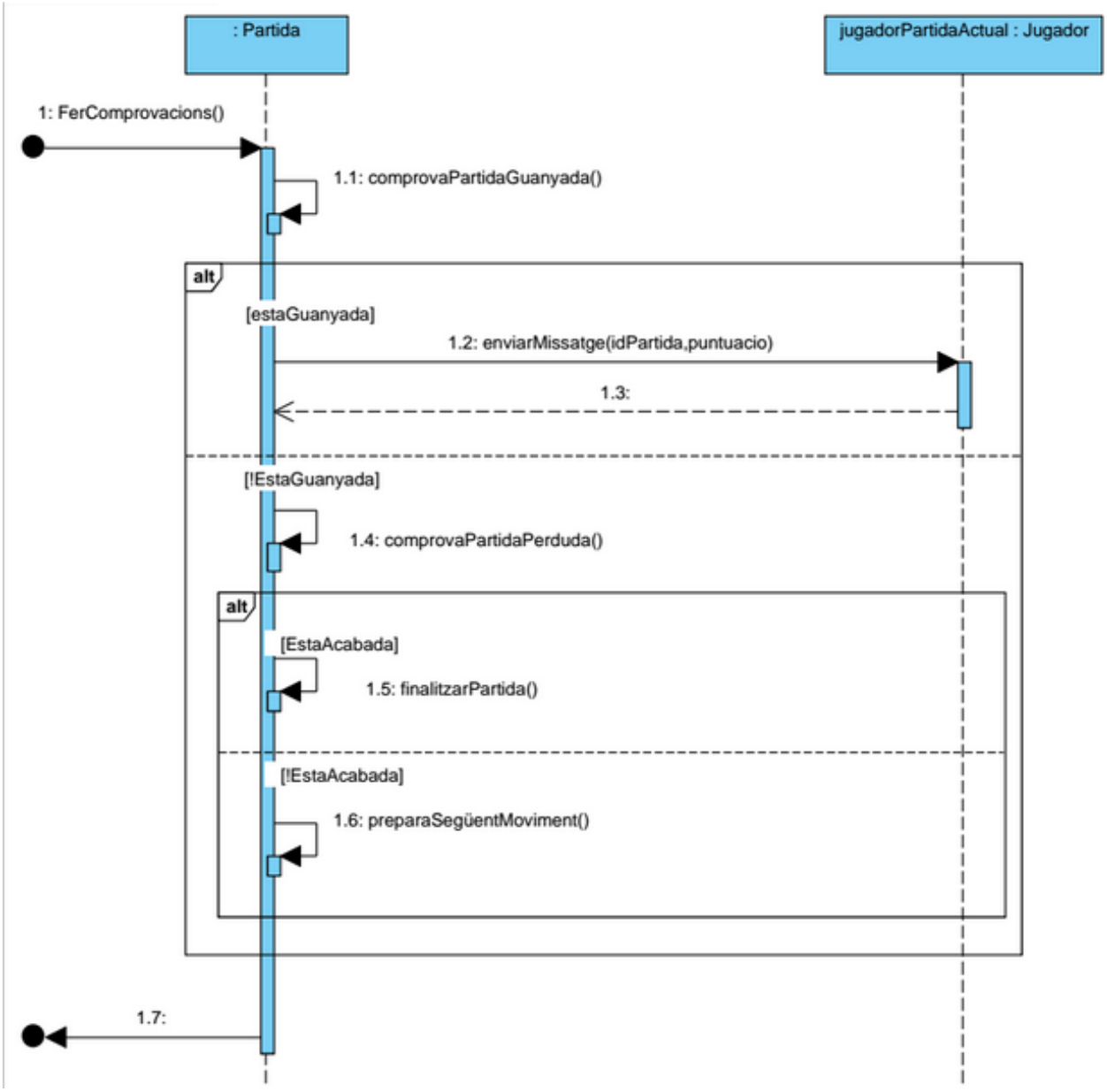

: Amunt
2: PrimerElementY()
fila = 0
2.1:

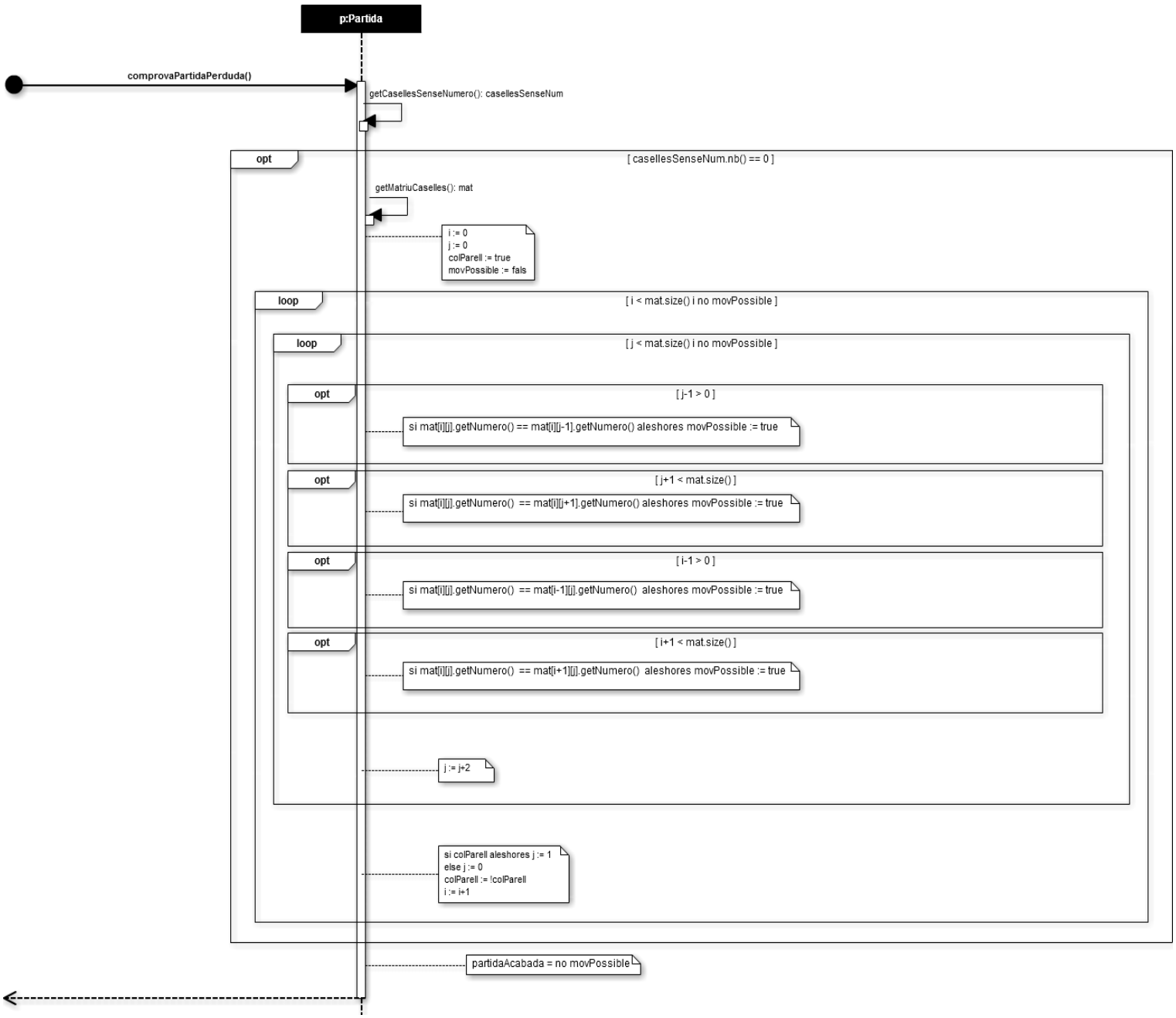

: Avall
2: SegüentElementY()
fila --
2.1: fila < 0

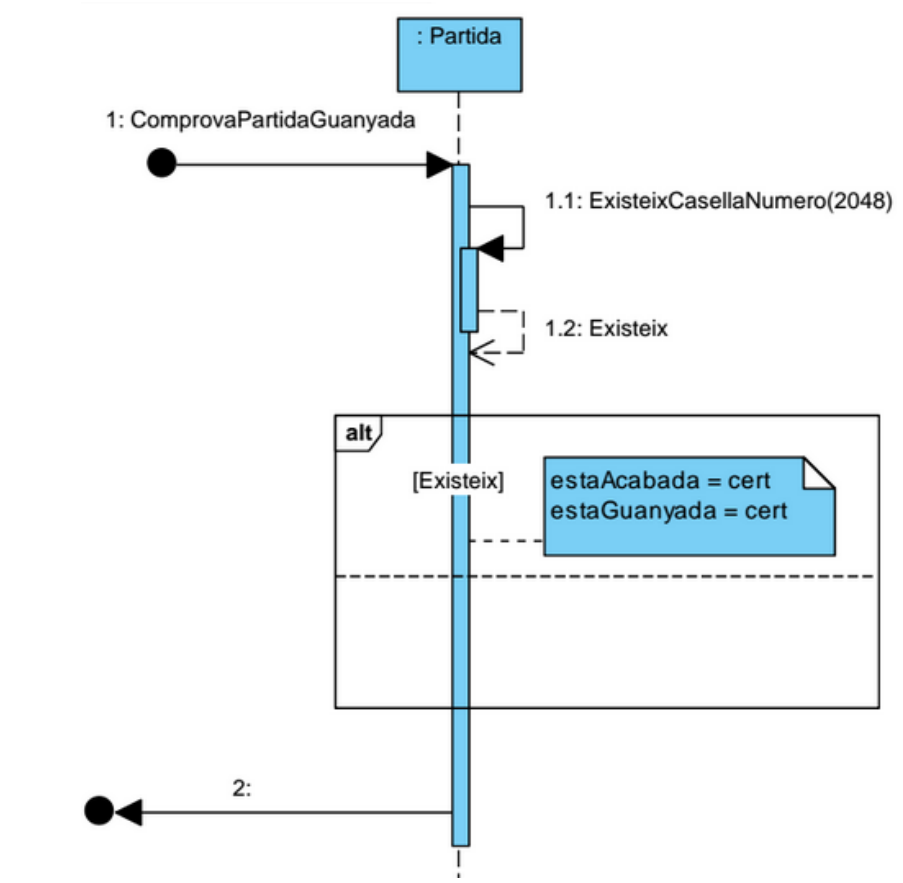

: Avall
1: PrimerElementY()
fila = Matriu.size() - 1
1.1:

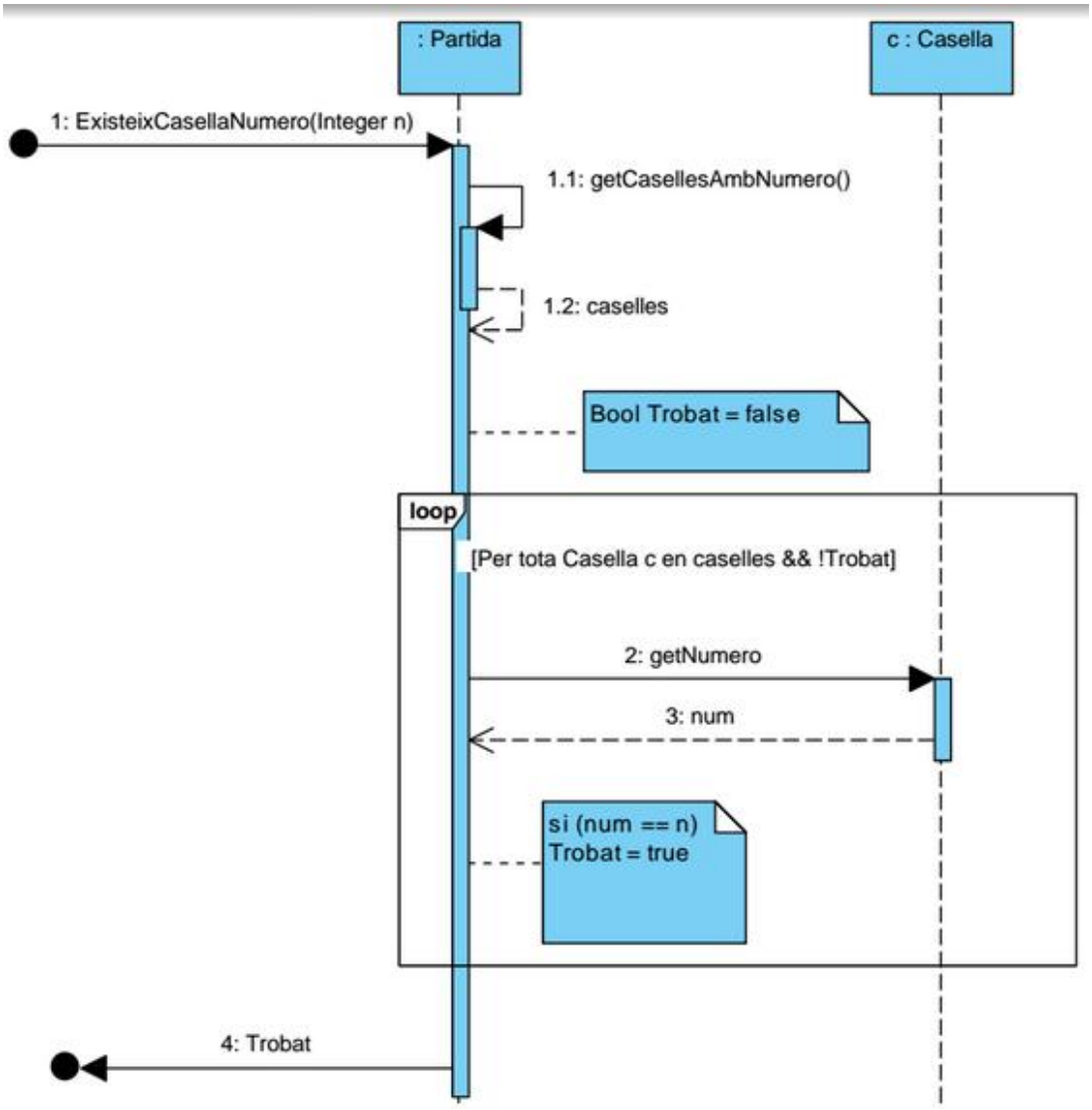

p:Partida
moure(linea)
i := 1
sumat := false
loop
[i < linea.size()]
opt
[linea[i].getNumero() != buit]
fi := fals
k := i-1
loop
[no fi]
opt
[linea[k].getNumero() != buit]
fi := true
opt
[linea[i].getNumero() == linea[k].getNumero() i not sumat]
puntuacio := puntuacio +linea[i].getNumero()
linea[k].setNumero(linea[k].getNumero()*2)
linea[i].setNumero() := buit
sumat := true
[else]
mouCasella(linea[i],linea[k+1],sumat)
opt
[linea[k].getNumero() == buit i k==0]
fi := true
mouCasella(linea[i],linea[k],sumat)
k := k-1
i := i+1

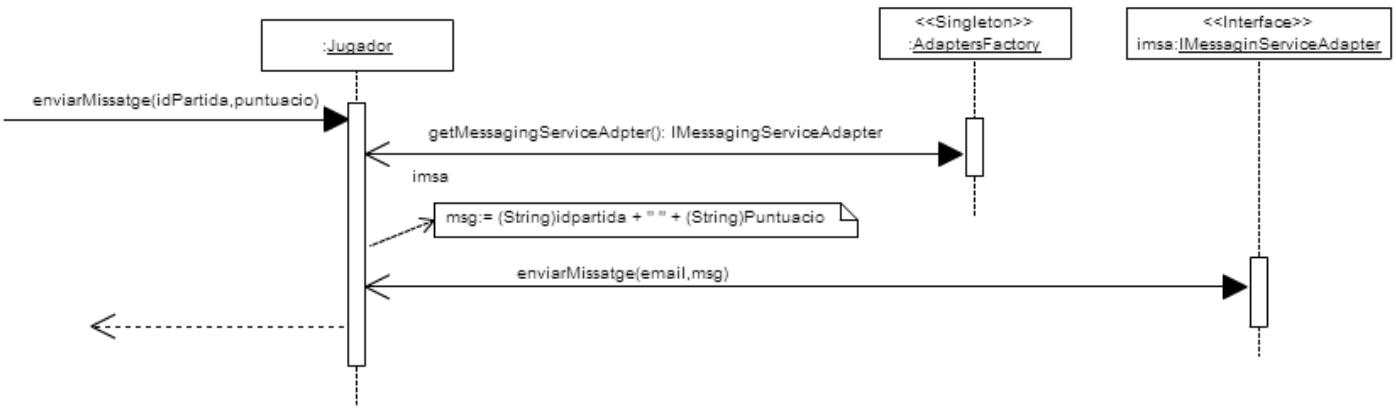

:Partida
mouCasella(c1,c2,inout sumat)
c2.setNumero(c1.getNumero())
c1.setNumero(buit)
sumat := false

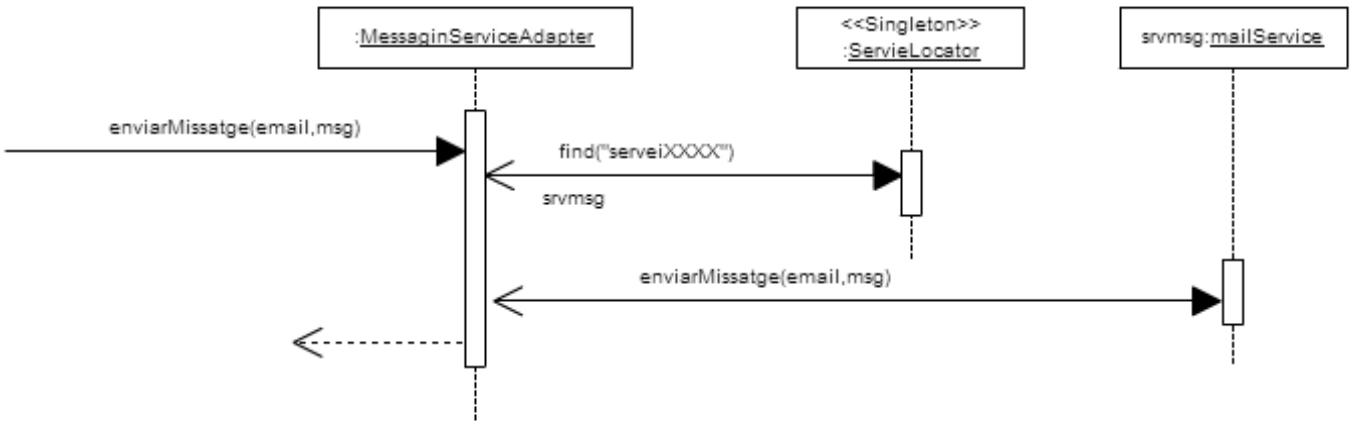

: Partida
jugadorPartidaActual : Jugador
1: FerComprovacions()
1.1: comprovaPartidaGuanyada()
alt
[estaGuanyada]
1.2: enviarMissatge(idPartida,puntuacio)
1.3:
[!EstaGuanyada]
1.4: comprovaPartidaPerduda()
alt
[EstaAcabada]
1.5: finalitzarPartida()
[!EstaAcabada]
1.6: preparaSegüentMoviment()
1.7:

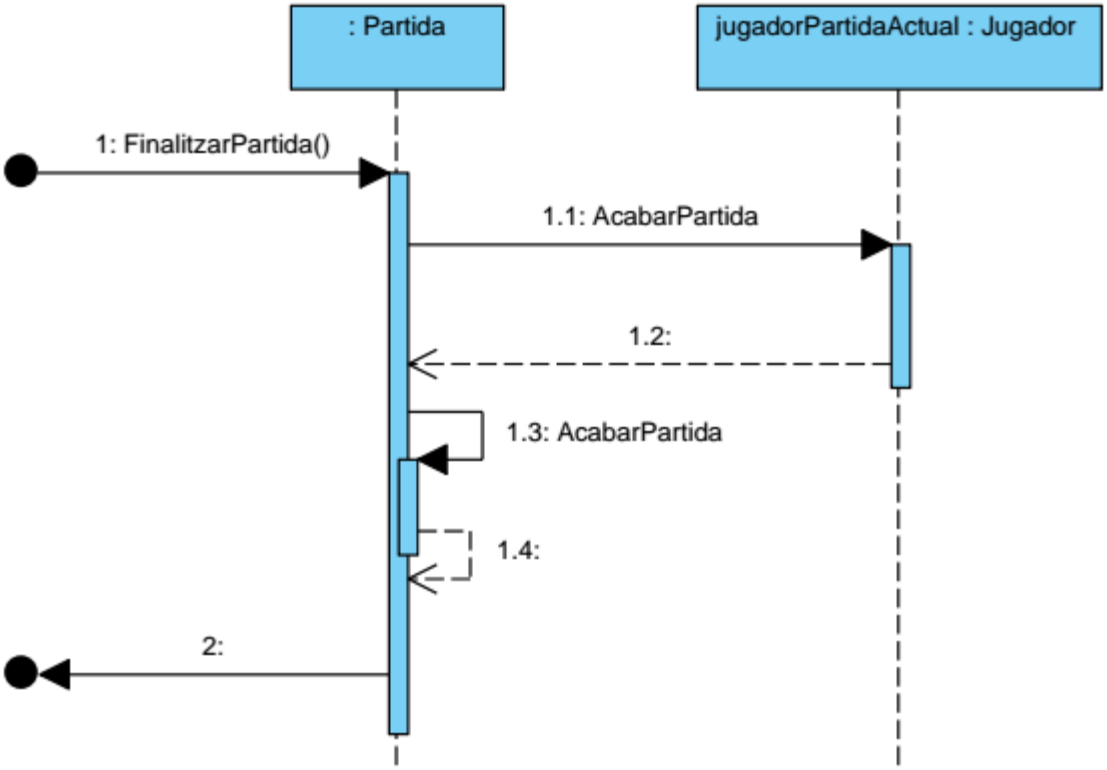

p:Partida
comprovaPartidaPerduda()
getCasellesSenseNumero(): casellesSenseNum
opt
[casellesSenseNum.nb() == 0]
getMatriuCaselles(): mat
i := 0
j := 0
colParell := true
movPossible := fals
loop
[i < mat.size() i no movPossible]
loop
[j < mat.size() i no movPossible]
opt
[j-1 > 0]
si mat[i][j].getNumero() == mat[i][j-1].getNumero() aleshores movPossible := true
opt
[j+1 < mat.size()]
si mat[i][j].getNumero() == mat[i][j+1].getNumero() aleshores movPossible := true
opt
[i-1 > 0]
si mat[i][j].getNumero() == mat[i-1][j].getNumero() aleshores movPossible := true
opt
[i+1 < mat.size()]
si mat[i][j].getNumero() == mat[i+1][j].getNumero() aleshores movPossible := true
j := j+2
si colParell aleshores j := 1
else j := 0
colParell := !colParell
i := i+1
partidaAcabada = no movPossible

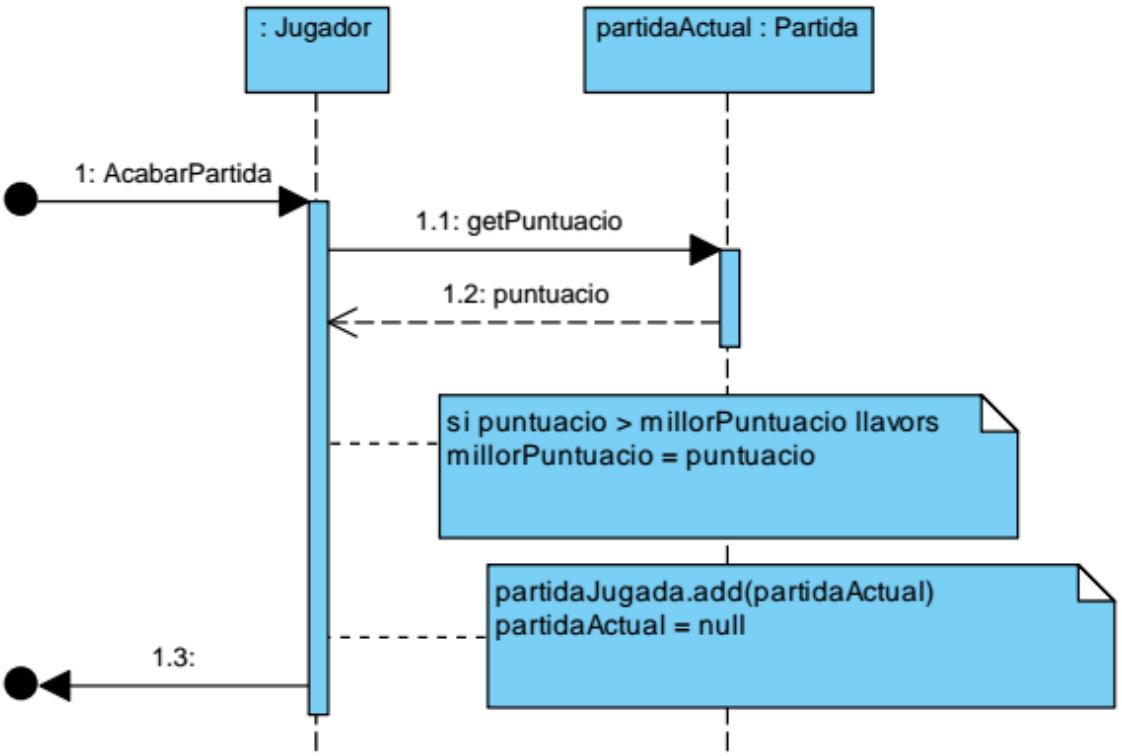

: Partida
1: ComprovaPartidaGuanyada
1.1: ExisteixCasellaNumero(2048)
1.2: Existeix
alt
[Existeix]
estaAcabada = cert
estaGuanyada = cert
2:

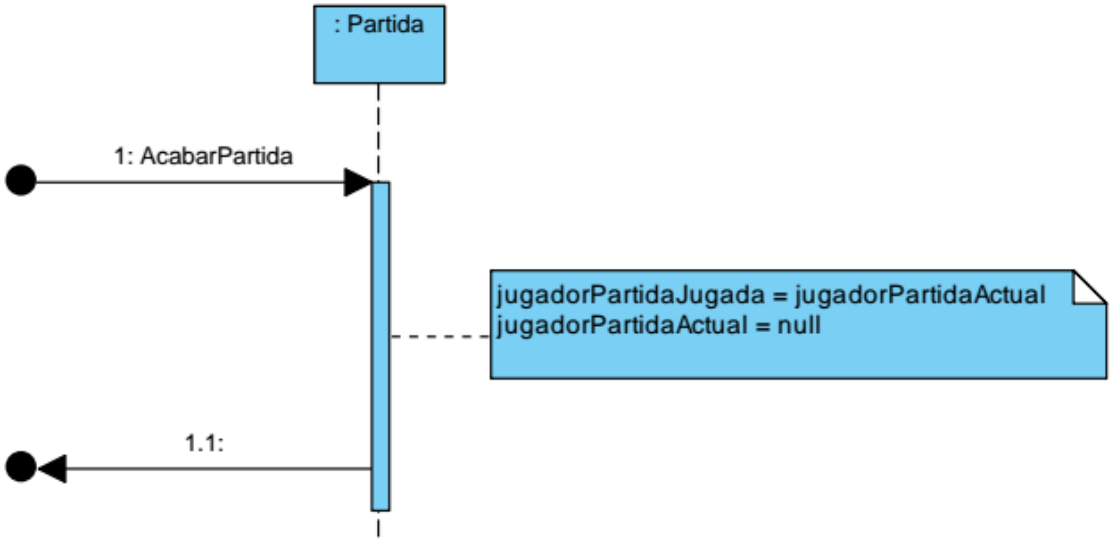

: Partida
c : Casella
1: ExisteixCasellaNumero(Integer n)
1.1: getCasellesAmbNumero()
1.2: caselles
Bool Trobat = false
loop
[Per tota Casella c en caselles && !Trobat]
2: getNumero
3: num
si (num == n)
Trobat = true
4: Trobat

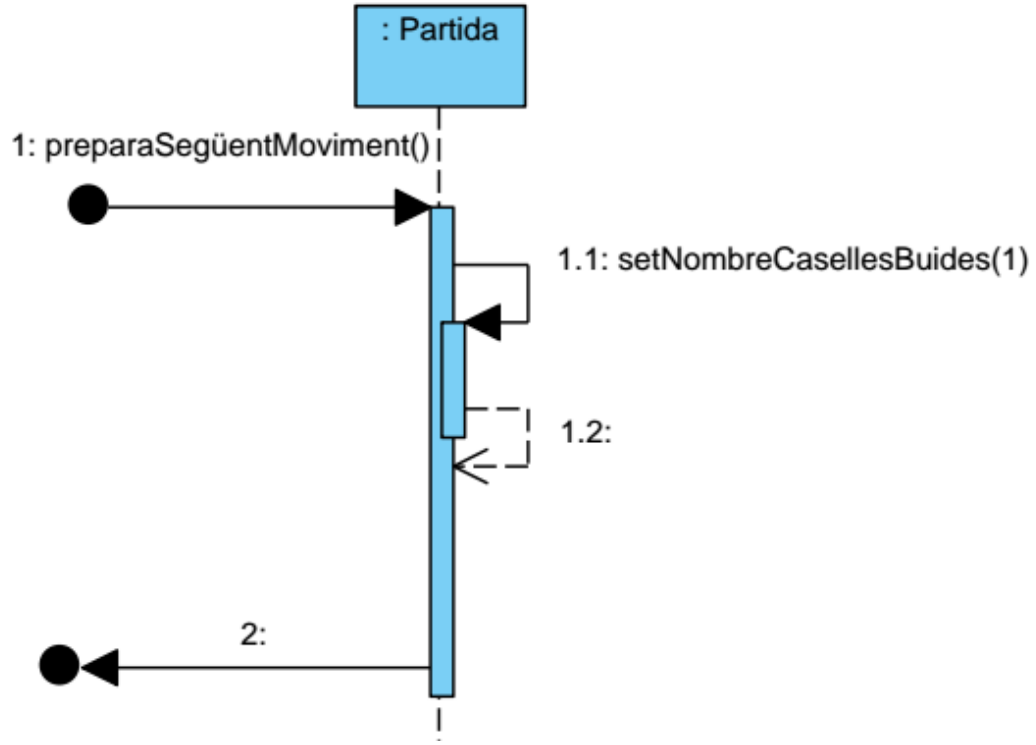

:Jugador
<<Singleton>>
:AdaptersFactory
<<Interface>>
imsa:IMessaginServiceAdapter
enviarMissatge(idPartida,puntuacio)
getMessagingServiceAdpter(): IMessagingServiceAdapter
imsa
msg:= (String)idpartida + " " + (String)Puntuacio
enviarMissatge(email,msg)


:MessaginServiceAdapter
<<Singleton>>
:ServieLocator
srvmsg:mailService
enviarMissatge(email,msg)
find("serveiXXXX")
srvmsg
enviarMissatge(email,msg)

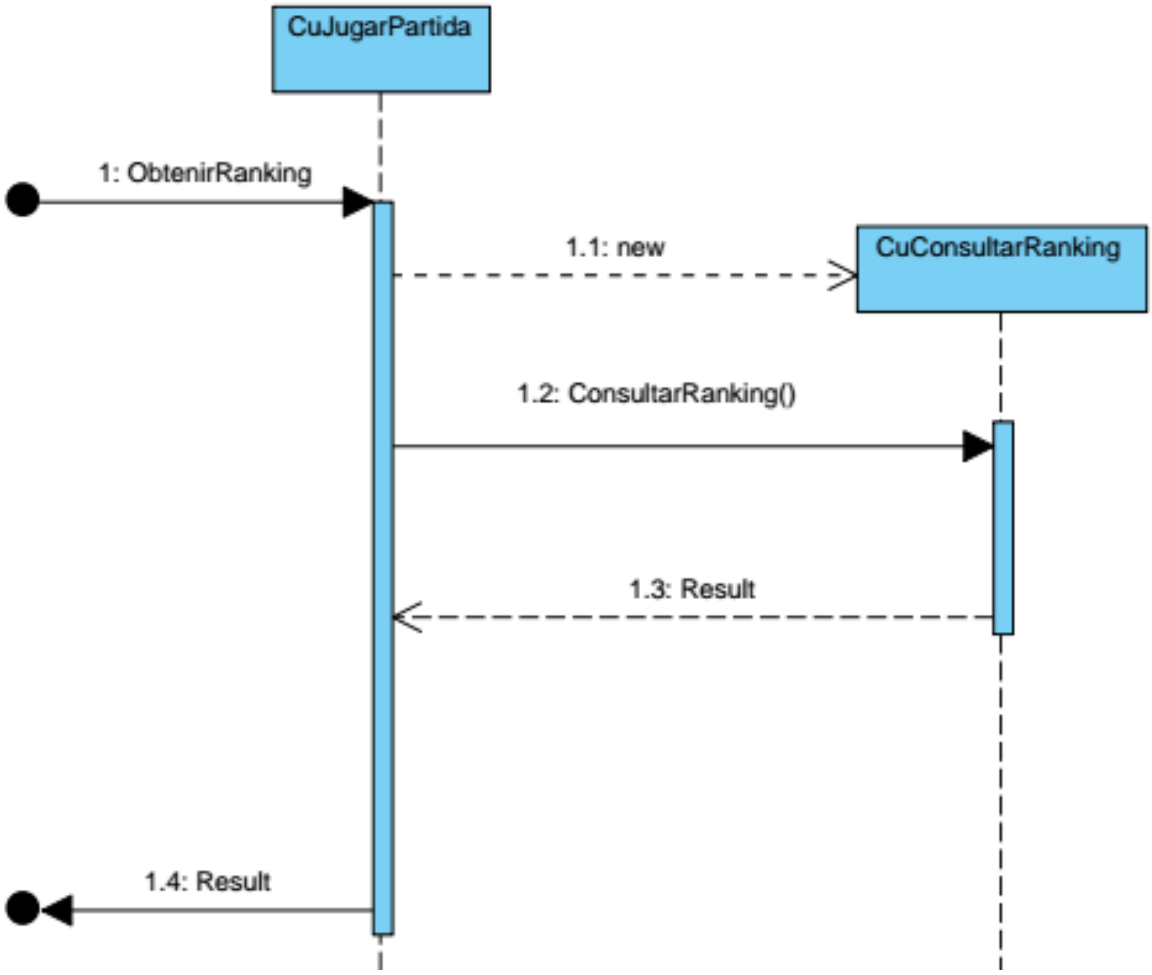

: Partida
jugadorPartidaActual : Jugador
1: FinalitzarPartida()
1.1: AcabarPartida
1.2:
1.3: AcabarPartida
1.4:
2:


: Jugador
partidaActual : Partida
1: AcabarPartida
1.1: getPuntuacio
1.2: puntuacio
si puntuacio > millorPuntuacio llavors
millorPuntuacio = puntuacio
partidaJugada.add(partidaActual)
partidaActual = null
1.3:

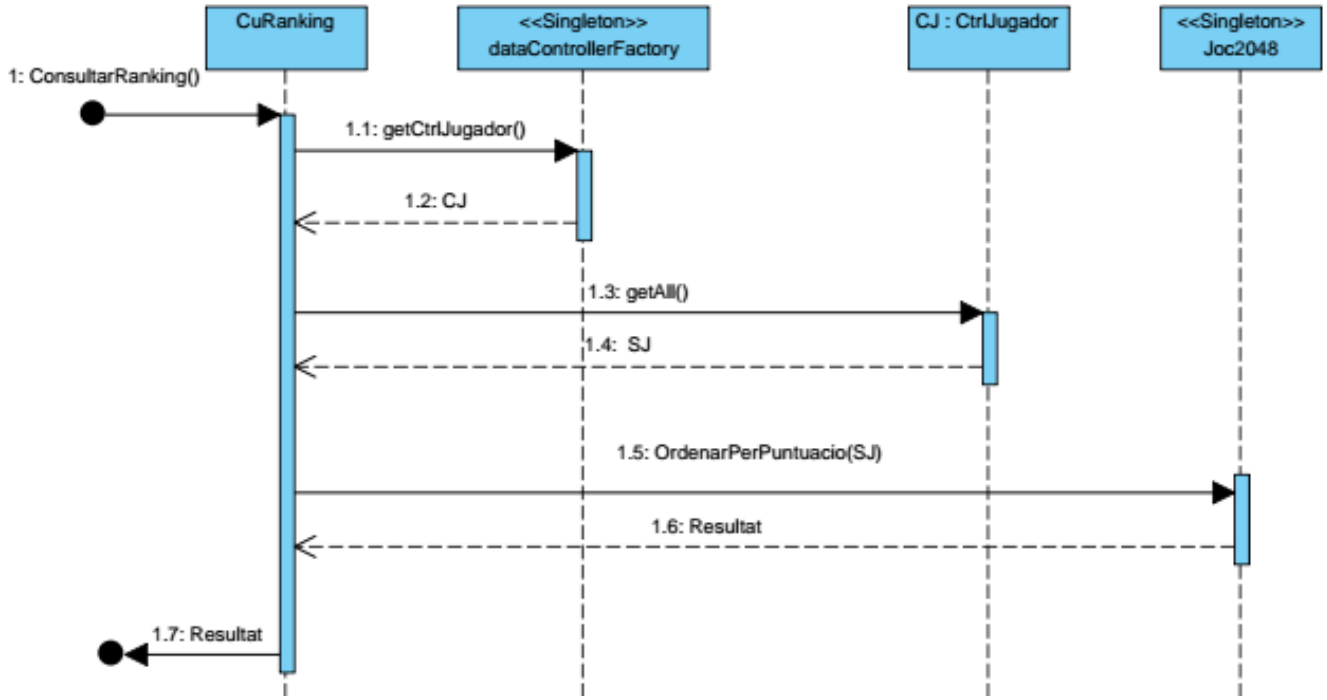

: Partida
1: AcabarPartida
jugadorPartidaJugada = jugadorPartidaActual
jugadorPartidaActual = null
1.1:

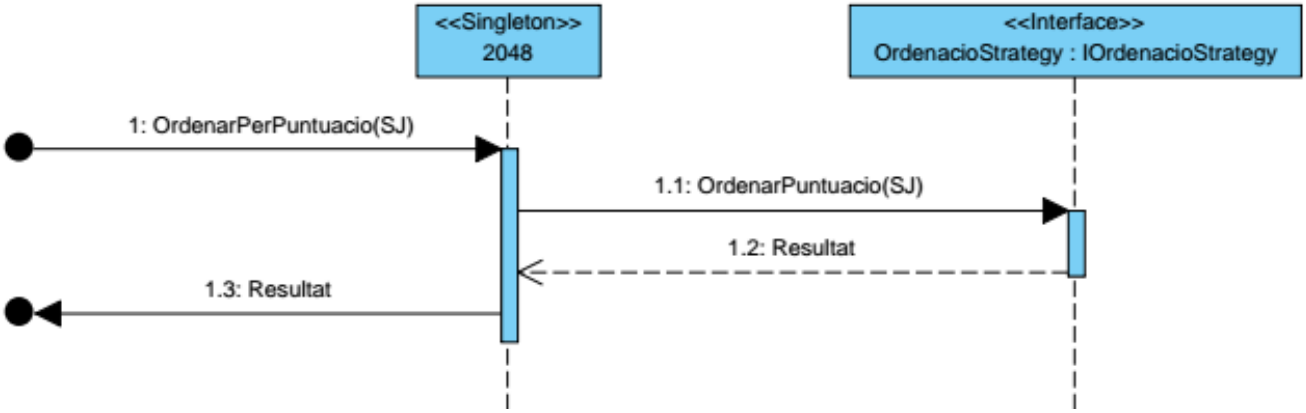


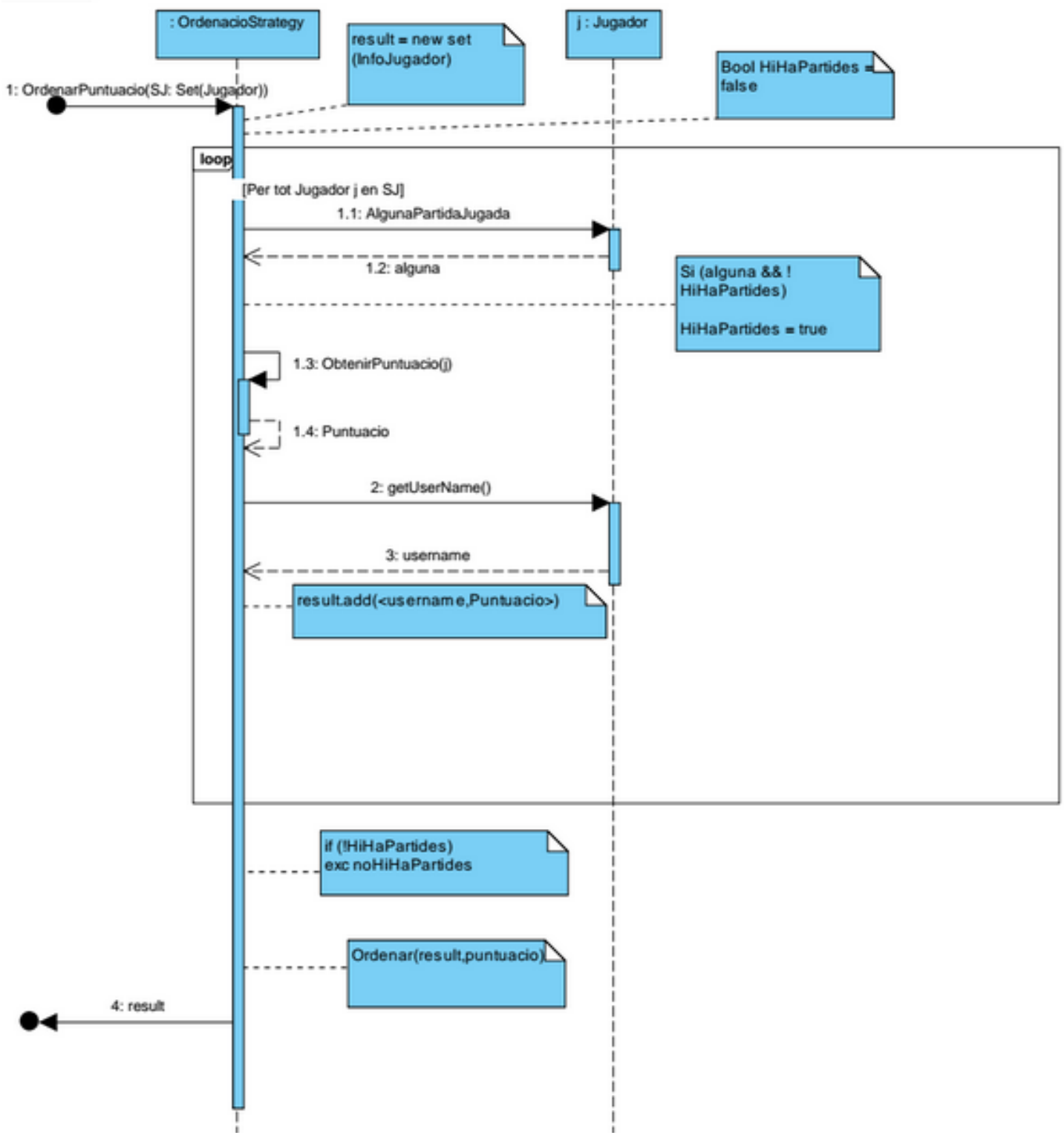
iv. Contrato ObtenirRanking

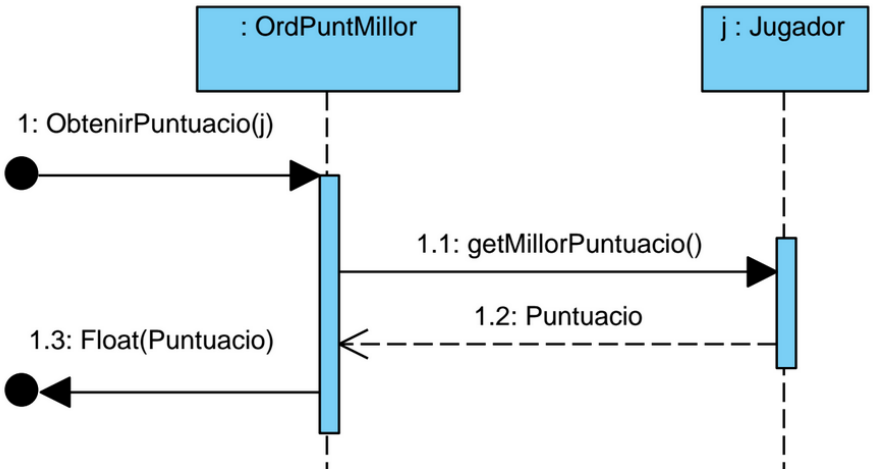


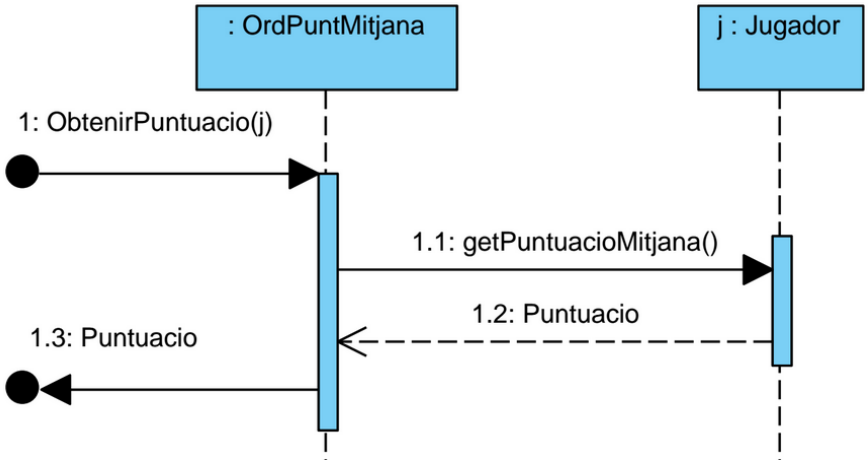
c. Caso de uso Consultar Ranking

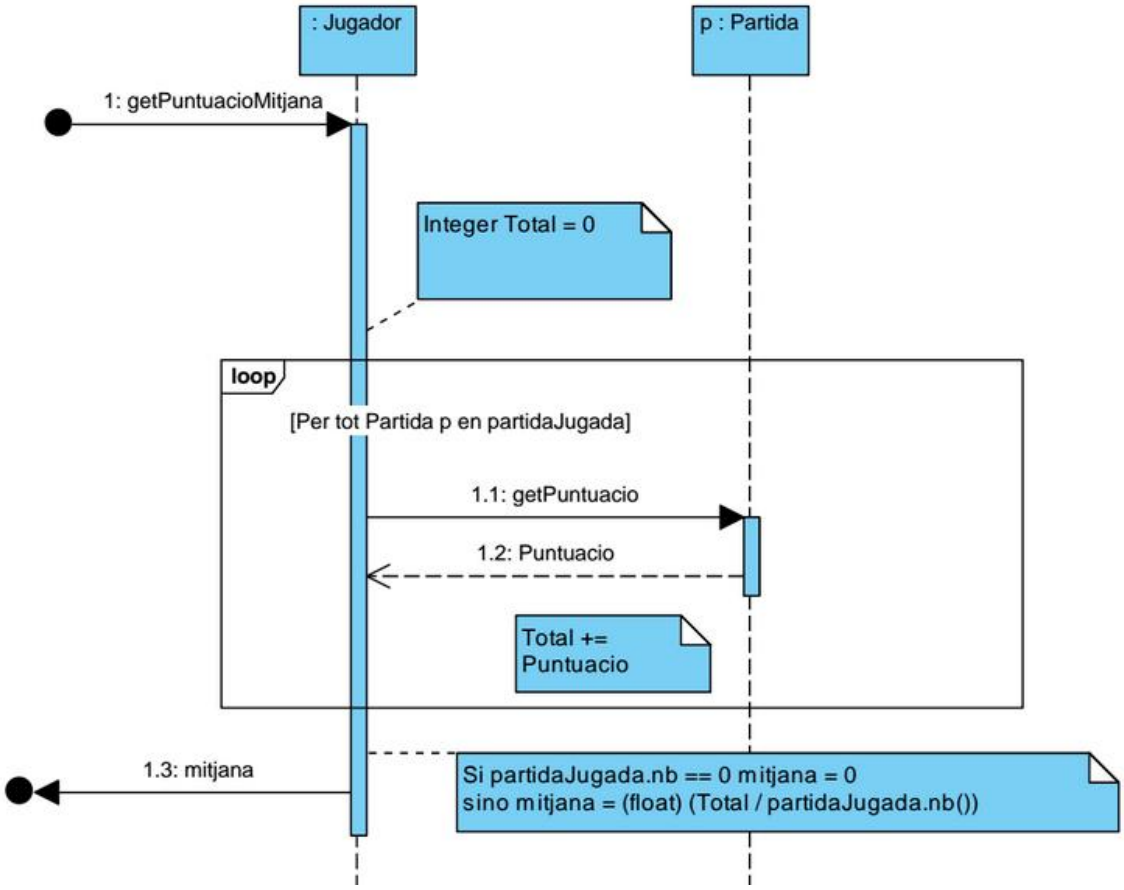


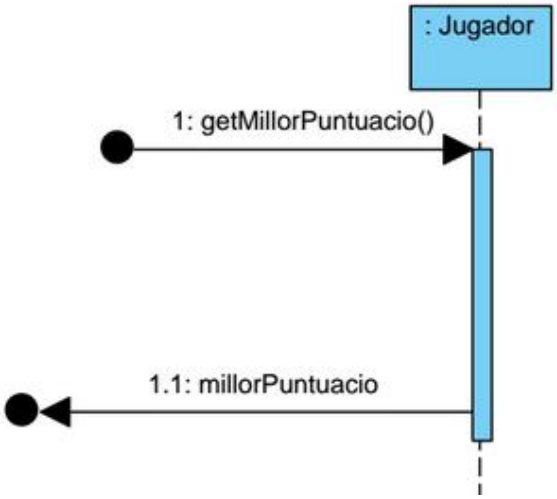


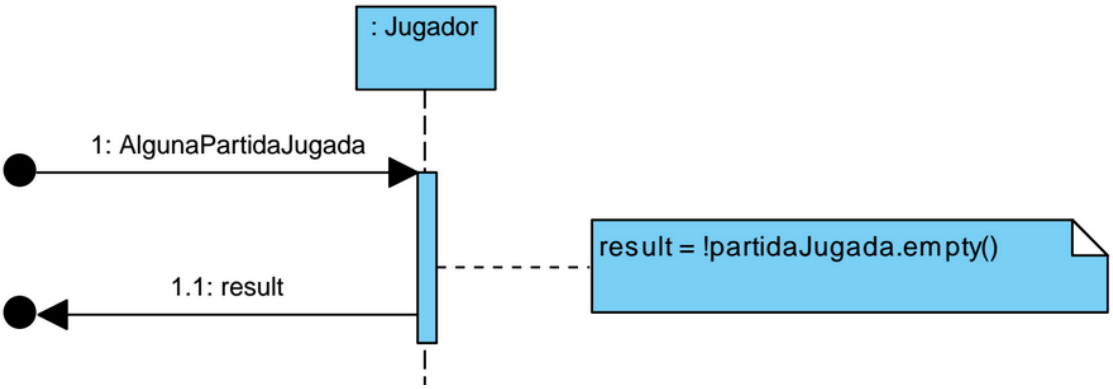

: OrdenacioStrategy
result = new set (InfoJugador)
j : Jugador
Bool HiHaPartides = false
1: OrdenarPuntuacio(SJ: Set(Jugador))
loop
[Per tot Jugador j en SJ]
1.1: AlgunaPartidaJugada
1.2: alguna
Si (alguna && ! HiHaPartides)
HiHaPartides = true
1.3: ObtenirPuntuacio(j)
1.4: Puntuacio
2: getUserName()
3: username
result.add(<username,Puntuacio>)
if (!HiHaPartides) exc noHiHaPartides
Ordenar(result,puntuacio)
4: result


: OrdPuntMillor
j : Jugador
1: ObtenirPuntuacio(j)
1.1: getMillorPuntuacio()
1.2: Puntuacio
1.3: Float(Puntuacio)

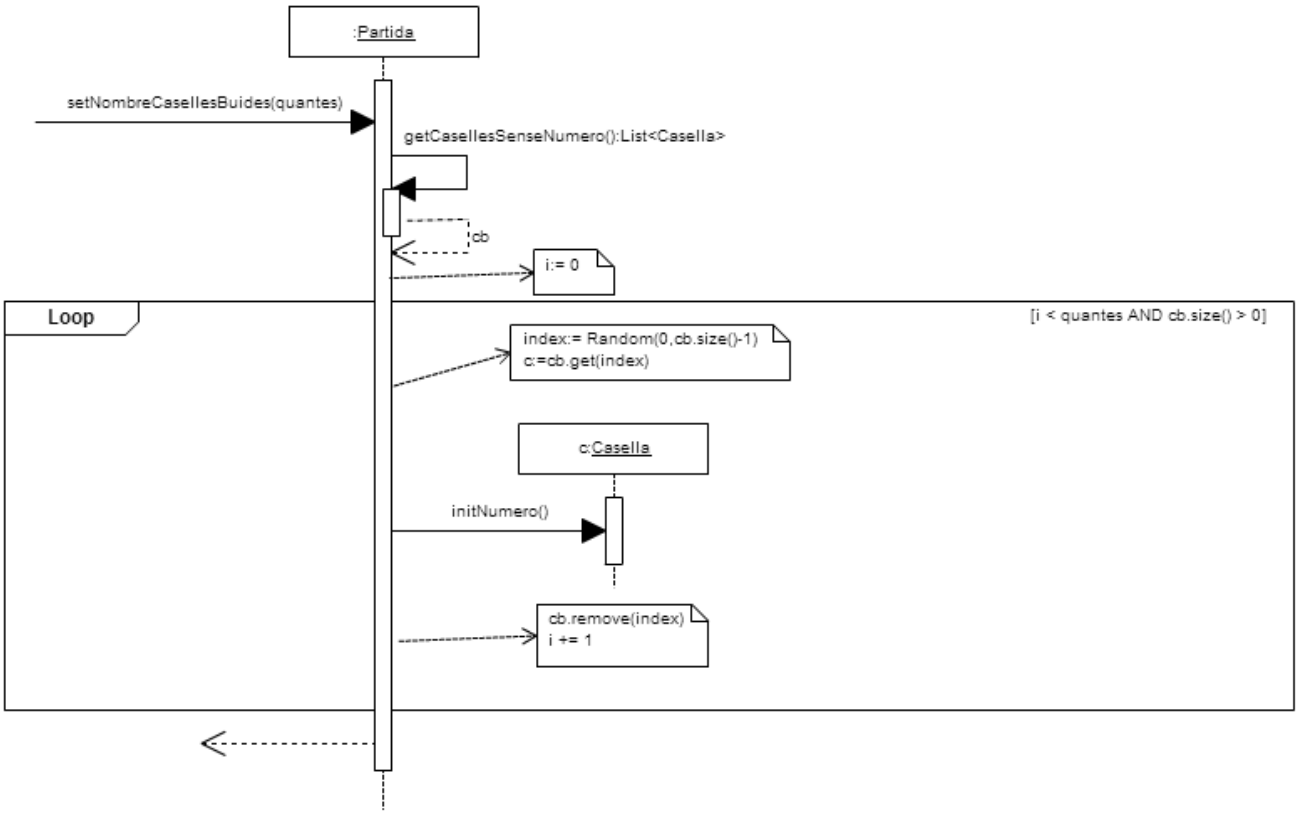

: OrdPuntMitjana
j : Jugador
1: ObtenirPuntuacio(j)
1.1: getPuntuacioMitjana()
1.2: Puntuacio
1.3: Puntuacio

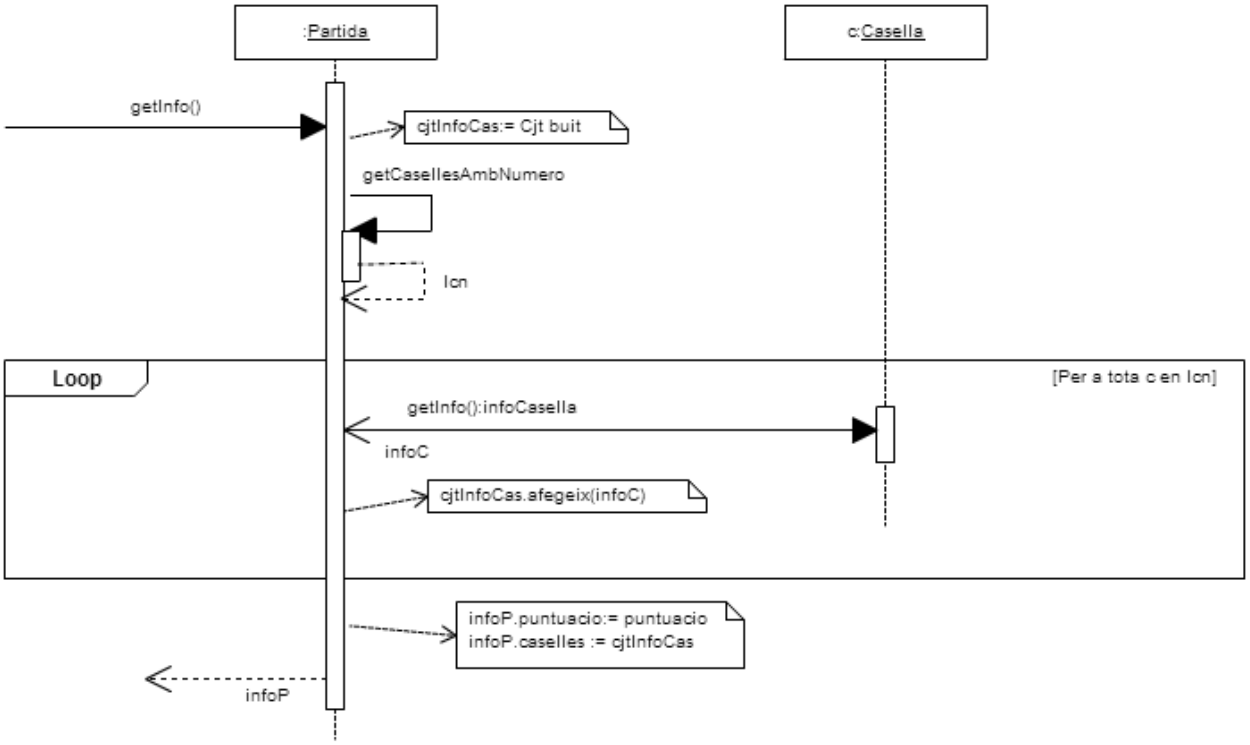

: Jugador
p : Partida
1: getPuntuacioMitjana
Integer Total = 0
loop
[Per tot Partida p en partidaJugada]
1.1: getPuntuacio
1.2: Puntuacio
Total +=
Puntuacio
1.3: mitjana
Si partidaJugada.nb == 0 mitjana = 0
sino mitjana = (float) (Total / partidaJugada.nb())

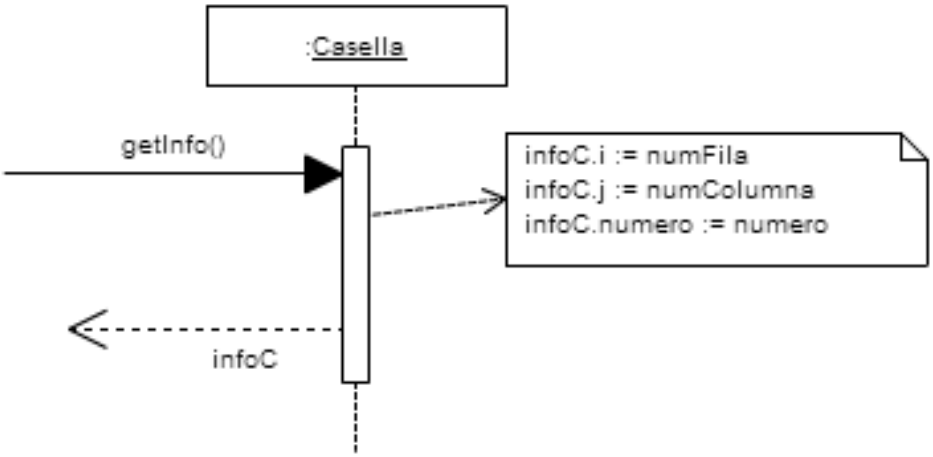

: Jugador
1: getMillorPuntuacio()
1.1: millorPuntuacio

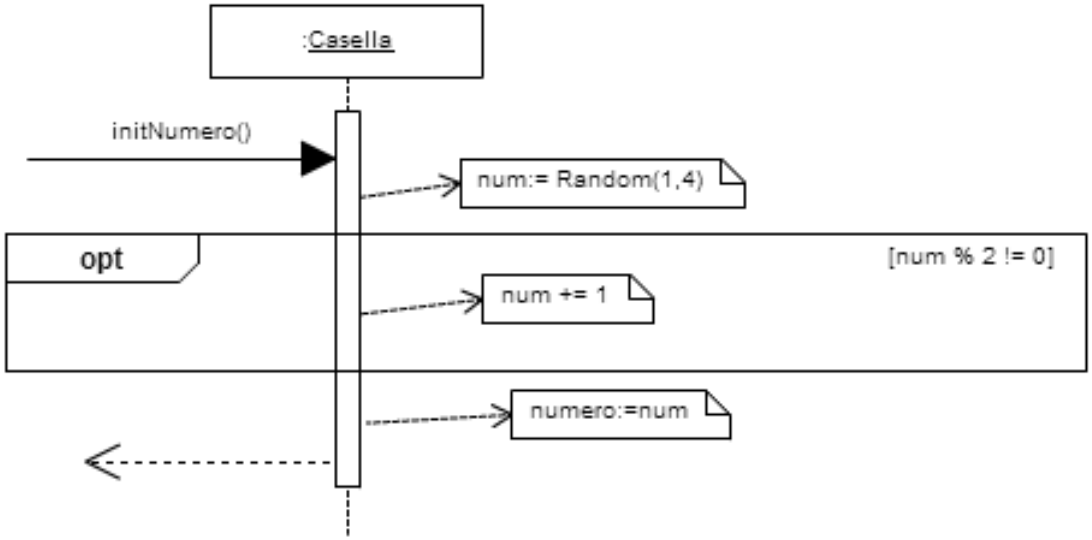

: Jugador
1: AlgunaPartidaJugada
result = !partidaJugada.empty()
1.1: result

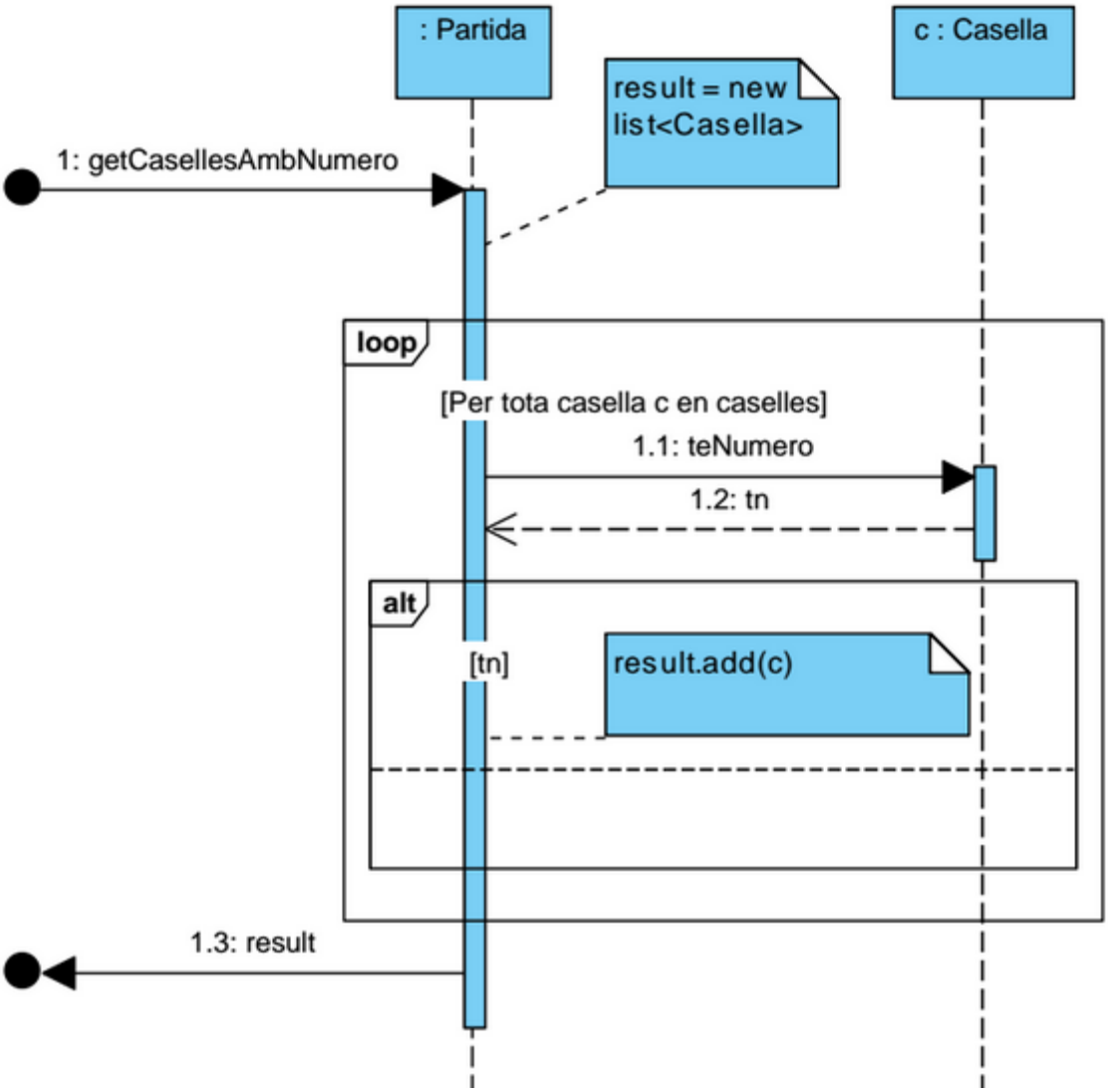
d. Utilitats comunes entre contractes i casos d'us

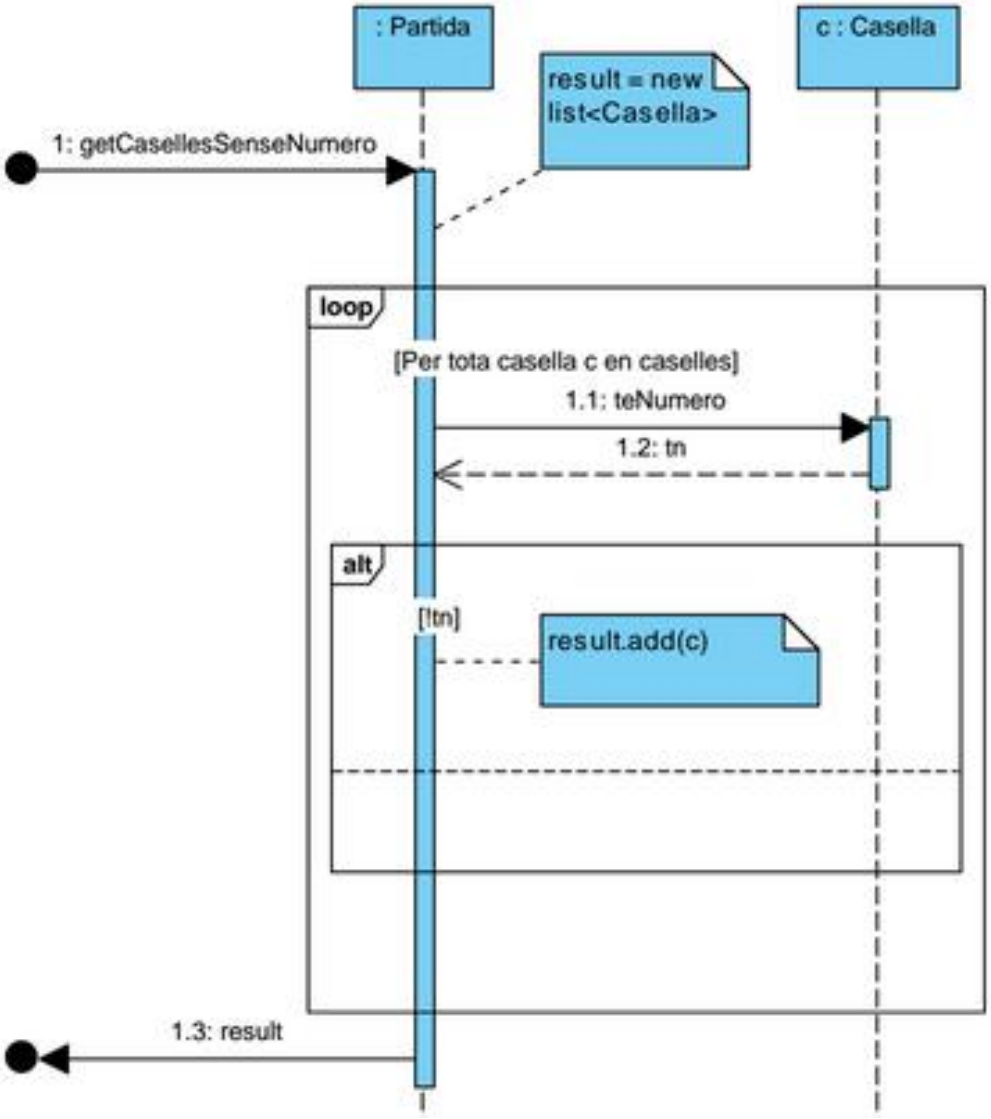


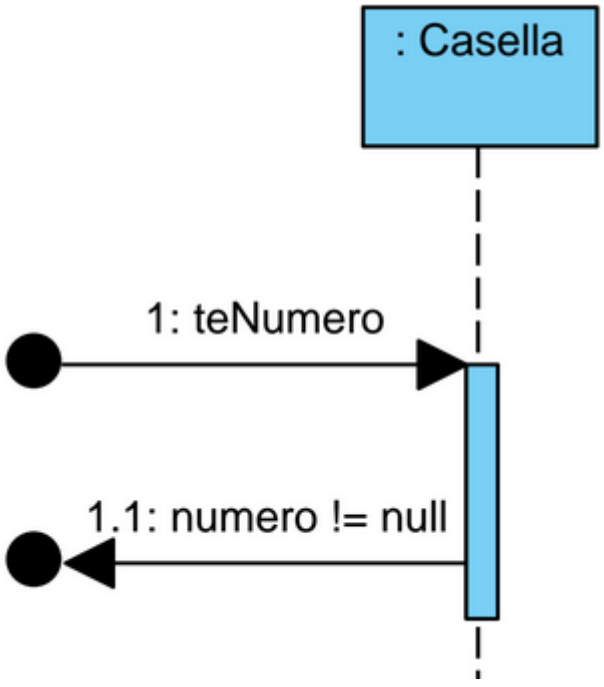


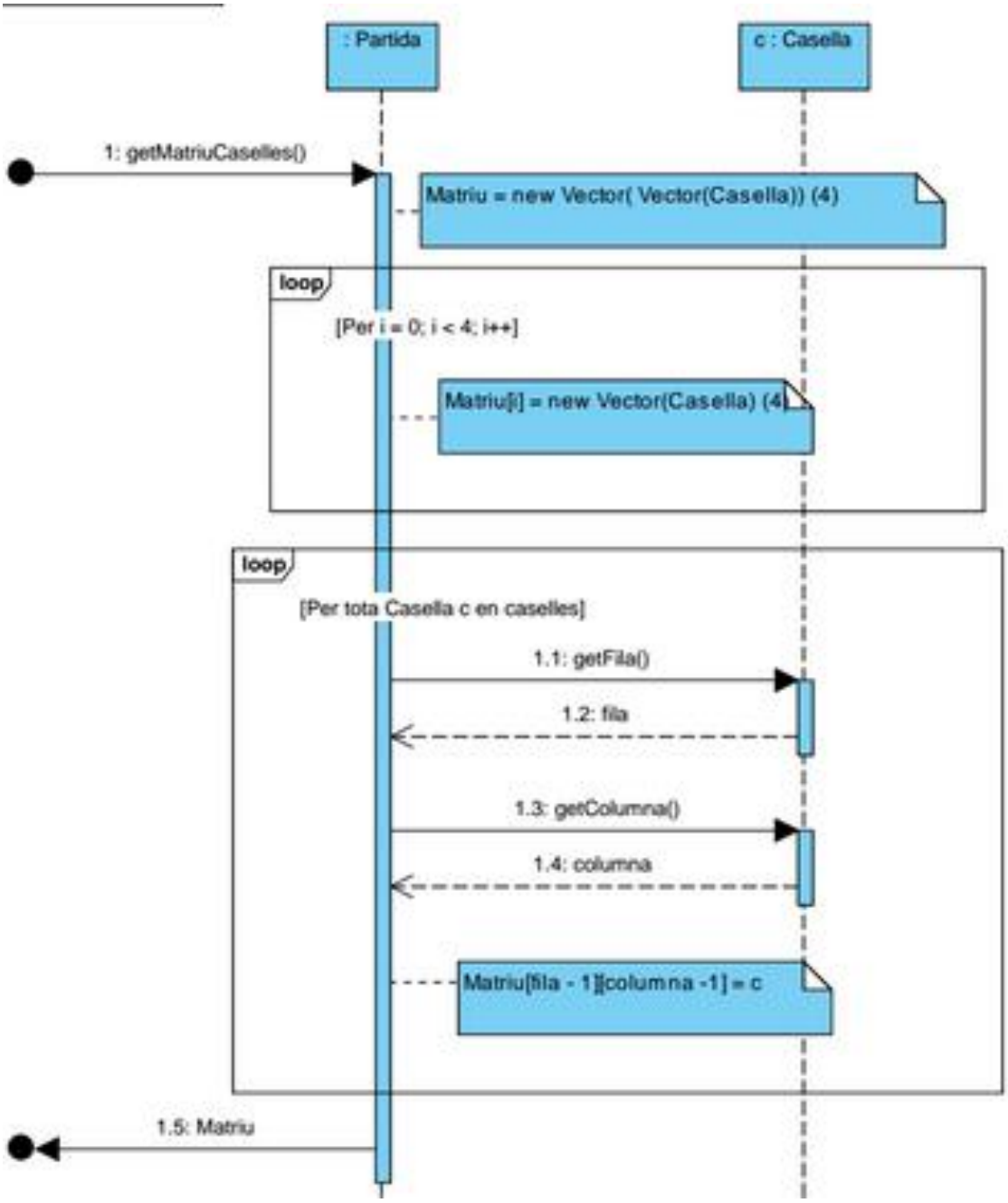

:Casella
getInfo()
infoC.i := numFila
infoC.j := numColumna
infoC.numero := numero
infoC

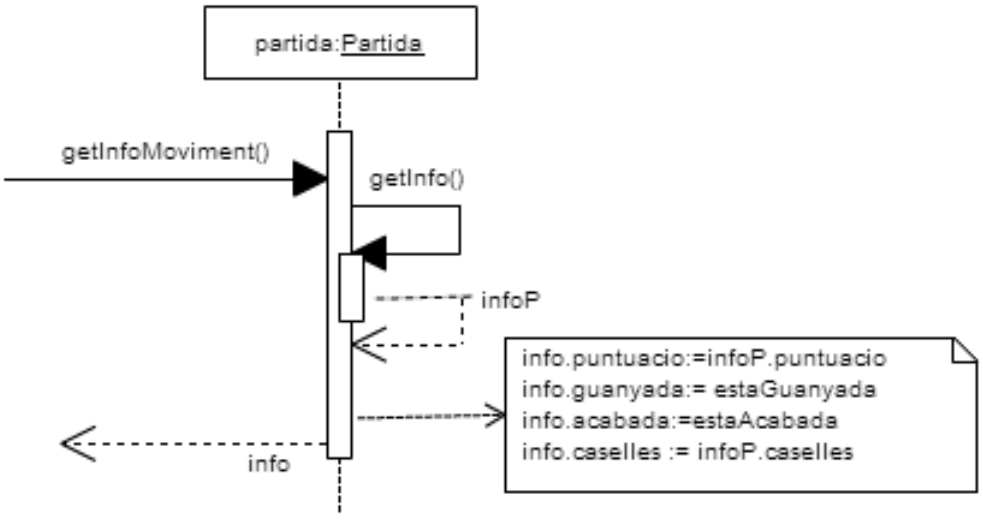

:Casella
initNumero()
num:= Random(1,4)
opt
[num % 2 != 0]
num += 1
numero:=num


: Partida
c : Casella
result = new list<Casella>
1: getCasellesAmbNumero
loop
[Per tota casella c en caselles]
1.1: teNumero
1.2: tn
alt
[tn]
result.add(c)
1.3: result

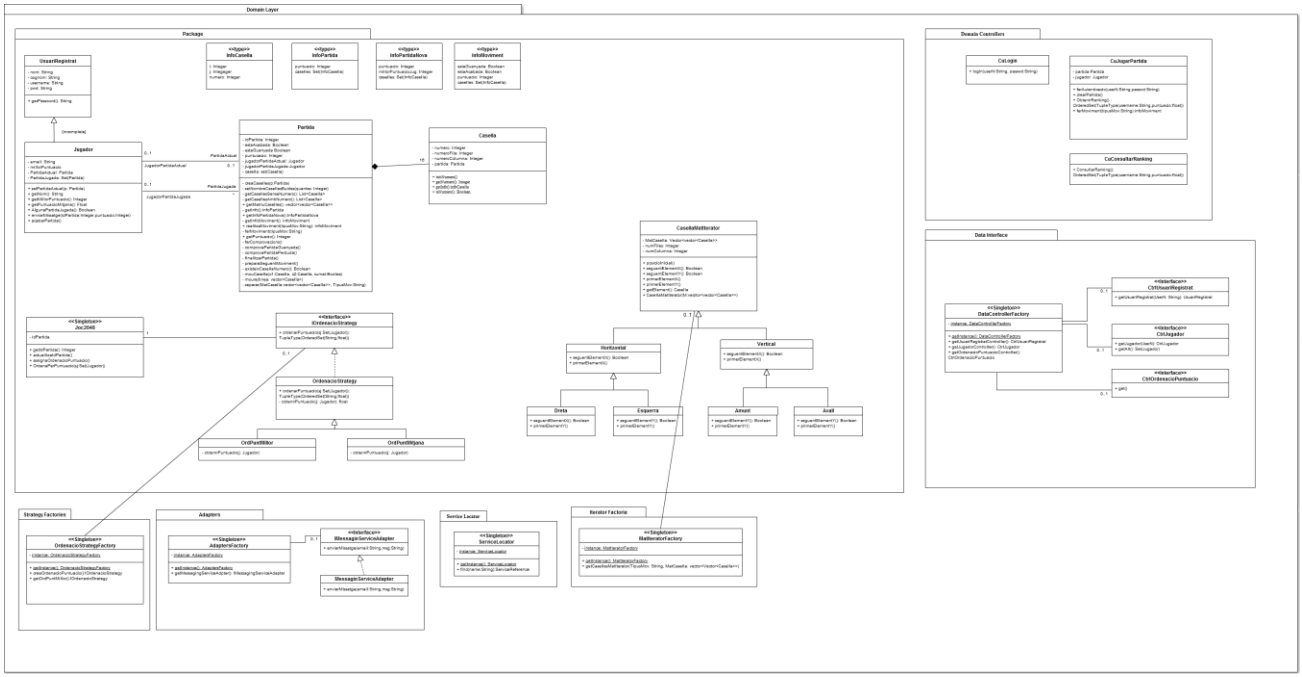

: Partida
c : Casella
result = new list<Casella>
1: getCasellesSenseNumero
loop
[Per tota casella c en caselles]
1.1: teNumero
1.2: tn
alt
[!tn]
result.add(c)
1.3: result


: Casella
1: teNumero
1.1: numero != null


: Partida
c : Casella
1: getMatriuCaselles()
Matriu = new Vector(Vector(Casella)) (4)
loop
[Per i = 0; i < 4; i++]
Matriu[i] = new Vector(Casella) (4)
loop
[Per tota Casella c en caselles]
1.1: getFila()
1.2: fila
1.3: getColumna()
1.4: columna
Matriu[fila - 1][columna -1] = c
1.5: Matriu


partida:Partida
getInfoMoviment()
getInfo()
infoP
info.puntuacio:=infoP.puntuacio
info.guanyada:= estaGuanyada
info.acabada:=estaAcabada
info.caselles := infoP.caselles
info

c. Ejercicio 3



Adjuntem links del diagrama per a visualitzar-lo millor:

https://cacoo.com/diagrams/k3H6arAw0SPfpxzo

https://cacoo.com/diagrams/k3H6arAw0SPfpxzo-78B44.png

d. Ejercicio 4

a. Patró adaptador

El patró adaptador s'utilitza durant la fase de fer comprovacions del contracte de FerMoviment. La post comprovaPartidaGuanyada ens indica que cal enviar un missatge en el moment en que un jugador guanya la partida. Aquest missatge s'ha d'enviar mitjançant un servei de missatgeria del qual no ens donen més informació que la capcelera de l'operació. Aleshores, per la necessitat de permetre que la nostra interfície i la del servei es comuniquin i per desacoplar aquest servei del nostre sistema, hem usat el patró adaptador: hem definit la interfície IMessaginServiceAdapter implementada per la classe MessaginServiceAdapter, que se n'encarrega de cercar el servei de missatgeria i enviar-li el missatge (cridant a la funció corresponent del servei) que el jugador li proporciona.

b. Patró factoria

El patró factoria s'utilitza a les classes OrdenacioStrategyFactory, AdaptersFactory, MatIteratorFactory i DataControllerFactory. L'aplicació d'aquest patró sorgeix de la necessitat de tenir classes responsables de la creació d'objectes amb una creació lògica complexa (OrdenacioStrategyFactory, MatIteratorFactory) o bé per millorar la cohesió (AdaptersFactory, DataControllerFactory).

c. Patró estratègia i patró plantilla

El patró estratègia s'utilitza durant la fase d'obtenció del rànking de puntuacions. Sorgit de la necessitat de fer un disseny canviable per a dues polítiques d'ordenació diferents però relacionades creem la interfície IOrdenacióStrategy implementada per OrdenacioStrategy. A aquesta classe, donat que tenim una part de l'estratègia d'ordenació que és comuna independentment de quina estratègia fem servir, apliquem el patró plantilla. Així doncs, definim dues classes filles OrdPuntMillor i OrdPuntMitjana. D'aquesta forma, a OrdenacioStrategy fem la part comuna de les dues estratègies i a les classes filles hi definim la part no comuna.

d. Patró singleton

El patró singleton el fem servir a les classes OrdenacioStrategyFactory, AdaptersFactory, ServiceLocator, MatIteratorFactory i DataControllerFactory. L'aplicació d'aquest patró a la pràctica sorgeix de la necessitat de tenir al sistema certes classes (les mencionades anteriorment) amb una sola instància que sempre ha de ser accessible.

e. Patró iterador

El patró iterador s'utilitza durant la fase de separació de les caselles del tauler, la classe que el duu a terme és la "iteradorMatriuCaselles" que defineix una forma d'accedir a les caselles del tauler.

Aquesta classe utilitza una combinació del patró estratègia i l'iterador, integrant dins seu un iterador bastant simple sobre una matriu (utilitza 2 variables, que corresponen a la fila i la columna de la matriu en referència)

La necessitat d'utilitzar-lo sorgeix de l'abstracció de que qualsevol moviment (amunt,avall,dreta,esquerra) realitza el mateix si "girem" el tauler. La seqüència del moviment es podria definir com a única d'aquesta manera:

1: Girar el tauler segons el moviment tal que el moviment s'orientiï sempre cap a un únic sentit

2: Separar el tauler en 4 grups, els quals seràn format pel conjunt de caselles que es desplaçaràn de forma conjunta en el moviment

3: Moure cada grup de caselles

Aquesta classe s'encarrega del cas 1, utilitzant la estratègia i l'iterador, però en comptes de girar el tauler en sí mateix, interpreta els índex de la matriu com a punts en el pla i després canvia els eixos de coordenades aconseguint, d'aquesta forma, un efecte similar a la rotació.

La part d'aquesta classe que s'encarrega de realitzar el patró estratègia realitza el següent:

Diguem que 4 sub classes implementen la estratègia sobre com separar una matriu de caselles. Donat que existeixen 4 moviments, cada estratègia interpreta en el seu interior els eixos direccionals de la matriu per tal de dur a terme aquesta tasca.

Nota: no és una aplicació estricta del patró estratègia en el sentit de que no implementa una interfície, tot i així el comportament és similar.

La part que realitza l'iterador, s'encarrega d'enmagatzemar la posició actual de la matriu dins de cada objecte, i permet modificar els dos apuntadors en qualsevol moment dins de la pròpia classe, a més d'oferir una operació per accedir a l'element de la matriu actual. La forma de moure l'iterador sobre la matriu la duen a terme les 4 sub-classes, cadascuna de la seva forma.

L'eix de les abscises (x) serà l'eix que contindrà els 4 grups de caselles (independents entre ells pel moviment). L'eix de les ordenades (y) contindrà l'apuntador a la casella dins de cada grup.

Així doncs, per tal de separar la matriu caldrà primer recorrer tots els elements de l'eix x; i per cada un d'ells agafar les 4 caselles dels punts (x,0) ... (x,3) Aquesta feina la realitza el pas 2, i és inherent d'aquesta classe

En definitiva, el que aquesta classe permet és dividir convenientment la matriu segons el tipus de moviment desitjat.

Exemple:

Partim d'aquest tauler i del moviment cap amunt:

		2	4
	4	4	
8			8
	4	2	

La classe amunt interpetarà la matriu com:

eix X				
		2	4	eix Y
	4	4		
8			8	
	4	2		