

## "Write short units of code" (2.kapitula)

### DAVID:

.DataAccess clasearen barruan gertaerakSortu metodoa oso code luzea da. 15 code baino gutxiago izan behar duenez. Refactor egin dut refactor > extract method erabiliz for loopa ateratzen eta honela sinpleagoa da codea. Hemen azpian errefaktORIZAZIO preview:

Changes to be performed


- ✓ DataAccess.java - Bets21/src/main/java/dataAccess
- ✓ DataAccess
  - ✓ gertaerakSortu(String, Date, String)
  - ✓ Substitute statements with call to isEventInDB
  - ✓ Create new method 'isEventInDB' from selected statements

DataAccess.java




Original Source	Refactored Source
882 883     uu.persist(ad); 884     db.getTransaction().commit(); 885 } 886 887 public boolean gertaerakSortu(String description, Date eventDate, String sport) { 888     boolean b = true; 889     db.getTransaction().begin(); 890     Sport spo = db.find(Sport.class, sport); 891     if(spo==null) { 892         TypedQuery<Event> Equery = db.createQuery("SELECT e FROM Event e WHERE e.getEventDate() =?1 ",Event.class); 893         Equery.setParameter(1, eventDate); 894         for(Event ev: Equery.getResultList()) { 895             if(ev.getDescription().equals(description)) { 896                 b = false; 897             } 898         } 899         if(b) { 900             String[] taldeak = description.split("-"); 901             Team lokala = new Team(taldeak[0]); 902             Team kanpokoa = new Team(taldeak[1]); 903             Event e = new Event(description, eventDate, lokala, kanpokoa); 904             e.setSport(spo); 905             spo.addEvent(e); 906             db.persist(e); 907         } 908         return false; 909         } 910         db.getTransaction().commit(); 911         return b; 912     } 913 } 914 915 public Quote storeQuote(String forecast, Double Quote, Question question) throws QuoteAlreadyExist { 916     //System.out.println(">>> DataAccess: createQuestion=> event= "+event+" question= "+question+" betMinimum 917     Question q = db.find(Question.class, question.getQuestionNumber()); 918     if (q.doesQuoteExist(forecast)) throw new QuoteAlreadyExist("Kuota existitzen da"); 919     db.getTransaction().begin(); 920 } 921	882     db.persist(ad); 883     db.getTransaction().commit(); 884 } 885 886 public boolean gertaerakSortu(String description, Date eventDate, String sport) { 887     boolean b = true; 888     db.getTransaction().begin(); 889     Sport spo = db.find(Sport.class, sport); 890     if(spo==null) { 891         b = isEventInDB(description, eventDate, b); 892     } 893     if(b) { 894         String[] taldeak = description.split("-"); 895         Team lokala = new Team(taldeak[0]); 896         Team kanpokoa = new Team(taldeak[1]); 897         Event e = new Event(description, eventDate, lokala, kanpokoa); 898         e.setSport(spo); 899         spo.addEvent(e); 900         db.persist(e); 901     } 902     return false; 903 } 904     db.getTransaction().commit(); 905     return b; 906 } 907 908 private boolean isEventInDB(String description, Date eventDate, boolean b) { 909     TypedQuery<Event> Equery = db.createQuery("SELECT e FROM Event e WHERE e.getEventDate() =?1 ",Event.class); 910     Equery.setParameter(1, eventDate); 911     for(Event ev: Equery.getResultList()) { 912         if(ev.getDescription().equals(description)) { 913             b = false; 914         } 915     } 916     return b; 917 } 918 919 public Quote storeQuote(String forecast, Double Quote, Question question) throws QuoteAlreadyExist { 920     //System.out.println(">>> DataAccess: createQuestion=> event= "+event+" question= "+question+" betMinimum 921

## ALAIA:

```
public BusinessLogicServer() {  
    addWindowListener(new WindowAdapter() {  
        @Override  
        public void windowClosed(WindowEvent arg0) {  
            System.exit(1);  
        }  
    });  
    setTitle("BusinessLogicServer: running the business logic");  
    setBounds(100, 100, 486, 209);  
    getContentPane().setLayout(new BorderLayout());  
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));  
    getContentPane().add(contentPanel, BorderLayout.CENTER);  
    contentPanel.setLayout(new BorderLayout(0, 0));  
}
```


 Extract this nested code block into a method. [Why is this an issue?](#)

last month ▾ L64 




 Code Smell  Minor  Open Not assigned 10min effort

No

```
textArea = new JTextArea();  
contentPanel.add(textArea);  
}
```


 Extract this nested code block into a method. [Why is this an issue?](#)

last month ▾ L68 




 Code Smell  Minor  Open Not assigned 10min effort

No

```
JPanel buttonPane = new JPanel();  
buttonPane.setLayout(new FlowLayout(FlowLayout.RIGHT));  
getContentPane().add(buttonPane, BorderLayout.SOUTH);  
}
```

 Extract this nested code block into a method. [Why is this an issue?](#)

last month ▾ L72 

 Code Smell  Minor  Open Not assigned 10min effort

No



111 Nesting + 1  
112 try {  
113 try {  
114 if (!c.isDatabaseLocal()) {  
115 System.out.println("WARNING: Please be sure ObjectdbManagerServer is launched");  
116 }  
117 service = "http://" + c.getBusinessLogicNode() + ":" + c.getBusinessLogicPort() + "/ws/";  
118 Endpoint.publish(service, new BLFacadeImplementation());  
119 }  
120 catch (Exception e) {  
121 System.out.println("Error in BusinessLogicServer: " + e.toString());  
122 textArea.append("\nYou should have not launched DBManagerServer...\n");  
123 textArea.append("\n\nOr maybe there is a BusinessLogicServer already launched...\n");  
124 throw e;  
125 }  
126 textArea.append("Running service at:\n" + service);  
127 textArea.append("\n\nPress button to exit this server... ");  
128 }  
129 catch (Exception e) {  
130 textArea.append(e.toString());  
131 }  
132 }  
133 }  
134 }  
135 }  
136 }  
137 }

Problems @ Javadoc Declaration Console SonarLint Report X SonarLint Issue Locations Coverage Debug Git Staging

5 items

Resource	Date	Description
BusinessLogic		Rename this package name to match the regular expression <code>^[a-z_](\.[a-z_0-9])*\$</code> .
BusinessLogic		Extract this nested try block into a separate method. [+1 location]

Kode bloke hauek

metodo bereiz batera aldatzean, kode unitateen luzera limitatzen gaude eta horrela denboran zehar kodearen mantenua errezten gaude.

Hau egiteko, hiru metodo berri sortu duguz eta kode blokeak zeuden lekuetatik deitu:

```

    }
    private void publishTry() {
        try{

            if (!c.isDatabaseLocal()) {
                System.out.println("\nWARNING: Please be sure ObjectdbManagerServer is launched\n");
            }

            service= "http://" + c.getBusinessLogicNode() + ":" + c.getBusinessLogicPort() + "/ws/" + c.getBusinessLogicNode();
            Endpoint.publish(service, new BLFacadeImplementation());

        }
        catch (Exception e) {
            System.out.println("Error in BusinessLogicServer: " + e.toString());
            textArea.append("\nYou should have not launched DBManagerServer...\n");
            textArea.append("\n\nOr maybe there is a BusinessLogicServer already launched...\n");
            throw e;
        }
    }

    private void configureContentPane() {
        setTitle("BusinessLogicServer: running the business logic");
        setBounds(100, 100, 486, 209);
        getContentPane().setLayout(new BorderLayout());
        contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
        getContentPane().add(contentPanel, BorderLayout.CENTER);
        contentPanel.setLayout(new BorderLayout(0, 0));

        configureTextArea();
    }

    private void configureTextArea() {

        textArea = new JTextArea();
        contentPanel.add(textArea);
    }

    private JPanel configureJPanel() {
        JPanel buttonPane = new JPanel();
        buttonPane.setLayout(new FlowLayout(FlowLayout.RIGHT));
        getContentPane().add(buttonPane, BorderLayout.SOUTH);
        return buttonPane;
    }

    if (c.isBusinessLogicLocal()) {
        textArea.append("\nERROR, the business logic is configured as local");
    }
    else {
        try {
            publishTry();

            textArea.append("Running service at:\n\t" + service);
            textArea.append("\n\nPress button to exit this server... ");

        } catch (Exception e) {
            textArea.append(e.toString());
        }
    }
}

```

**Andoni:**

Hasierako kodea:

```
public boolean gertaerakSortu(String description, Date eventDate, String sport) {
    boolean b = true;
    db.getTransaction().begin();
    Sport spo = db.find(Sport.class, sport);
    if(spo != null) {
        b = isEventInDB(description, eventDate, b);
        if(b) {
            String[] taldeak = description.split("-");
            Team lokala = new Team(taldeak[0]);
            Team kanpokoak = new Team(taldeak[1]);
            Event e = new Event(description, eventDate, lokala, kanpokoak);
            e.setSport(spo);
            spo.addEvent(e);
            db.persist(e);
        }
    } else {
        return false;
    }
    db.getTransaction().commit();
    return b;
}
```

ErrefaktORIZATUTAKO kodea:

```
public boolean gertaerakSortu(String description, Date eventDate, String sport) {
    boolean b = true;
    db.getTransaction().begin();
    Sport spo = db.find(Sport.class, sport);
    if(spo != null) {
        b = isEventInDB(description, eventDate, b);
        if(b) extractedForGertaerakSortu(description, eventDate, spo);
    } else {
        return false;
    }
    db.getTransaction().commit();
    return b;
}

private void extractedForGertaerakSortu(String description, Date eventDate, Sport spo) {
    String[] taldeak = description.split("-");
    Team lokala = new Team(taldeak[0]);
    Team kanpokoak = new Team(taldeak[1]);
    Event e = new Event(description, eventDate, lokala, kanpokoak);
    e.setSport(spo);
    spo.addEvent(e);
    db.persist(e);
}
```

SonarCloud-ek bad smell bezala ez markatu arren, metodoak 15 ilara baino gehiago ditu. Benetan gertaera sortu eta hau datu basean sartzen duen kode blokea metodo berri batean sartu dut, horrela metodoa motzagoa geratzen da.

## "Write simple units of code" (3. kapituloa)

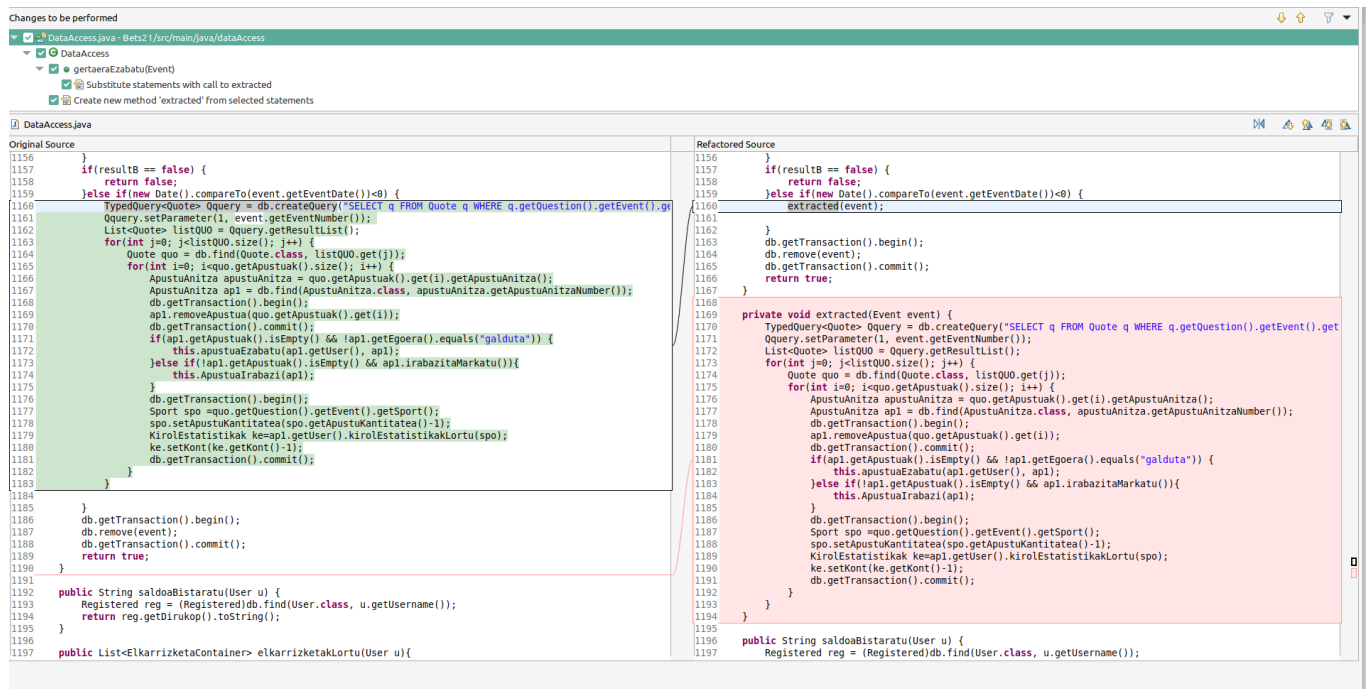
**DAVID:**

Refactor this method to reduce its cognitive complexity bad smell-a bilatu dut. Eta DataAccess-en barruan gertaerak ezabatu metodoan bat zegoen. Hasieran kodea honakoa da:

```
public boolean gertaeraEzabatu(Event ev) {
    Event event = db.find(Event.class, ev);
    boolean resultB = true;
    List<Question> listQ = event.getQuestions();

    +1
    1 for(Question q : listQ) {
        +2 (incl 1 for nesting)
        2 if(q.getResult() == null) {
            resultB = false;
        }
    }
    +1
    3 if(resultB == false) {
        return false;
    }
    +1
    4 if(new Date().compareTo(event.getEventDate())<0) {
        TypedQuery<Quote> Qquery = db.createQuery("SELECT q FROM Quote q WHERE q.getQuestion().getEvent().getEventNumber() =?1", Quote.class)
        Qquery.setParameter(1, event.getEventNumber());
        List<Quote> listQUO = Qquery.getResultList();
        +2 (incl 1 for nesting)
        5 for(int j=0; j<listQUO.size(); j++) {
            Quote quo = db.find(Quote.class, listQUO.get(j));
            +3 (incl 2 for nesting)
            6 for(int i=0; i<quo.getApustuak().size(); i++) {
                ApustuAnitza apustuAnitza = quo.getApustuak().get(i).getApustuAnitza();
                ApustuAnitza apl = db.find(ApustuAnitza.class, apustuAnitza.getApustuAnitzaNumber());
                db.getTransaction().begin();
                apl.removeApustua(quo.getApustuak().get(i));
                db.getTransaction().commit();
                +4 (incl 3 for nesting) | +1
                7 if(apl.getApustuak().isEmpty() 8 && !apl.getEgoera().equals("galduta")) {
                    this.apustuaEzabatu(apl.getUser(), apl);
                }
                +1 | +1
            } else 9 if(!apl.getApustuak().isEmpty() 10 && apl.irabazitaMarkatu()){
                this.ApustuaIrabazi(apl);
            }
            db.getTransaction().begin();
            Sport spo =quo.getQuestion().getEvent().getSport();
            spo.setApustuKantitatea(spo.getApustuKantitatea()-1);
            KirolEstatistikak ke=apl.getUser().kirolEstatistikakLortu(spo);
            ke.setKont(ke.getKont()-1);
            db.getTransaction().commit();
        }
    }
    db.getTransaction().begin();
    db.remove(event);
    db.getTransaction().commit();
    return true;
}
```

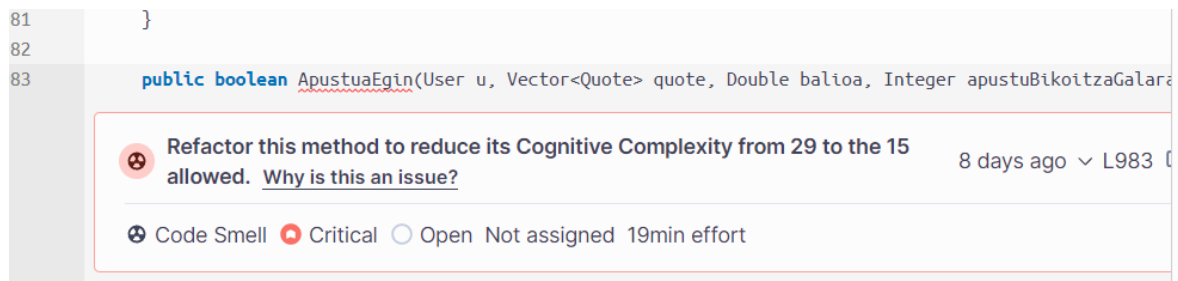
10 branch daudela ikusi dezakegu, baina refactor > extract method eginda if else barruko kodeari honela geratzen zaigu eta bad-smella desagertzen da:



Bi metodo separatuak 4 branch dituzte.

## ALAI:

29ko konplexutasunetik 15 era aldatu behar da. Hasierako kodea:



```

public boolean ApustuaEgin(User u, Vector<Quote> quote, Double balioa, Integer apustuBikoitzaGalarazi) {
    Registered user = (Registered) db.find(User.class, u.getUsername());
    Boolean b;
    if(user.getDirukop()>=balioa) {
        db.getTransaction().begin();
        ApustuAnitza apustuAnitza = new ApustuAnitza(user, balioa);
        db.persist(apustuAnitza);
        for(Quote quo: quote) {
            Quote kuote = db.find(Quote.class, quo.getQuoteNumber());
            Apustua ap = new Apustua(apustuAnitza, kuote);
            db.persist(ap);
            apustuAnitza.addApustua(ap);
            kuote.addApustua(ap);
        }
        db.getTransaction().commit();
        db.getTransaction().begin();
        if(apustuBikoitzaGalarazi==1) {
            apustuBikoitzaGalarazi=apustuAnitza.getApustuAnitzaNumber();
        }
        apustuAnitza.setApustuKopia(apustuBikoitzaGalarazi);
        user.updateDirukontua(-balioa);
        Transaction t = new Transaction(user, balioa, new Date(), "ApustuaEgin");
        user.addApustuAnitza(apustuAnitza);
        for(Apustua a: apustuAnitza.getApustuak()) {
            Apustua apu = db.find(Apustua.class, a.getApustuNumber());
            Quote q = db.find(Quote.class, apu.getKuota().getQuoteNumber());
            Sport spo = q.getQuestion().getEvent().getSport();
            spo.setApustuKantitatea(spo.getApustuKantitatea()+1);
            if(!user.containsKirola(spo)) {
                KirolEstatistikak ke=new KirolEstatistikak(user, spo);
                ke.setKont(1);
                user.addKirolEstatistikak(ke);
                spo.addKirolEstatistikak(ke);
            } else {
                KirolEstatistikak ke=user.kirolEstatistikakLortu(spo);
                ke.eguneratuKont(1);
            }
        }
        user.addTransaction(t);
        db.persist(t);
        db.getTransaction().commit();
        for(Jarraitzailea reg:user.getJarraitzaileLista()) {
            Jarraitzailea erab=db.find(Jarraitzailea.class, reg.getJarraitzaileaNumber());
            b=true;
            for(ApustuAnitza apu: erab.getNork().getApustuAnitzak()) {
                if(apu.getApustuKopia().equals(apustuAnitza.getApustuKopia())) {
                    b=false;
                }
            }
            if(b) {
                if(erab.getNork().getDirulimitea()<balioa) {
                    this.ApustuaEgin(erab.getNork(), quote, erab.getNork().getDirulimitea(), apustuBikoitzaGalarazi);
                } else {
                    this.ApustuaEgin(erab.getNork(), quote, balioa, apustuBikoitzaGalarazi);
                }
            }
        }
        return true;
    } else {
        return false;
    }
}

```

Beraz, hau zuzentzeko 3 for-ak kanpoko metodo modura errefaktorizatu ditugu. Horrela geratuko da:

```

public boolean ApustuaEgin(User u, Vector<Quote> quote, Double balioa, Integer apustuBikoitzaGalarazi) {
    Registered user = (Registered) db.find(User.class, u.getUsername());
    Boolean b;
    if(user.getDirukop()>=balioa) {
        db.getTransaction().begin();
        ApustuAnitza apustuAnitza = new ApustuAnitza(user, balioa);
        db.persist(apustuAnitza);
        extractedForQuoteApustuaEgin(quote, apustuAnitza);
        db.getTransaction().commit();
        db.getTransaction().begin();
        if(apustuBikoitzaGalarazi==1) {
            apustuBikoitzaGalarazi=apustuAnitza.getApustuAnitzaNumber();
        }
        apustuAnitza.setApustuKopia(apustuBikoitzaGalarazi);
        user.updateDirukontua(-balioa);
        Transaction t = new Transaction(user, balioa, new Date(), "ApustuaEgin");
        user.addApustuAnitza(apustuAnitza);
        extractedForApustuaApustuaEgin(user, apustuAnitza);
        user.addTransaction(t);
        db.persist(t);
        db.getTransaction().commit();
        extractedJarraitzaileaForApustuaEgin(quote, balioa, apustuBikoitzaGalarazi, user, apustuAnitza);
        return true;
    }else{
        return false;
    }
}
}

```

## Andoni:

```

public void EmaitzakIpini(Quote quote) throws EventNotFinished{

    Quote q = db.find(Quote.class, quote);
    String result = q.getForecast();

    if(new Date().compareTo(q.getQuestion().getEvent().getEventDate())<0)
        throw new EventNotFinished();

    Vector<Apustua> listApustuak = q.getApustuak();
    db.getTransaction().begin();
    Question que = q.getQuestion();
    Question question = db.find(Question.class, que);
    question.setResult(result);
    for(Quote quo: question.getQuotes()) {
        for(Apustua apu: quo.getApustuak()) {

            Boolean b=apu.galdutaMarkatu(quo);
            if(b) {
                apu.getApustuAnitza().setEgoera("galduta");
            }else {
                apu.setEgoera("irabazita");
            }
        }
    }
    db.getTransaction().commit();
    for(Apustua a : listApustuak) {
        db.getTransaction().begin();
        Boolean bool=a.getApustuAnitza().irabazitaMarkatu();
        db.getTransaction().commit();
        if(bool) {
            this.ApustuaIrabazi(a.getApustuAnitza());
        }
    }
}
}

```



```

public void EmaitzakIpini(Quote quote) throws EventNotFinished{

    Quote q = db.find(Quote.class, quote);
    String result = q.getForecast();

    if(new Date().compareTo(q.getQuestion().getEvent().getEventDate())<0)
        throw new EventNotFinished();

    Vector<Apustua> listApustuak = q.getApustuak();
    db.getTransaction().begin();
    Question que = q.getQuestion();
    Question question = db.find(Question.class, que);
    question.setResult(result);
    extractedForEmaitzakIpini(question);
    db.getTransaction().commit();
    for(Apustua a : listApustuak) {
        db.getTransaction().begin();
        Boolean bool=a.getApustuAnitza().irabazitaMarkatu();
        db.getTransaction().commit();
        if(bool) {
            this.ApustuaIrabazi(a.getApustuAnitza());
        }
    }
}

private void extractedForEmaitzakIpini(Question question) {
    for(Quote quo: question.getQuotes()) {
        for(Apustua apu: quo.getApustuak()) {

            Boolean b=apu.galdutaMarkatu(quo);
            if(b) {
                apu.getApustuAnitza().setEgoera("galduta");
            }else {
                apu.setEgoera("irabazita");
            }
        }
    }
}
}

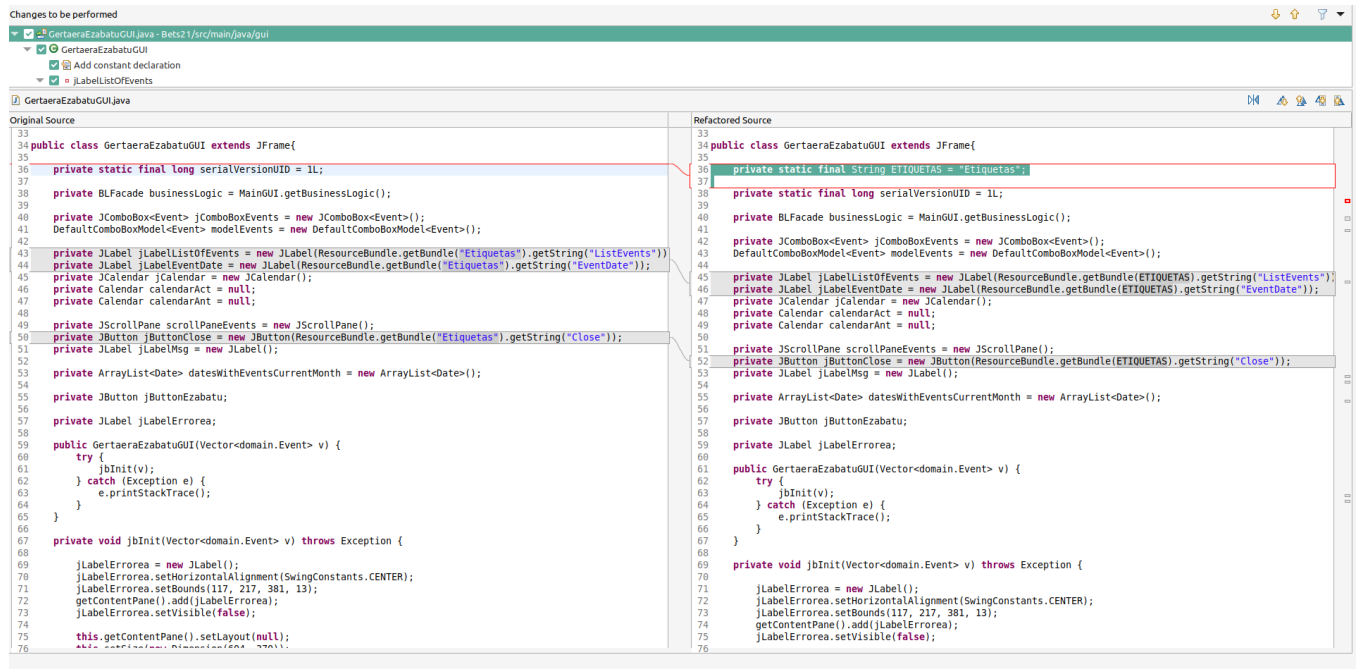
```

Berriz ere sonarcloud-ek ez du smell hau inon ez markatzen baina hau da arazoa izatetik gertuen dagoena. Azken for loop-a betse metodo batera errefaktorizatuz konplexutasun ziklomartikoa murrizten da.

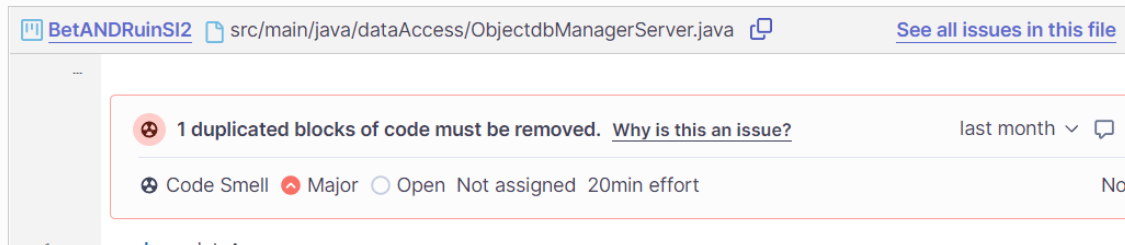
## “Duplicate code” (4. kapituloa)

### DAVID

Gertaerak ezabatu GUIan “9 duplicated lines of code must be removed” bad smell-a aurkitzen dugu. Gertatzen dena “Etiquetas” String-a askotan errepikatzen dela. Orduan refactor> extract constant erabiliz hau conpontzen da. Kodearen errefaktORIZAZIO zati bat hemen:



### ALAI



Kode smell hau bi alditan agertzen da: ObjectManageServer.java eta BusinessLogicServer.java-n. Hau horrela da, duplikatuta dagoen kodea klase bi hauetan dagoelako.

Zuzentzeko, klase berri bat sortu dugu ServerKonfig.java deitua non duplikatutako metodo eta kode hori idatzi dugun. Gero beste klaseetatik ezabatu eta ServerKonfig klaseko objektu bat sortu dugu metodoeri deitu ahal izateko.

Kasu honetan, duplikatutako kodea 2.kapitulurako zuzenketak egitean sortutako metodo berriak dira. Beraz, horiek kendu eta klase berrian kopiatuta:

```

1 package configuration;
2
3 import java.awt.BorderLayout;
4
5
6 public class ServerConfig extends JDialog{
7
8     public ServerConfig() {}
9
10    //whichServer==true BusinessLogic, false ObjectdbManager
11    public void configureContentPane(JPanel contentPanel, boolean whichServer) {
12        if(whichServer) {
13            setTitle("BusinessLogicServer: running the business logic");
14        } else {
15            setTitle("objectDBManagerServer: running the database server");
16        }
17
18        setBounds(100, 100, 486, 209);
19        getContentPane().setLayout(new BorderLayout());
20        contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
21        getContentPane().add(contentPanel, BorderLayout.CENTER);
22        contentPanel.setLayout(new BorderLayout(0, 0));
23
24    }
25
26    public void configureTextArea(JPanel contentPanel, JTextArea textArea) {
27
28        textArea = new JTextArea();
29        contentPanel.add(textArea);
30    }
31
32    public JPanel configureJPanel() {
33        JPanel buttonPane = new JPanel();
34        buttonPane.setLayout(new FlowLayout(FlowLayout.RIGHT));
35        getContentPane().add(buttonPane, BorderLayout.SOUTH);
36        return buttonPane;
37    }
38
39 }
40
41
42
43
44
45
46
47
48
49
50

```

```

ServerConfig sc = new ServerConfig();

sc.configureContentPane(contentPanel,true);
sc.configureTextArea(contentPanel, textArea);

```

Andoni: Ez dago beste kode biokoizturik.

## "Keep unit interfaces small" (5. kapituloa)

Errore tipo honetarako bakarra aurkitu dugu: mezuakBidali metodoa.

```

public boolean mezuakBidali(User igorlea, String hartzailea, String titulo, String test, Elkarriketa elkarriketa) {
    User igorle = db.find(User.class, igorlea.getUsername());
    User hartzaile = db.find(User.class, hartzailea);
    Elkarriketa elk=null;
    if(hartzaile==null) {
        return false;
    } else {
        db.getTransaction().begin();
        Message m = new Message(igorle, hartzaile, test);
        db.persist(m);
        if(elkarriketa!=null) {
            elk = db.find(Elkarriketa.class, elkarriketa.getElkarriketaNumber());
        } else {
            elk= new Elkarriketa(titulo, igorle, hartzaile);
            db.persist(elk);
            m.setElkarriketa(elk);
            igorle.addElkarriketak(elk);
            hartzaile.addElkarriketak(elk);
        }
        elk.addMezua(m);
        igorle.addBidalitakoMezuak(m);
        hartzaile.addJasotakoMezuak(m);
        db.getTransaction().commit();
        return true;
    }
}

```

Ikusi dezakegunez 5 parametro ditu, hau da, larregi. Zuzentzeko, mezuakBidaliGUI begiratzuz metodoaren deia horrela egiten da:

```
name().equals(txtHartzailea.getText())) {  
    businessLogic.mezuaBidali(user, txtHartzailea.getText(), txtAsunto.getText(), txtTestua.getText(), elkarrizketa);
```

Ikusten dugunez, hartzailetik, asuntutik eta testutik testua lortzen da. Apur bat beherago, jarraiko kodea ikusi dezakegu:

```
txtAsunto.setText(elkarrizketa.getTitulo());
```

Beraz, mezuakBidali-ri bidalitako titulo hori elkarrizketaren tituloa da. Hau jakinda, ez dago beharrik elkarrizketa eta elkarrizketaren titulua biak bidaltzeko.

Horrela geratu da:

```
public boolean mezuaBidali(User igorlea, String hartzailea, String test, Elkarrizketa elkarrizketa) {  
    User igorle = db.find(User.class, igorlea.getUsername());  
    User hartzaile = db.find(User.class, hartzailea);  
    Elkarrizketa elk=null;  
    if(hartzaile==null) {  
        return false;  
    }else {  
        db.getTransaction().begin();  
        Message m = new Message(igorle, hartzaile, test);  
        db.persist(m);  
        if(elkarrizketa!=null) {  
            elk = db.find(Elkarrizketa.class, elkarrizketa.getElkarrizketaNumber());  
        }else {  
            elk= new Elkarrizketa(elkarrizketa.getTitulo(), igorle, hartzaile);  
            db.persist(elk);  
            m.setElkarrizketa(elk);  
            igorle.addElkarrizketak(elk);  
            hartzaile.addElkarrizketak(elk);  
        }  
        elk.addMezua(m);  
        igorle.addBidalitakoMezuak(m);  
        hartzaile.addJasotakoMezuak(m);  
        db.getTransaction().commit();  
        return true;  
    }  
}
```