

# BetANDRuin PROBAK



David: 17 ordu

Alaia: 18 ordu

Andoni:

# ESTEKAK:

Github: <https://github.com/DavidMartin112/BetANDRuinSI2>

Sonarcloud.io:

[https://sonarcloud.io/project/overview?id=DavidMartin112\\_BetANDRuinSI2](https://sonarcloud.io/project/overview?id=DavidMartin112_BetANDRuinSI2)

## SONARCLOUD Zuzenketak:

David

- Replace this if-then-else statement by a single method invocation

```
}  
if(modelQuotes.getSize()==0) {  
    btnApustuaGehitu.setEnabled(false);  
}else {  
    btnApustuaGehitu.setEnabled(true);  
}  
}
```

```
bel jLabelDiruKopurua = new JLabel();  
belDiruKopurua.setFont(new Font("Tahoma", Font.PLAIN, 18));
```

163 items		
	Date	Description
UI.java		Replace the type specification in this constructor call with the diamond
UI.java		Replace the type specification in this constructor call with the diamond
UI.java		Replace the type specification in this constructor call with the diamond
UI.java		Replace the type specification in this constructor call with the diamond
UI.java		Replace this if-then-else statement by a single method invocation.
UI.java		Replace this if-then-else statement by a single method invocation.

Goian ikusi dezakegun if else aldatu eta ondorengo jarri dut:

```
btnApustuaGehitu.setEnabled(modelQuotes.getSize()==0);
```

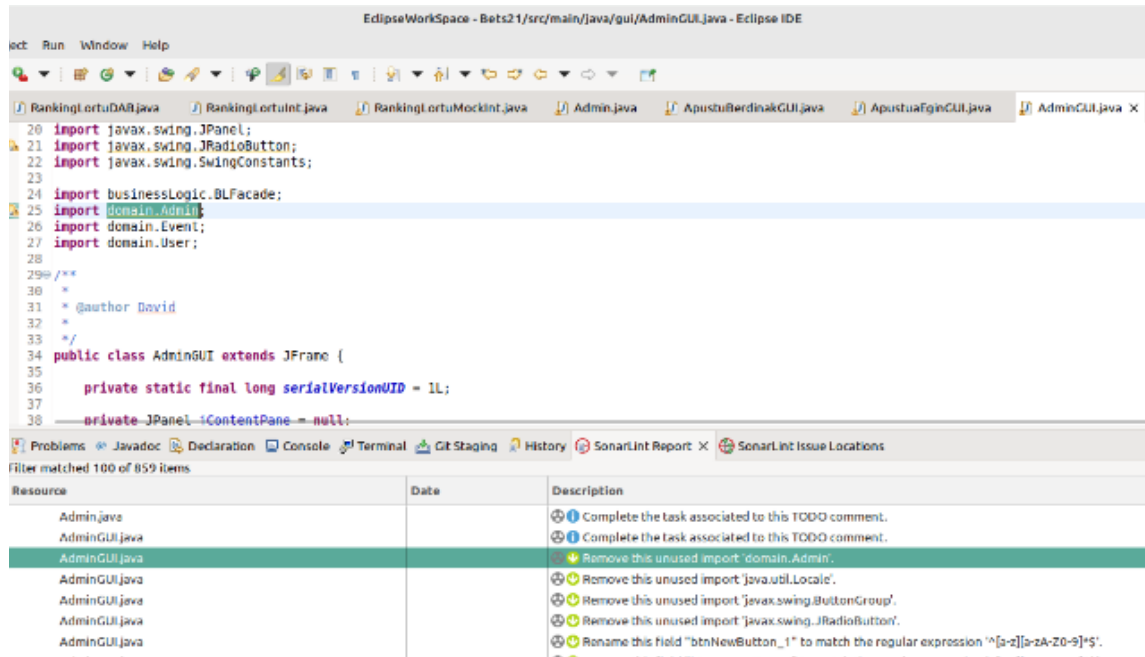
- Replace this if-then-else statement by a single method invocation

hed 100 of 862 items		
	Date	Description
pustuaEginGUI.java		replace the type specification in this constructor call with the diamond
pustuaEginGUI.java		Replace the type specification in this constructor call with the diamond
pustuaEginGUI.java		Replace the type specification in this constructor call with the diamond
pustuaEginGUI.java		Replace the type specification in this constructor call with the diamond
pustuaEginGUI.java		Replace this if-then-else statement by a single method invocation.
pustuaEginGUI.java		Replace this if-then-else statement by a single method invocation.
pustuaEginGUI.java		The type of "datesWithEventsCurrentMonth2" should be an interface su
pustuaEginGUI.java		Use isEmpty() to check whether the collection is empty or not

Goian ikusi dezakegun if else aldatu eta ondorengoa jarri dut:

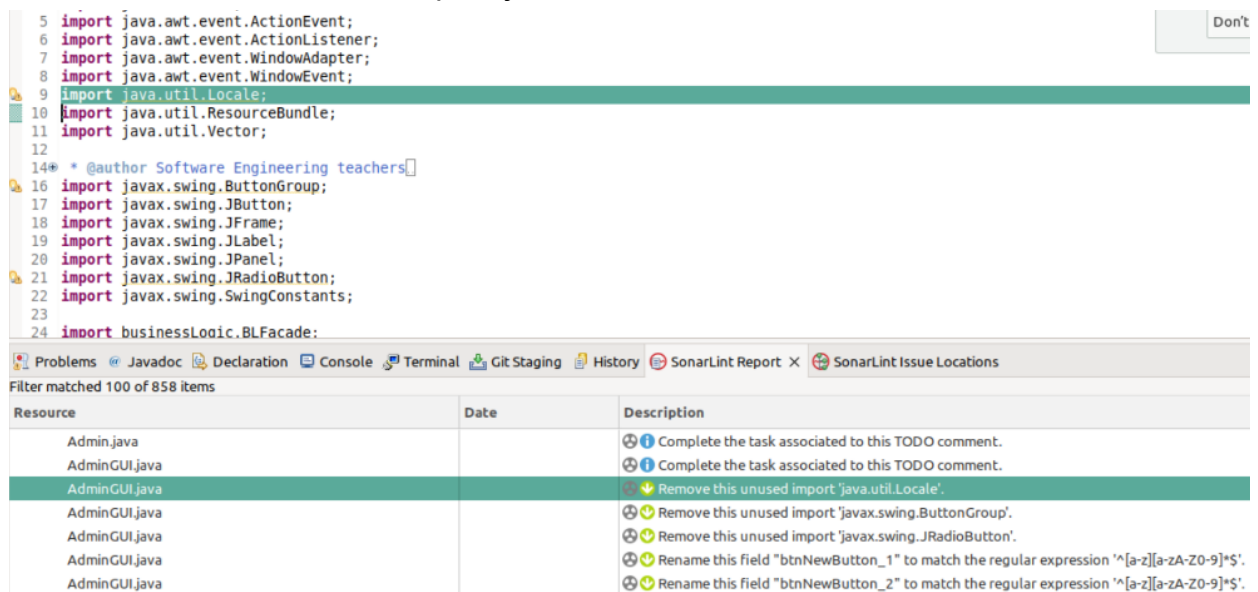
```
btnApustuaGehitu.setEnabled(modelQuotes.getSize()>0);
```

- Remove this unused import 'domain.Admin'



Kasu honetan egin dudana sinplea da. Importa berratu.

- Remove this unused import 'java.util.locale'



Aurreko 'code smell'ean egindako berdina.

- Remove this unused method parameter “e”

```

475     }
476 }
477
478 private void jButtonClose.actionPerformed() {
479     this.setVisible(false);
480 }
481 }
482

```

3.mdj

Problems Javadoc Declaration Console Terminal Git Staging History SonarLint Report X SonarLint Issue Locations

Filter matched 100 of 863 items



Resource	Date	Description
ApustuaEginGUI.java		Replace this if-then-else statement by a single method invocation.
ApustuaEginGUI.java		The type of "datesWithEventsCurrentMonth2" should be an inter
ApustuaEginGUI.java		Use isEmpty() to check whether the collection is empty or not.
ApustuaEginGUI.java		Use the primitive boolean expression here.
ApustuaEginGUI.java		Define and throw a dedicated exception instead of using a generi
ApustuaEginGUI.java		Make this anonymous inner class a lambda
ApustuaEginGUI.java		Make this anonymous inner class a lambda
ApustuaEginGUI.java		Make this anonymous inner class a lambda
ApustuaEginGUI.java		Make this anonymous inner class a lambda
ApustuaEginGUI.java		Make this anonymous inner class a lambda
ApustuaEginGUI.java		Make this anonymous inner class a lambda
ApustuaEginGUI.java		Provide the parametrized type for this generic.
ApustuaEginGUI.java		Provide the parametrized type for this generic.
ApustuaEginGUI.java		Remove this unused method parameter "e".
ApustuaEginGUI.java		Remove this unused method parameter "v".
ApustuaEginGUI.java		Replace the synchronized class "Vector" by an unsynchronized on
ApustuaEginGUI.java		Replace the synchronized class "Vector" by an unsynchronized on




uinSI2  
oIy/.git

Ikusten da argazkian parametroa ezabatu dudala. Kontuan izan nuen ezabatzea funtzio honi egindako dei guztietan ezabatzea ere bai.

## Alaia

```
1 Jarraitzailea erab=db.find(Jarraitzailea.class, reg.getJarraitzaileaNumber());
2 b=true;
3 for(ApustuAnitza apu: erab.getNork().getApustuAnitzak()) {
4     if(apu.getApustuKopia()==apustuAnitza.getApustuKopia()) {
```

 Strings and Boxed types should be compared using "equals()". [Why is this an issue?](#) 24 days ago ∨ L1024 



 Bug  Major  Open Not assigned 5min effort No t



```
5         b=false;
6     }
7 }
```

Kasu honetan aldatu behar izandako kodea if horretako ==-a equals batekin ordezkatu:

```
if(apu.getApustuKopia().equals(apustuAnitza.getApustuKopia()))
```

```
16 import domain.Message;
17 import domain.MezuakContainer;
18 //import domain.Booking;
```



 This block of commented-out lines of code should be removed. [Why is this an issue?](#) 24 days ago ∨ L18 




 Code Smell  Major  Open Not assigned 5min effort No t

```
19 import domain.Question;
20 import domain.Quote;
```



Komentatua dagoen import hori ezabatu det.




```
6 */
7 @WebMethod public Vector<Event> getEvents(Date date);
```

 Replace the synchronized class "Vector" by an unsynchronized one such as "ArrayList" or "LinkedList". [Why is this an issue?](#) 24 days ago ∨ L57 

 Code Smell  Major  Open Not assigned 20min effort No t

```
8
9 /**
0  * This method retrieves from the database the dates a month for which there are events
1  *
2  * @param date of the month for which days with events want to be retrieved
3  * @return collection of dates
4  */
5 @WebMethod public Vector<Date> getEventsMonth(Date date);
```

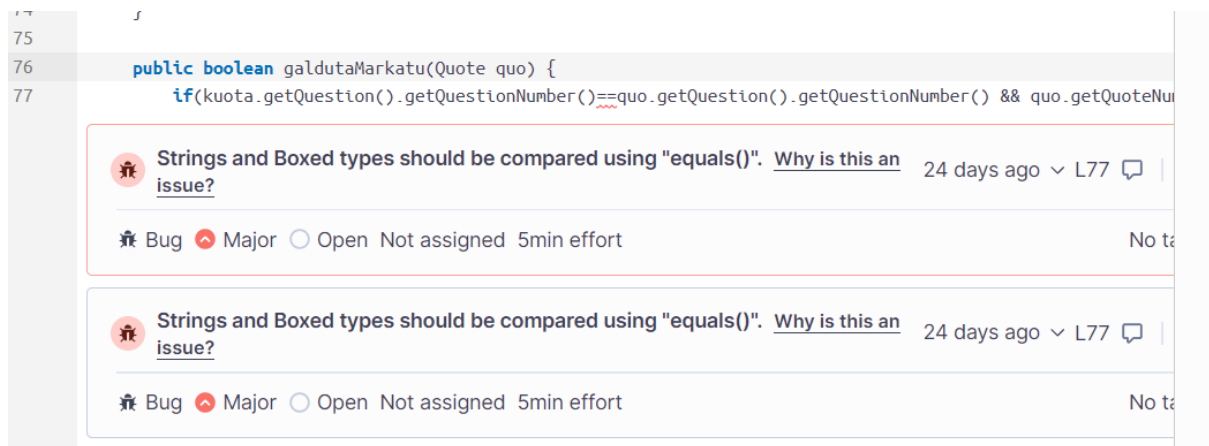
 Replace the synchronized class "Vector" by an unsynchronized one such as "ArrayList" or "LinkedList". [Why is this an issue?](#) 24 days ago ∨ L65 

 Code Smell  Major  Open Not assigned 20min effort No t

Bi hauetan Vector horiek ArrayList batekin aldatu ditut.

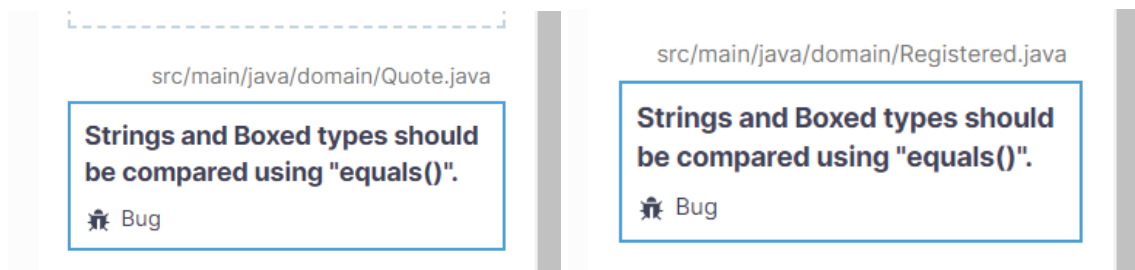
```
public ArrayList<Event> getEvents(Date date)
public ArrayList<Date> getEventsMonth(Date date)
```

Kasu honetan Interfazetik aldatu eta gero metodoen implementazioa aldatu behar izan det Vector<Event> eta Vector<Date> guztiak ArrayListekin ordezkatzeko.

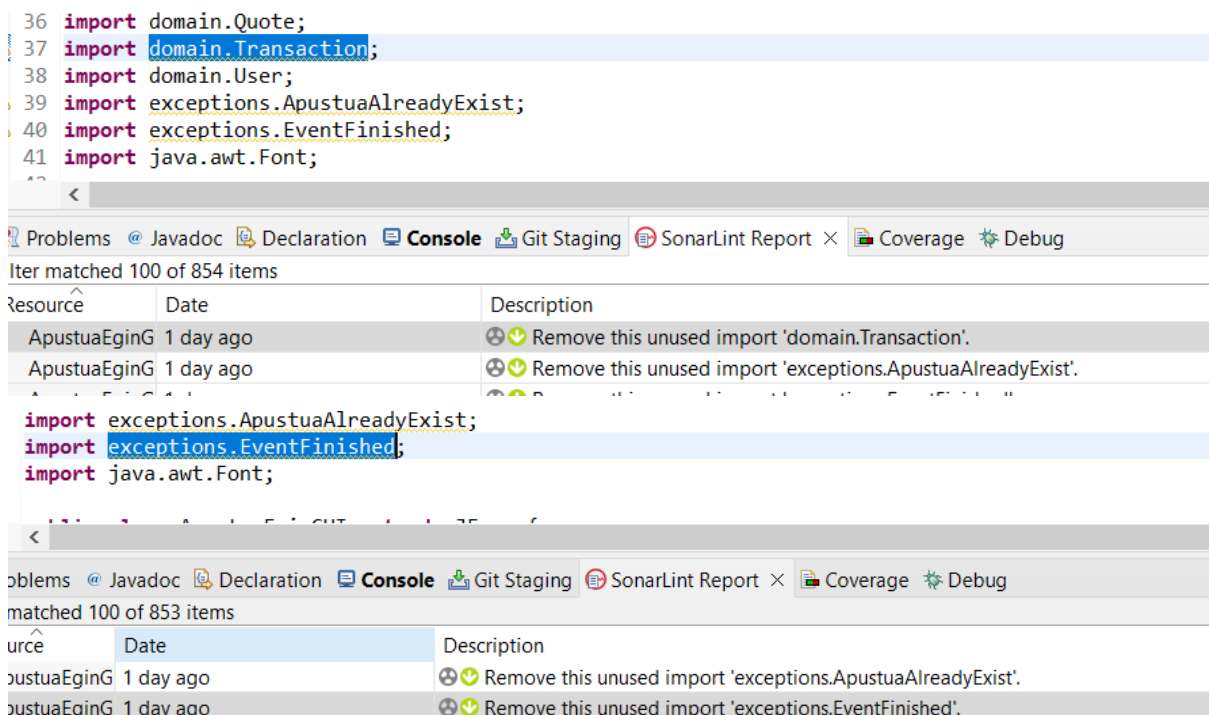


Lehenengo arazoaren berdina, ==-ak equals metodoarekin ordezkatu:

```
if(kuota.getQuestion().getQuestionNumber.equals(quo.getQuestion().getQuestionNumber())
    && quo.getQuestion().getQuestionNumber() &&
    !quo.getQuoteNumber().equals(kuota.getQuoteNumber()))
```



Goikoaren berdina.



```
import domain.Transaction;
import exceptions.ApustuaAlreadyExist;
import exceptions.EventFinished;
Goiko import hauek ezabatu.
```

# Andoni

- Indentation of **if** statement:

```
        if(new Date().compareTo(eventDate)>0) {
            throw new EventFinished("Gertaera honen data dagoeneko pasa da");
        }
    }
}
```

- Avoid repetition of “initialize” string, declare a constant that follows naming conventions(`^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$`):

```
@WebService(endpointInterface = "businessLogic.BLFacade")
public class BLFacadeImplementation implements BLFacade {
    DataAccess dbManager;
    final static String INIT="initialize";
    public BLFacadeImplementation() {
        System.out.println("Creating BLFacadeImplementation instance");
        ConfigXML c=ConfigXML.getInstance();
        |
        if (c.getDataBaseOpenMode().equals(INIT)) {
            dbManager=new DataAccess(c.getDataBaseOpenMode().equals(INIT));
            dbManager.initializeDB();
        } else
            dbManager=new DataAccess();
        dbManager.close();

    }

    public BLFacadeImplementation(DataAccess da) {

        System.out.println("Creating BLFacadeImplementation instance with DataAccess parameter");
        ConfigXML c=ConfigXML.getInstance();

        if (c.getDataBaseOpenMode().equals(INIT)) {
            da.open(true);
        }
    }
}
```

- Avoid repetition of numerous strings by declaring constants:

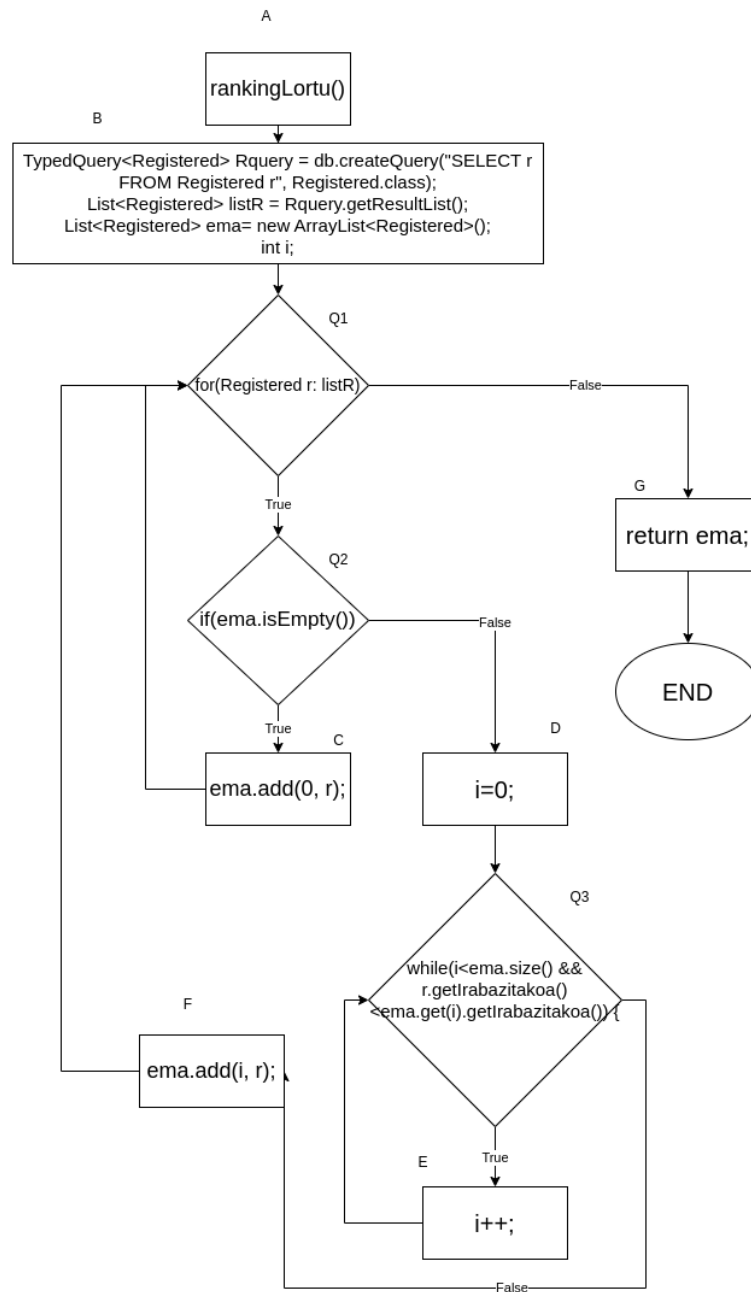
```
final String winnerQuestionES="¿Quién ganará el partido?";
final String winnerQuestionENG="Who will win the match?";
final String winnerQuestionEUS="Zeinek irabaziko du partidua?";

if (Locale.getDefault().equals(new Locale("es"))) {
    q1=ev1.addQuestion(winnerQuestionES,1);
    q2=ev1.addQuestion("¿Quién meterá el primer gol?",2);
    q3=ev11.addQuestion(winnerQuestionES,1);
    q4=ev11.addQuestion("¿Cuántos goles se marcarán?",2);
    q5=ev17.addQuestion(winnerQuestionES,1);
    q6=ev17.addQuestion("¿Habrá goles en la primera parte?",2);
}
else if (Locale.getDefault().equals(new Locale("en"))) {
    q1=ev1.addQuestion(winnerQuestionENG,1);
    q2=ev1.addQuestion("Who will score first?",2);
    q3=ev11.addQuestion(winnerQuestionENG,1);
    q4=ev11.addQuestion("How many goals will be scored in the match?",2);
    q5=ev17.addQuestion(winnerQuestionENG,1);
    q6=ev17.addQuestion("Will there be goals in the first half?",2);
}
else {
    q1=ev1.addQuestion(winnerQuestionEUS,1);
    q2=ev1.addQuestion("Zeinek sartuko du lehenengo gola?",2);
    q3=ev11.addQuestion(winnerQuestionEUS,1);
    q4=ev11.addQuestion("Zenbat gol sartuko dira?",2);
    q5=ev17.addQuestion(winnerQuestionEUS,1);
    q6=ev17.addQuestion("Golak sartuko dira lehenengo zatian?",2);
}
```

# JUNIT eta MOCKITO metodoak

David: rankingLortu()

## Kutxa txuria



4 KONPLEXUTASUN ZIKLOMATIKOA



Proba-kasuen taula:

Kasua	Bidea	Sarrera Baldintza	Sarrera	Irteera
1	A-B-Q1(f)-G-END ND	listR.isEmpty()	listR = [ ]	ema = [ ]
2	A-B-Q1(T)-Q2(T)-C-G-END	!listR.isEmpty() && ema.isEmpty()	listR = [david,] ema = [ ]	ema = [david]
3	A-B-Q1(T)-Q2(F)-D-Q3(F)-F-G-END	!listR.isEmpty() && !ema.isEmpty() && !(i<ema.size() && r.getIrabazitakoa()<ema .get(i).getIrabazitakoa())	listR = [A, B, C, D, E], ema = [B,C,D,A]	ema = [E,B,C,D,A]
4	A-B-Q1(T)-Q2(F)-D-Q3(T)-E-F-G-END	!listR.isEmpty() && !ema.isEmpty() && (i<ema.size() && r.getIrabazitakoa()<ema .get(i).getIrabazitakoa())	listR = [A, B, C ,], ema = [A, B]	ema = [A,C,B]

Lau proba kasu hauek junitekin aplikatzeko arazorik ez nuen kodean aurkitu.

## Kutxa Beltza

### BALIOKIDETASUN-KLASEAREN EMAITZA-TAULA

Baldintza	Baliokidetasun-klase egokiak	Baliokidetasun-klase ez egokiak
SARRERARO PARAMETROAK		
PROGRAMAREN PORTAERA		
DB-an ez dago "registered user"	DB hutsa (1)	_____
DB-an "registered user" bat	DB-an registered.size()==1 (2)	_____
Hainbat "registered user"	DB-an registered user()>1 (3)	_____
Irabazitakoa berdina	DB-an getIrabazitakoa berdinak (4)	_____
Irabazi kopuru desberdina	DB-an getIrabazitakoa != (5)	_____

### PROBA KASUEN ERATORPENA

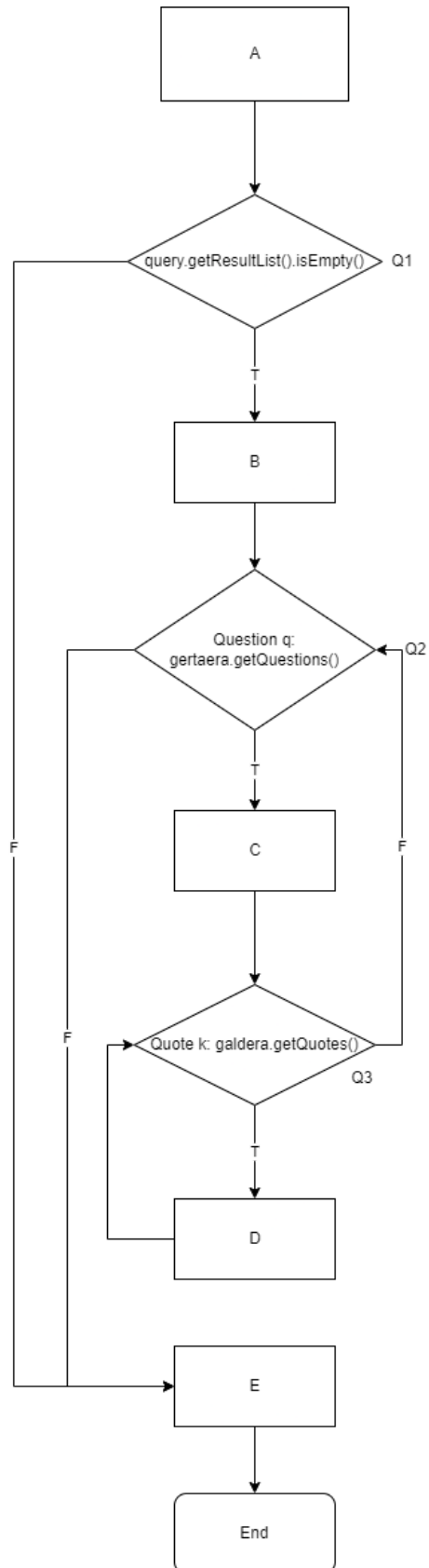
Sarrera	DB egoera	Estalitako baliokidetasunklaseak	Emitza
_____	registered.isEmpty() = true	1	[]
_____	registered.size() = 1 us1 = david, 15\$	2	[us1]
_____	registered.size() = 1 us1 = david, 1\$ us2 = manuel, 1\$ us3 = pedro, 1\$ us4 = pepito, 1\$ us5 = andoni, 1\$	2, 3, 4	[us1, us2, us3, us4, us5]
_____	registered.size() = 1 us1 = david, 24\$ us2 = manuel, 13\$ us3 = pedro, 16\$	2, 3, 5	[us1, us3, us2]

Kutxa beltzeko proba-kasu hauek erabili ditut dataAccessa testatzeko eta integraazio probak(BL eta Mockito testak) egiteko. Ez dagoelako desberdintasun nabaririk BL eta dataAccess barruko metodoarekin. Arazorik ez testetan, guztiak pasatzen dute.

## Alaia: gertaerakKopiatu()

public boolean gertaerakKopiatu(Event e, Date date)

### Kutxa txuria



Konplexutasun ziklomatikoa= 4

- $KZ = \# \text{erabaki nodo} + 1 = 3 + 1 = 4$
- $KZ = \text{ingurune itxi kopuru} + 1 = 3 + 1 = 4$

## **PROBA-KASUEN TAULA**

### **Sarrerako datuak:**

Event E1, Date D1 -> `!query.getResultList().isEmpty()`

Event E1, Date D2 -> `query.getResultList().isEmpty() && E1 ez dauka galderarik (gertaera.getQuestions().isEmpty())`

Event E2, Date D1 --> `query.getResultList().isEmpty() && E2 galderak dauzka (!gertaera.getQuestions().isEmpty()) && lehen galderak ez du kuotarik (galdera.getQuotes().isEmpty())`

Event E2, Date D2 -> `query.getResultList().isEmpty() && E2 galderak dauzka (!gertaera.getQuestions().isEmpty()) && lehen galderak kuotak ditu`

Kasua	Bidea	Sarrera baldintza	Sarrera	Irteera
1	A-Q1(F)-E-End	<code>!query.getResultList().isEmpty()</code>	E1,D1	FALSE
2	A-Q1(T)-B-Q2(F)-E-End	<code>query.getResultList().isEmpty() &amp;&amp; gertaera.getQuestions().isEmpty()</code>	E1,D2	TRUE
3	A-Q1(T)-B-Q2(T)-C-Q3(F)-E-End	<code>query.getResultList().isEmpty() &amp;&amp; !gertaera.getQuestions().isEmpty() &amp;&amp; galdera.getQuotes().isEmpty()</code>	E2,D1	TRUE
4	A-Q1(T)-B-Q2(T)-C-Q3(T)-D-E-End	<code>query.getResultList().isEmpty() &amp;&amp; !gertaera.getQuestions().isEmpty() &amp;&amp; !galdera.getQuotes().isEmpty()</code>	E2,D2	TRUE

Proba kasu hauek programatzean ez dut arazorik izan metodoarekin, ondo implementatua baitago. Database batekin JUnit-ak kodetzean izan dut arazo gehiena.

## **Kutxa Beltza**

### **BALIOKIDETASUN-KLASEAREN EMAITZA-TAULA**

Metodoak egiten duena da, jadanik existitzen den eventua bat hartu eta kopiatu data aldatuz.

### **BALIOKIDETASUN-KLASEAREN EMAITZA-TAULA**

Metodoak ez du inoiz salbuespenik altxatzen, beraz sarrera parametroak beti egokiak izango direla suposatuko dut. Hau da, event-a datubasean egongo dela data kontuan izan gabe.

Baldintza	Baliokidetasun-klase egokiak	Baliokidetasun-klase ez egokiak
Event-a ez datubasean dago	$\text{Event} \in \text{DB} \text{ (1)}$	$\text{Event} \notin \text{DB} \text{ (2)}$

Event-a existitzen da data horretarako	EventKopiatua $\in$ DB (3)	-----
Event-a ez da existitzen data horretarako	EventKopiatua $\notin$ DB (4)	-----
Eventuak galderak ditu	EventKopiatua.getQuestions() != null (5)	-----
Eventuak ez ditu galderarik	EventKopiatua.getQuestions() = null (6)	-----
Eventuaren galderak kuotak ditu	galdera.getQuotes() != null (7)	-----
Eventuaren galderak ez du kuotarik	galdera.getQuotes() = null (8)	-----

### **PROBA KASUAK: STATEFULL**

Parametroak: E eventuentzako, EK eventu kopiatuentzako

G izango dute gertaeradun eventu eta eventu kopiatuek (EG, EKG)

C izango dute kuotaren bat duten eventuek (EGC, EKGK)

Adibidea: Datubasean EG badagoen, E eventua eta G galderak datubasean gordeta daude.

#	Estalitako klaseak	Sarrera kontestua		Irteera kontestua	
		DB egoera	Parametroak	DB egoera	Irteera
1	2		E, 15/9/2022		false
2	1,3,6,8	E(08/9/2022)	E, 08/9/2022	E(08/9/2022)	false
3	1,4,5,7	EGC(08/9/2022)	EGC, 15/10/2022	EGC(08/9/2022), EKGK(15,10/2022)	true
4	1,4,5,8	EG(08/9/2022)	EG, 15/10/2022	EG(08/9/2022), EKG(15,10/2022)	true
5	1,4,6,8	E(08/9/2022)	E, 15/10/2022	E(08/9/2022), EK(15,10/2022)	true

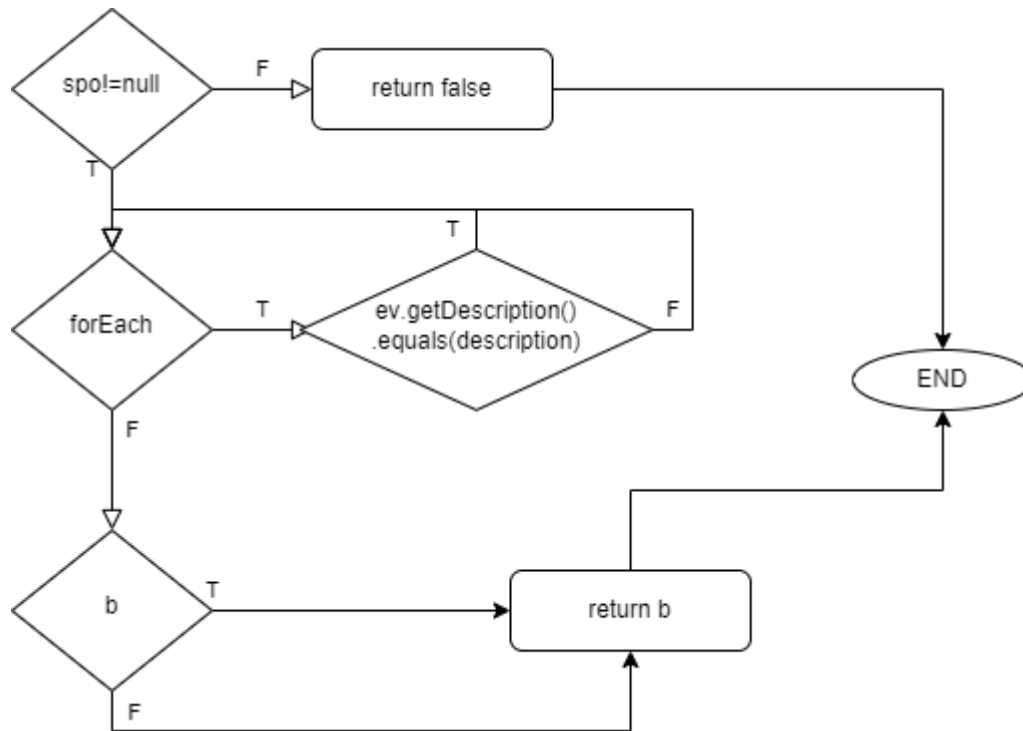
### **MUGA BALIOEN PROBAK**

Ezin da muga proben balioa egin, ez dagoelako zer aztertu.

Ez dut arazorik izan proba-kasu hauek implementatzean eta proba-kasu berdinak erabili ditut integrazio probak sortzeko. Azkenean, BL-k egiten duena DataAccess-eko metodoari deitzea da, beraz proba-kasuak berdinak dira. Hauek ere, arazo barik.

## Andoni: gertaerakSortu()

### KUTXA TXURIA



Kasua	Sarrera baldintza	Sarrera	bidea	irteera
1	Sport $\notin$ DB	("FCB-RMD", 01/01 /2002, "Petanca")	A/F	false
2	Description $\in$ D B	("FCB-RMD", 01/01 /2002, "Futbol")	A/T-B/T-C/T- B/F-D/F	false
3	Description $\notin$ DB	("FCB-RMD", 01/01 /2002, "Futbol")	A/T-B/T-C/F- B/F-D/T	true

# GERTAERAK SORTU KUTXA BELTZA

public boolean gertaerakSortu(String description, Date eventDate, String sport)

Suposatzen dut sarrera parametroak beti direla egokiak ez baitu salbuespenik altxatzen.  
Dokumentazio ikusita programak hurreneko portaera duela uste dut.

## BALIOKIDETASUN-KLASEAREN EMAITZA-TAULA

Baldintza	Baliokidetasun-klase egokiak	Baliokidetasun-klase ez egokiak
Kirola ez dago datu basean	Sport $\notin$ DB(1)	-----
Kirola datu basean dago	Sport $\in$ DB(2)	-----
description datu basean dago, baina ez data horrekin	Description $\in$ DB, Date $\notin$ DB(3)	-----
description datu basean dago, data zuzenarekin	Description $\in$ DB, Date $\in$ DB(4)	-----
description ez dago datu basean	Description $\notin$ DB(5)	-----

## PROBA KASUEN ERATORPENA

Sarrera	DB egoera	Estalitako baliokidetasun-klase(ak)	Emitza
("FCB-RMD", 01/01/2002, "Petanca")	("FCB-RMD", 01/01/2002, "Futbol")	1	false
("FCB-RMD", 01/01/2002, "Futbol")	("FCB-RMD", 01/01/2002, "Futbol")	2,4	false
("FCB-RMD", 01/01/2002, "Futbol")	("FCB-RMD", 01/08/2002, "Futbol")	2,3	true
("FCB-RSO", 01/01/2002, "Futbol")	("FCB-RMD", 01/01/2002, "Futbol")	2,5	true

Ezin da muga balioen azterketa egin, ez baitago zer aztertu

Bi test motetan erroreak izan ditut, ez baitu inoiz ez true balioa bueltatzen. Badirudi errorea datubasearen hasieraketan dagoela, izan ere, hau true balioarekin irekitzean, *open(boolean)* metodoarekin, datu base zaharra ezabatu beharko luke eta ez du egiten.

## GERTAERAK SORTU(BL) KUTXA BELTZA

```
public boolean gertaerakSortu(String description,Date eventDate, String sport)
```

### BALIOKIDETASUN-KLASEAREN EMAITZA-TAULA

Baldintza	Baliokidetasun-klase egokiak	Baliokidetasun-klase ez egokiak
Iraganeko data	Future date(1)	Past date(2)
Kirola ez dago datu basean	Sport $\notin$ DB(3)	-----
Kirola datu basean dago	Sport $\in$ DB(4)	-----
description datu basean dago, baina ez data horrekin	Description $\in$ DB, Date $\notin$ DB(5)	-----
description datu basean dago, data zuzenarekin	Description $\in$ DB, Date $\in$ DB(6)	-----
description ez dago datu basean	Description $\notin$ DB(7)	



## PROBA KASUEN ERATORPENA

Sarrera	DB egoera	Estalitako baliokidetasunklase(ak)	Emitza
("FCB-RMD",01/01/2022, "Futbol")	-----	2	Exception
("FCB-RMD",01/01/2023, "Petanca")	("FCB-RMD",01/01/2023, "Futbol")	1,3	false
("FCB-RMD",01/01/2023, "Futbol")	("FCB-RMD",01/01/2023, "Futbol")	1,4,6	false
("FCB-RMD",01/01/2023, "Futbol")	("FCB-RMD",01/08/2023, "Futbol")	1,4,5	true
("FCB-RSO",01/01/2023, "Futbol")	("FCB-RMD",01/01/2023, "Futbol")	1,4,7	true

Ezin da muga balioen azterketa egin, ez baitago zer arteztu

Metodo honek ez du arazorik eman DataAcces klaseko metodoa ondo funtzionatzeko duela simulatu dugunean.