

# Software Ingeniaritza 2

## Sonar ingurunea

Iñaki Moreno, Zdravko Todorov eta Aitor Paredes

### SonarLint aurreko urteko Bets proiektuari aplikatua

#### Bug eta Vulnerability

##### Bug#1:

Either override `Object.equals(Object)` or rename the method to prevent any confusion.

Lehen:

```
public boolean equals(Apustua apus) {
    if(this.pronostikoKop!=apus.getPronostikoKop()) {
        return false;
    }
    for(Pronostikoa p : pronostikoak) {
        if(!apus.baduPronostikoa(p)) {
            return false;
        }
    }
    return true;
}
```

Orain:

```
public boolean apustuaEquals(Apustua apus) {
    if(this.pronostikoKop!=apus.getPronostikoKop()) {
        return false;
    }
    for(Pronostikoa p : pronostikoak) {
        if(!apus.baduPronostikoa(p)) {
            return false;
        }
    }
    return true;
}
```

**Bug#2:**

Use equals method if value comparison was intended

Lehen:

```
public boolean eguneratuAsmatutakoKop() {  
    asmatutakoKop=asmatutakoKop+1;  
    System.out.println(asmatutakoKop+"=="+"pronostikoKop");  
    return asmatutakoKop==pronostikoKop;  
}
```

Orain:

```
public boolean eguneratuAsmatutakoKop() {  
    asmatutakoKop=asmatutakoKop+1;  
    System.out.println(asmatutakoKop+"=="+"pronostikoKop");  
    return asmatutakoKop.equals(pronostikoKop);  
}
```

**Bug#3:**

Remove or correct this removeAll call

Lehen:

```
pronostikoak.removeAll(pronostikoak);
```

Orain:

```
pronostikoak.clear();
```

\*pronostikoak ArrayList bat da

**Bug#4:**

Use equals method if value comparison was intended

Lehen:

```
public boolean hartutaDago(Question question) {
    for(Pronostikoa p : pronostikoak) {

        if(p.getQuestion().getQuestionNumber()==question.getQuestionNumber()) {
            return true;
        }
    }
    return false;
}
```

Orain:

```
public boolean hartutaDago(Question question) {
    for(Pronostikoa p : pronostikoak) {
        if(p.getQuestion().getQuestionNumber().equals(
question.getQuestionNumber())) {
            return true;
        }
    }
    return false;
}
```

**Bug#5:**

Use equals method if value comparison was intended

Lehen:

```
public Mugimendua addMugimendua(String deskripzioa, double kuota, String
mota, Date data) {
    eguneratuDirua(kuota);
    Mugimendua berria = new Mugimendua(deskripzioa, kuota, this,
mota, data);
    mugimenduak.add(berria);
    if(mota=="irabazi") {
        irabaziak++;
    }
    return berria;
}
```

Orain:

```
public Mugimendua addMugimendua(String deskripzioa, double kuota, String mota, Date data) {
    eguneratuDirua(kuota);
    Mugimendua berria = new Mugimendua(deskripzioa, kuota, this, mota, data);
    mugimenduak.add(berria);
    if(mota.equals("irabazi")) {
        irabaziak++;
    }
    return berria;
}
```

**Bug#6:**

**A "NullPointerException" could be thrown; "getElkarrizketa()" can return null.**

Lehen:

```
row.add(getElkarrizketa((ArretaMezua)m).getGaia());
```

Orain:

```
ArretaElkarrizketa tmp = getElkarrizketa((ArretaMezua)m);
if(tmp != null) {
    row.add(tmp.getGaia());
}
```

**Bug#7:**

**Use equals method if value comparison was intended**

Lehen:

```
bet.getAsmatutakoKop()==bet.getPronostikoKop()
```

Orain:

```
bet.getAsmatutakoKop().equals(bet.getPronostikoKop())
```

**Bug#8:**

Use equals method if value comparison was intended

Lehen:

```
am.getIdentifikadorea()==m.getIdentifikadorea()
```

Orain:

```
am.getIdentifikadorea().equals(m.getIdentifikadorea())
```

**Bug#9:**

Either override Object.equals(Object) or rename the method to prevent any confusion.

Lehen:

```
public boolean equals(Pronostikoa p) {  
    return p.deskripzioa.equals(this.deskripzioa) &&  
    p.question.getQuestionNumber()==question.getQuestionNumber();  
}
```

Orain:

```
public boolean pronostikoaEquals(Pronostikoa p) {  
    return p.deskripzioa.equals(this.deskripzioa) &&  
    p.question.getQuestionNumber()==question.getQuestionNumber();  
}
```

**Bug#10:**

Refactor this repetition that can lead to a stack overflow for large inputs.

Lehen:

```
pattern=Pattern.compile("^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9- ]+)*@"+"[A-Za-z0-9]+(\\. [A-Za-z0-9]+)*([A-Za-z]{2,})$");
```

Orain:

```
pattern=Pattern.compile("^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9- ]+){0,3}@"+"[A-Za-z0-9]+(\\. [A-Za-z0-9]+){0,1}([A-Za-z]{2,})$");
```

**Bug#11:**

Use equals method if value comparison was intended

Lehen:

```
this.pronostikoKop!=apus.getPronostikoKop()
```

Orain:

```
!this.pronostikoKop.equals(apus.getPronostikoKop())
```

**Bug#12:**

Use equals method if value comparison was intended

Lehen:

```
return p.deskripzioa.equals(this.deskripzioa) &&  
p.question.getQuestionNumber()==question.getQuestionNumber()
```

Orain:

```
return p.deskripzioa.equals(this.deskripzioa) &&  
p.question.getQuestionNumber().equals(question.getQuestionNumber())
```

**Bug#13:**

Either re-interrupt this method or rethrow the “InterruptedException” that can be caught here

Lehen:

```
} catch (Exception e) {  
    System.out.println("Something has happened in  
ObjectdbManagerServer: "+e.toString());  
}
```

Orain:

```
} catch (Exception e) {  
    Thread.currentThread().interrupt();  
    System.out.println("Something has happened in  
ObjectdbManagerServer: "+e.toString());  
}
```

### Vulnerability#1:

#### Disable access to external entities in XML parsing

Lehen:

```
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
```

Orain:

```
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();  
dbFactory.setFeature("http://apache.org/xml/features/disallow-doctype-de  
cl", true);
```

## Code Smell

### Code Smell (Minor) #1:

Replace the type specification in this constructor call with the diamond operator("<>")

Lehen:

```
private ArrayList<Pronostikoa> pronostikoak = new  
ArrayList<Pronostikoa>();
```

Orain:

```
private ArrayList<Pronostikoa> pronostikoak = new ArrayList<>();
```

### Code Smell (Major) #2:

This block of commented-out lines of code should be removed

Lehen:

```
//import domain.Booking;  
import domain.Question;
```

Orain:

```
import domain.Question;
```

### Code Smell (Info) #3:

Complete the task associated to this TODO comment

Lehen:

```
// TODO Auto-generated catch block
System.out.println("Error: "+e1.toString()+" , probably problems with
Business Logic or Database");
```

Orain:

```
System.out.println("Error: "+e1.toString()+" , probably problems with
Business Logic or Database");
```

### Code Smell (Critical) #4:

Use static access with “javax.swing.WindowConstants” for “EXIT\_ON\_CLOSE”

Lehen:

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Orain:

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

### Code Smell (Minor) #5:

Rename this method to match the regular expression ‘^[a-z][a-zA-Z0-9]\*\$’

Lehen:

```
public boolean DoesPronosticExists(String deskripzioa)
```

Orain:

```
public boolean doesPronosticExists(String deskripzioa)
```

### Code Smell (Major) #6:

Change the visibility of this constructor to “protected” (klase abstraktuen erailkitzaileei soilik beraien subklaseek egin diezaiekete dei, beraz public izateak ez du zentzurik)



Lehen:

```
public Mezua (String mezua)
```

Orain:

```
protected Mezua (String mezua)
```

### Code Smell (Critical) #7:

Define a constant instead of duplicating this literal “initialize” 3 times

Lehen:

```
dbManager=new DataAccess(c.getDataBaseOpenMode().equals("initialize"));  
...  
if (c.getDataBaseOpenMode().equals("initialize"))  
...  
if (c.getDataBaseOpenMode().equals("initialize"))
```

Orain:

```
private String initializeStr = "initialize";  
...  
dbManager=new DataAccess(c.getDataBaseOpenMode().equals(initializeStr);  
...  
if (c.getDataBaseOpenMode().equals(initializeStr))  
...  
if (c.getDataBaseOpenMode().equals(initializeStr))
```

### Code Smell (Minor) #8:

Remove this empty statement

Lehen:

```
...  
};
```

Orain:

```
...  
}
```

#### **Code Smell (Critical) #9:**

**Use static access with “javax.swing.WindowConstants” for “DISPOSE\_ON\_CLOSE”**

Lehen:

```
dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
```

Orain:

```
dialog.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
```

#### **Code Smell (Major) #10:**

**This block of commented-out lines of code should be removed**

Lehen:

```
//if (Locale.getDefault().equals(new Locale("es")))  
//    offset += 4;  
//else  
    offset += 5;
```

Orain:

```
offset += 5;
```

#### **Code Smell (Minor) #11:**

**Use “Double.parseDouble” for this string-to-double conversion**

Lehen:

```
double bet = Double.valueOf(textFieldBet.getText());
```

Orain:

```
double bet = Double.parseDouble(textFieldBet.getText());
```

**Code Smell (Minor) #12:****Remove this unnecessary cast to “Component”**

Lehen:

```
Component o = (Component) jCalendar.getDayChooser().getDayPanel()
```

Orain:

```
Component o = jCalendar.getDayChooser().getDayPanel()
```

**Code Smell (Major) #13:****Remove this useless assignment to local variable “pronos12”**

Lehen:

```
pronos12 = q1.addPronostic("1", 1.2);  
...  
pronos12 = q3.addPronostic("1", 1.2);
```

Orain:

```
q1.addPronostic("1", 1.2);  
...  
pronos12 = q3.addPronostic("1", 1.2);
```

**Code Smell (Minor) #14:****Remove this unnecessary cast to “int”**

Lehen:

```
double kuotaZaharra = pronostikoak.get((int)pronosI).getKuota();
```

Orain:

```
double kuotaZaharra = pronostikoak.get(pronosI).getKuota();
```

**Code Smell (Major) #15:****This block of commented-out lines of code should be removed**

Lehen:

```
//private BezeroartekoMezuaContainer selectedMezua;  
private JScrollPane scrollPaneMezuak = new JScrollPane();
```

Orain:

```
private JScrollPane scrollPaneMezuak = new JScrollPane();
```

# Software proben praktika

Iñaki Moreno, Zdravko Todorov eta Aitor Paredes

Lan orduak guztira: 20 bat ordu

Github: <https://github.com/inakimoreno/Bets21>

SonarCloud: <https://sonarcloud.io/dashboard?id=BetsSI2>

## *register() - Aitor Paredes*

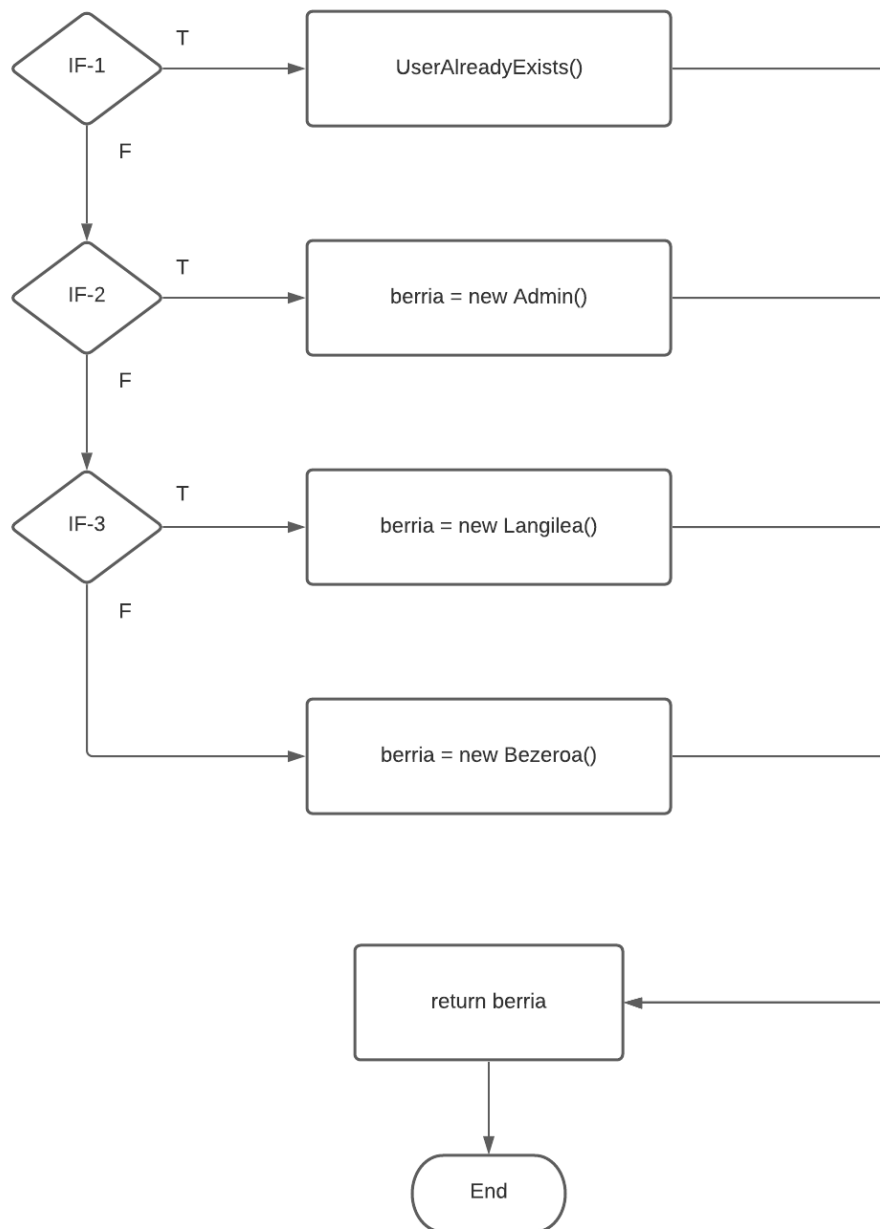
### Kodea

```
public Pertsona register(String izena, String abizena1, String abizena2, String erabiltzaileIzena,
                        String pasahitza, String telefonoZbkia, String emaila, Date jaiotzeData, String mota) throws UserAlreadyExist {

    TypedQuery<Pertsona> query = db.createQuery("SELECT p FROM Pertsona p WHERE p.erabiltzaileIzena=?1", Pertsona.class);
    query.setParameter(1, erabiltzaileIzena);
    List<Pertsona> pertsona = query.getResultList();
    if(!pertsona.isEmpty()) {
        throw new UserAlreadyExist();
    }else {
        Pertsona berria = null;
        if(mota.equals("admin")) {
            berria = new Admin(izena, abizena1, abizena2, erabiltzaileIzena, pasahitza, telefonoZbkia, emaila, jaiotzeData);
        }else if (mota.equals("langilea")) {
            berria = new Langilea(izena, abizena1, abizena2, erabiltzaileIzena, pasahitza, telefonoZbkia, emaila, jaiotzeData);
        }else {
            berria = new Bezeroa(izena, abizena1, abizena2, erabiltzaileIzena, pasahitza, telefonoZbkia, emaila, jaiotzeData);
        }
        db.getTransaction().begin();
        db.persist(berria);
        db.getTransaction().commit();
        return berria;
    }
}
```

## Kutxa txuriko probak

### Fluxu diagrama



## Konplexutasun ziklomatikoa

Metodo honen konplexutasun ziklomatikoa, erabaki nodoak zenbatuz eta bat gehitzen, 4 da.

### Bide posibleak

- 1- IF-1 (T)
- 2- IF-1 (F) IF-2 (T)
- 3- IF-1 (F) IF-2 (F) IF-3 (T)
- 4- IF-1 (F) IF-2 (F) IF-3 (F)

### Proba taula

#	Baldintza	Proba kontextua		Itxaron emaitzak	
		DB egoera	Sarrera	DB egoera	Irteera
1	1	$p \in DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "langile"	Ez da aldatzen	Exception UserAlreadyExist
2	2	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "admin"	$p = ("Antxon", "Urrutia",$ "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "admin") $\in$ DB	$p = ("Antxon", "Urrutia",$ "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "admin")
3	3	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "langilea"	$p = ("Antxon", "Urrutia",$ "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "langilea") $\in$ DB	$p = ("Antxon", "Urrutia",$ "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "langilea")
4	4	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	$p = ("Antxon", "Urrutia",$ "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa") $\in$ DB	$p = ("Antxon", "Urrutia",$ "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa")

$p = ("Antxon", "Urrutia", "Garcia", "antxon18", "12345678",$   
"666666666", ["antxon@gmail.com"](mailto:antxon@gmail.com), 10/10/1990, "langile" edo "bezeroa" edo "admin")

## Implementazioa: RegisterDAW.java

### Aurkitutako akatsak

Proba kasuek esperotako emaitza eman dute.

### Kutxa beltzeko probak

#### Baliokidetasun klaseak

<b>Sarrera-baldintza</b>	<b>Baliokidetasun klase egokiak</b>	<b>Baliokidetasun klase ez-egokiak</b>
izea balioa	izena != null (1)	izena == null (15)
abizena1 balioa	abizena1 != null (2)	abizena1 == null (16)
abizena2 balioa	abizena2 != null (3)	abizena2 == null (17)
erabiltzailelzena balioa	erabiltzailelzena != null (4)	erabiltzailelzena == null (18)
pasahitza balioa	pasahitza != null (5)	pasahitza == null (19)
telefonoZbkia balioa	telefonoZbkia != null (6)	telefonoZbkia == null (20)
emaila balioa	emaila != null (7)	emaila == null (21)
jaiotzeData balioa	jaiotzeData != null (8)	jaiotzeData == null (22)
mota balioa	mota != null (9)	mota == null (23)
adin handikoa	gaur.urtea - jaiotzeData.urtea >= 18 (10)	gaur.urtea - jaiotzeData.urtea < 18 (24)
pertsona ez DBn	pertsona ∉ DB (11)	pertsona ∈ DB (25)
pertsona administratzailea	mota == "admin" (12)	-
pertsona langilea	mota == "langilea" (13)	-
pertsona bezeroa	mota == "bezeroa" (14)	-

#### Proba taula



#	Estalitako BK	Proba kontextua		Esperotako emaitzak	
		DB egoera	Sarrera	DB egoera	Irteera
1	1,...,12	$p \notin \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "admin"	$p \in \text{DB}$	$p = (\text{"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", "antxon@gmail.com", 10/10/1990, "admin"})$
2	1,...,11,13	$p \notin \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "langilea"	$p \in \text{DB}$	$p = (\text{"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", "antxon@gmail.com", 10/10/1990, "langilea"})$
3	1,...,11,14	$p \notin \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	$p \in \text{DB}$	$p = (\text{"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", "antxon@gmail.com", 10/10/1990, "bezeroa"})$
4	15	$p \notin \text{DB}$	null, "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null
5	16	$p \notin \text{DB}$	"Antxon", null, "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null
6	17	$p \notin \text{DB}$	"Antxon", "Urrutia", null, "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null
7	18	*	"Antxon", "Urrutia", "Garcia", null, "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null
8	19	$p \notin \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", null, "666666666",	*	null

			" <a href="mailto:antxon@gmail.com">antxon@gmail.com</a> ", 10/10/1990, "bezeroa"		
9	20	$p \notin \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "null", " <a href="mailto:antxon@gmail.com">antxon@gmail.com</a> ", 10/10/1990, "bezeroa"	*	null
10	21	$p \notin \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", null, 10/10/1990, "bezeroa"	*	null
11	22	$p \notin \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", " <a href="mailto:antxon@gmail.com">antxon@gmail.com</a> ", null, "bezeroa"	*	null
12	23	$p \notin \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", " <a href="mailto:antxon@gmail.com">antxon@gmail.com</a> ", 10/10/1990, null	*	null
13	24	*	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", " <a href="mailto:antxon@gmail.com">antxon@gmail.com</a> ", 10/10/2015, "bezeroa"	*	null
14	25	$p \in \text{DB}$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", " <a href="mailto:antxon@gmail.com">antxon@gmail.com</a> ", 10/10/1990, "bezeroa"	*	UserAlreadyExist

$p =$  ("Antxon", "Urrutia", "Garcia", "antxon18", "12345678",  
"666666666", "[antxon@gmail.com](mailto:antxon@gmail.com)", 10/10/1990, "langile" edo "bezeroa" edo "admin")

Implementazioa: RegisterDAB.java

Aurkitutako akatsak

#	Proba kontextua	Esperotako emaitzak	Jasotako emaitzak
---	-----------------	---------------------	-------------------

	<b>DB egoera</b>	<b>Sarrera</b>	<b>DB egoera</b>	<b>Irteera</b>	<b>DB egoera</b>	<b>Irteera</b>
4	$p \notin DB$	null, "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null	$p \in DB$	$p = (\text{null, "Urrutia", "Garcia","antxon18", "12345678","666666666","antxon@gmail.com",10/10/1990, "bezeroa"})$
5	$p \notin DB$	"Antxon", null, "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null	$p \in DB$	$p = (\text{"Antxon", null, "Garcia","antxon18", "12345678","666666666","antxon@gmail.com",10/10/1990, "bezeroa"})$
6	$p \notin DB$	"Antxon", "Urrutia", null, "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null	$p \in DB$	$p = (\text{"Antxon", "Urrutia", null,"antxon18", "12345678","666666666","antxon@gmail.com",10/10/1990, "bezeroa"})$
7	$p \notin DB$	"Antxon", "Urrutia", "Garcia", null, "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null	?	$p = (\text{"Antxon", "Urrutia", "Garcia", null,"12345678","666666666","antxon@gmail.com",10/10/1990, "bezeroa"})$
8	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", null, "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null	$p \in DB$	$p = (\text{"Antxon", "Urrutia", "Garcia","antxon18", null,"666666666","antxon@gmail.com",10/10/1990, "bezeroa"})$
9	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", null, <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null	$p \in DB$	$p = (\text{"Antxon", "Urrutia", "Garcia","antxon18", "12345678",null,"antxon@gmail.com",10/10/1990, "bezeroa"})$
10	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", null, 10/10/1990, "bezeroa"	*	null	$p \in DB$	$p = (\text{"Antxon", "Urrutia", "Garcia","antxon18", "12345678","666666666",null, 10/10/1990, "bezeroa"})$
11	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , null, "bezeroa"	*	null	$p \in DB$	$p = (\text{"Antxon", "Urrutia", "Garcia","antxon18", "12345678","666666666","antxon@gmail.com", null, "bezeroa"})$

12	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "6666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, null	*	null	?	NullPointerException
13	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "6666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/2015, "bezeroa"	*	null	$p \in DB$	$p =$ ("Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "6666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/2015, "bezeroa")

## Integrazio probak

### Implementatutako proba kasuak

#	Estalitako BK	Proba kontextua		Esperotako emaitzak	
		DB egoera	Sarrera	DB egoera	Irteera
1	1,...,12	$p \notin DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "6666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "admin"	$p \in DB$	$p =$ ("Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "6666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "admin")
5	16	$p \notin DB$	"Antxon", null, "Garcia", "antxon18", "12345678", "6666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	null
14	25	$p \in DB$	"Antxon", "Urrutia", "Garcia", "antxon18", "12345678", "6666666666", <a href="mailto:antxon@gmail.com">"antxon@gmail.com"</a> , 10/10/1990, "bezeroa"	*	UserAlreadyExist

Implementazioa: RegisterMockInt.java

Aurkitutako akatsak:

Proba kasuek esperotako emaitza eman dute.

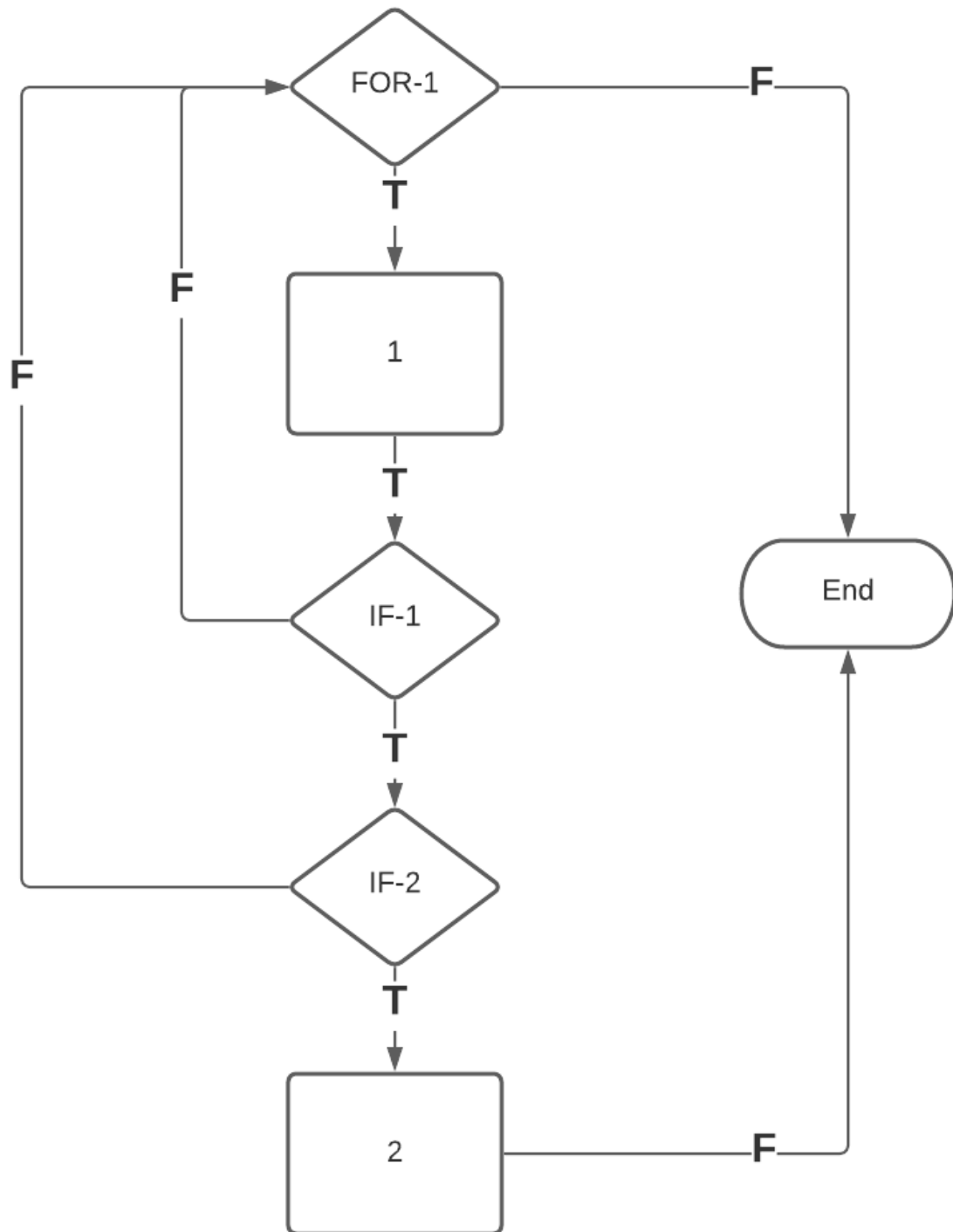
## *emaitzaIpini() - Iñaki Moreno*

### Kodea

```
public Vector<Double> emaitzaIpini(Question question, Pronostikoa pronostikoa){
    Pronostikoa p =
        db.find(Pronostikoa.class, pronostikoa.getIdentifikadorea());
    Question q = db.find(Question.class, question.getQuestionNumber());
    db.getTransaction().begin();
    q.setResult(pronostikoa.getDeskripzioa());
    Vector<Apustua> apustuak = p.getApustuak();
    Bezeroa bezeroa;
    double irabazia=0;
    boolean irabazi;
    double komisiao;
    double apustuKop=0;
    for(Apustua a : apustuak) {
        apustuKop++;
        irabazi=a.eguneratuAsmatutakoKop();
        komisiao=0;
        if(irabazi) {
            if (a.getErrepikatua()!=null) {
                Bezeroa bez = a.getErrepikatua();
                bezeroa = a.getBezeroa();
                Errepikapena errepikapen=bezeroa.getErrepikapena(bez);
                System.out.println(a.getKopurua()+" "+a.getKuotaTotala()+" "+a.getKopurua()+" "+errepikapen.getKomisiao());
                komisiao=(a.getKopurua()*a.getKuotaTotala()-a.getKopurua())*errepikapen.getKomisiao();
                System.out.println(komisioa);
                bez.addMugimendua("Apustu errepikatuaren komisiao ("+"bezeroa+""), komisiao,"irabazi");
            }
            bezeroa=a.getBezeroa();
            irabazia=a.getKopurua()*a.getKuotaTotala()-komisiao;
            bezeroa.addMugimendua("Apustua irabazi ("+"a.getIdentifikadorea()+"")", irabazia, "irabazi");
        }
    }
    db.getTransaction().commit();
    Vector<Double> em = new Vector<Double>();
    em.add(apustuKop);
    em.add(irabazia);
    return em;
}
```

## KUTXA TXURIKO PROBAK

Fluxu diagrama:



Konplexutasun ziklomatikoa:

Metodo honen konplexutasun ziklomatikoa 4 da, erabaki nodoak(3) +1 eginda kalkulatua.

Bide posibleak:

- 1- FOR-1(F), end
- 2- FOR-1(T), 1, IF-1(F), FOR-1(F), end
- 3- FOR-1(T), 1, IF-1(T), IF-2(F), FOR-1(F), end
- 4- FOR-1(T), 1, IF-1(T), IF-2(T), 2, end

#	Baldintza	Proba kontextua		Itxaron emaitzak	
		DB egoera	Sarrera	DB egoera	Irteera
1	1	pr.apustuak = []	(qu, pr)	Ez da aldatzen	[0,0]
2	2	pr.apustuak = [ap1]	(qu, pr)	Ez da aldatzen	[1,0]
3	3	pro.apustuak.eguneratuAsmatutakoKop()==true (Horretarako: ap**.pronostikokop=1, ap.asmatutakokop=0) eta ap.kopurua = 5, ap.kuotatotala = 5	(qu, pr)	bez.mugimeduak = [["Apustua irabazi ap.identifikadorea", 25.0, "irabazi"]]	[*,25]
4	4	pro.apustuak.eguneratuAsmatutakoKop()==true(Horretarako: ap.pronostikokop=1, ap.asmatutakokop=0) eta ap.kopurua = 5, ap.kuotatotala = 5 eta err*** = Errepikapena(bez****, noriBez, 20,20,2), bez.errepikatua(err), ap.errepikatua(bez)	(qu, pr)	bez.mugimeduak = [["Apustu errepikatuaren komisioa(bezeroa.izena) 10 irabazi"], ["Apustua irabazi a.identifikadorea", 5.0, "irabazi"]]	[*,5]

\* Sarrera guztiak berdinak dira, aldatzen dena hauen DB-ko atributuen egoera da, metodoak hori hartzen baitu kontuan.

\*\* DB-an dagoen pronostikoari lotutako apustua

\*\*\* DB-an dagoen Errepikapena motako objektua

\*\*\*\* DB-an dagoen Bezera motako objektua, errepikapenera eta apustura lotutakoa.

## KUTXA BELTZEKO PROBAK

Baliokidetasun klaseak:

<b>Sarrera-baldintza</b>	<b>Baliokidetasun klase egokiak</b>	<b>Baliokidetasun klase ez-egokiak</b>
Question balioa	Question!= null (1)	Question== null (5)
Pronostikoa balioa	Pronostikoa != null (2)	Pronostikoa ==null (6)
Question DB-an	Question $\in$ DB(3)	Question $\notin$ DB (7)
Pronostikoa DB-an	Pronostikoa $\in$ DB (4)	Pronostikoa $\notin$ DB (8)

Proba taula:

#	<b>Estalitako BK</b>	<b>Proba kontextua</b>		<b>Esperotako emaitzak</b>	
		<b>DB egoera</b>	<b>Sarrera</b>	<b>DB egoera</b>	<b>Irteera</b>
1	1,2,3,4	qu,pr $\in$ DB	("Zeinek irabaziko du?","Real Sociedad")	qu' $\in$ DB (Galdera pronostikoarekin)	*
2	5,6	*	(null, null)	*	Errorea
3	7,8	qu,pr $\notin$ DB	*	qu,pr $\notin$ DB	Errorea

## INTEGRAZIO PROBAK

Integrazio probetan inplementatutako proba kasuak kutxa beltzeko proba kasu berdinak dira.



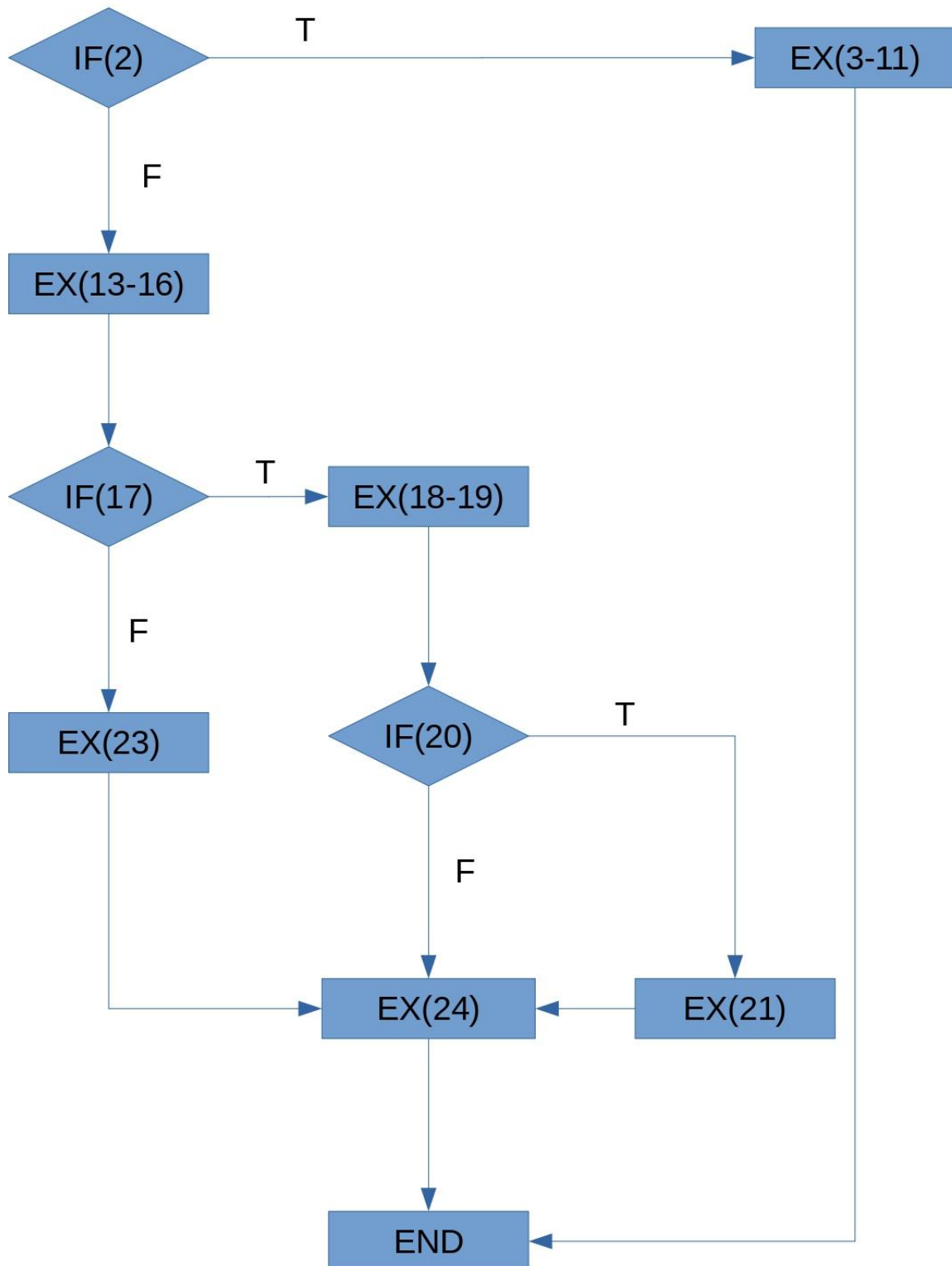
## removeMezua() - Zdravko Todorov

Kodea:

```
1. public void removeMezua(Mezua mezua) {
2.     if(mezua instanceof BezzeroartekoMezua) {
3.         BezzeroartekoMezua m = db.find(BezzeroartekoMezua.class,
4.                                         mezua.getIdentifikadorea());
5.         Bezeroa nork = m.getIgorlea();
6.         Bezeroa nori = m.getHartzailea();
7.         db.getTransaction().begin();
8.         nork.ezabatuBidalitakoBezzeroMezua(m);
9.         nori.ezabatuJasotakoBezzeroMezua(m);
10.        db.remove(m);
11.        db.getTransaction().commit();
12.    }else {
13.        ArretaMezua m = db.find(ArretaMezua.class,
14.                                mezua.getIdentifikadorea());
15.        ArretaElkarrizketa elkarrizketa = m.getElkarrizketa();
16.        db.getTransaction().begin();
17.        if(elkarrizketa.isAmaituta()) {
18.            elkarrizketa.removeMezua(m);
19.            db.remove(m);
20.            if(elkarrizketa.mezurikEz()) {
21.                db.remove(elkarrizketa); }
22.        }else {
23.            m.setIkusgaiBezzeroarentzat(false); }
24.        db.getTransaction().commit();
25.    }
26. }
```

# Kutxa Txuriko Probak

*Fluxu Diagrama:*



**Komplexutasun ziklotatikoa 4 da, erabaki nodoak + 1 eginez, kalkulatua.**

## Bide posibleak

- 1- IF-2 (T), EX(3-11), END
- 2- IF-2 (F), EX(13-16), IF-17 (F), EX(23), EX(24), END
- 3- IF-2 (F), EX(13-16), IF-17 (T), EX(18-19), IF-20 (T), EX(21), EX(24), END
- 4- IF-2 (F), EX(13-16), IF-17 (T), EX(18-19), IF-20 (F), EX(24), END

## Proba taula

#	Baldintza	Proba kontextua		Itxaron emaitzak	
		DB egoera	Sarrera	DB egoera	Irteera
1	1	$m \in DB$	$m$ (Bezeroarteko Mezua)	$m \notin DB$	---
2	2	$(m \in DB) \wedge$ $(m.getElkarrizketa().isAmaituta() == false)$	$m$ (ArretaMezua)	$(m \in DB) \wedge$ $(m.ikusgaiBezeroarentzat == false)$	---
3	3	$(m \in DB) \wedge$ $(m.getElkarrizketa().isAmaituta() == true) \wedge$ $(m.getElkarrizketa().mezuKop == 1)$	$m$ (ArretaMezua)	$(m \notin DB) \wedge$ $(m.getElkarrizketa() \notin DB)$	---
4	4	$(m \in DB) \wedge$ $(m.getElkarrizketa().isAmaituta() == true) \wedge$ $(m.getElkarrizketa().mezuKop > 1)$	$m$ (ArretaMezua)	$(m \notin DB) \wedge$ $(m.getElkarrizketa() \in DB)$	---

\* Sarrerako balioa beti Mezua klaseko objektu bat izango da, baina klase desberdin bateko instantzia bat, kasu ezberdinetan.

## Implementazioa: RemoveMezuaDAW.java

### Aurkitutako akatsak

Esan beharra dago, kodea gurea ez zenez, goitik behera analizatu beharra izan dugula, egoki nola funtzionatzen duen ulertzeko. Esfortzu guztiak egin arren, ezin izan ditugu DB-arekin egon diren hainbat portaera harraro zuzendu. Kutxa txuriaren kasuan, mezuak ezabatzen ziren eta "ikusgaiBezeroentzat" aldagaia aldatzen zen, eta hau ObjectDB-ren explorer tresnaz baliatuta ikusi dugu, baina ondoren gure kodetik bere balioa atzitzerakoan, beti hasierako balioa zuen. Beraz, JUniten portaerak erakusten du funtzioak ez duela egoki funtzionatzen, baina benetan ez da horrela. Edozein momentutan, funtzioak guk nahi genuena egin du.

# Kutxa Beltzeko Probak

## Baliokidetasun klaseak:

<i>Sarrera-baldintza</i>	<i>Baliokidetasun klase egokiak</i>	<i>Baliokidetasun klase ez-egokiak</i>
Mezua DB-an dago	mezua ∈ DB (1)	mezua ∉ DB (2)
BezeroartekoMezua da	mezua instanceof BezeroartekoMezua == true (3)	---
ArretaMezua da	mezua instanceof ArretaMezua == true (4)	---
Mezua ez da null	mezua != null (5)	mezua == null (6)

## Probak:

#	<i>Estalitako BK</i>	<i>Proba kontextua</i>		<i>Esperotako emaitzak</i>	
		<i>DB egoera</i>	<i>Sarrera</i>	<i>DB egoera</i>	<i>Irteera</i>
1	1, 3, 5	mezua ∈ DB	mezua (BezeroartekoMezua, "Proba mezua")	mezua ∉ DB	---
2	1, 4, 5	mezua ∈ DB	mezua (ArretaMezua, "Proba mezua")	mezua ∉ DB	---
3	2, 6	mezua ∉ DB	null	mezua ∉ DB	Errorea

Implementazioa: RemoveMezuaDABjava

## Aurkitutako akatsak

Kutxa txuriaren atalean aipatutakoaren ildoan, hemen ere ez ditugu esperatako emaitzak jaso kasu batzuetan, baina berez DB-kode arteko elkarrekintza zuzena izan da.

## Integrazio Probak

Integrazio probetan inplementatutako proba kasuak kutxa beltzeko proba kasu berdinak dira.

Implementazioa: RemoveMezuaMockInt.java