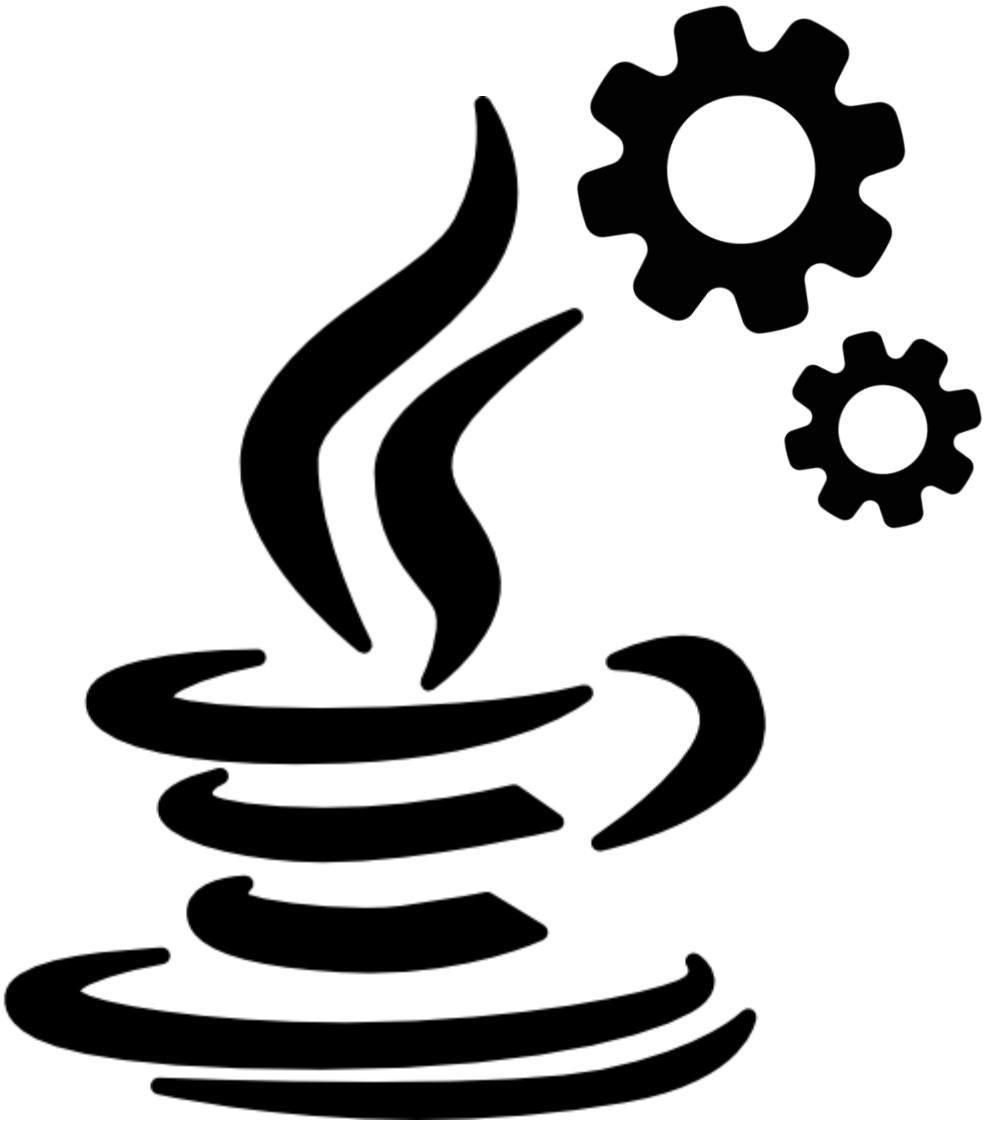
ERREFAKTORIZAZIOA JAVAN



Xabier Gonzalez, Mikel Arana, Unax Iraola

Aurkibidea

1.Sarrera

Javan kodeatzea zaila izan daiteke, baina batzuetan kodetzea baino zailagoa kodigoa ulertzea eta puskatuta dagoen kodigoa konpontzea ere. Beraz, gure kodigoa konpontzeko hau errefaktorizatu beharko dugu.

4 “Bad smell” mota ikasi ditugu gure kodigoan “arazoak” ekarri dezaketenak, hauek hurrengoak dira:

Metodoak 15 linea baino gehiagokoak izatea

Metodoen konplejotaxun ziklomatikoa 4 baino gehiagokoa ez izatea

Metodoak erabiltzen dituzten aldagaiak 4 baino gehiago ez izatea

Kodigoan dauden gauza errepikatuak sahiestea

Beraz, taldekide bakoitzak 4 “bad Smell” konpondu ditugu, 1 mota bakoitzeko. Hauek izan dira aldaketak.

1.Xabi

TakeMoney metodoan parametroen kantitatea murriztu dugu. Enuntziatuak 4 parametro edo gehioko metodoak aldatzea eskatzen zigun, baina ez geneuzkan askorik. Beraz hirukoak ere murriztea erabaki dugu. Kasu honetan float money, User us eta String desc aldatzen ditugu TakeMoneyClass t baten ordez. Klase horretan tratatzen ditugu lehen behar genituen parametroak.

Hau egiteko BLFacadeInplementation-en dagoen takeMoney metodoa ere aldatu behar izan dugu.

TakeMoney metodoa lehen:

```
public boolean takeMoney(float money, User us, String desc) {
      User us1= db.find(User.class, us);

      if (us1== null) {
         return false;
      }else {

         float wallet = us1.getWallet();
         if (wallet < money) {
            System.out.println("Eztago wallet-ean") ;
            return false;
         }else {
            db.getTransaction().begin();
            us1.setWallet(wallet-money);
            Movement move= new Movement(desc, money);
            db.getTransaction().commit();
            this.addMovement(us1, move);
         }
         return true;
      }
   }
```

```
public boolean takeMoney(float money, User us, String desc) {
      dbManager.open(false);
      boolean b=dbManager.takeMoney(money, us, desc);
      dbManager.close();
      return b;
   }
```

Orain:

```
public boolean takeMoney(TakeMoneyClass t) {
    User us1= db.find(User.class, t.getUs());

    if (us1== null) {
       return false;
    }else {

       float wallet = us1.getWallet();
       if (wallet < t.getMoney()) {
          System.out.println("Eztago wallet-ean") ;
          return false;
       }else {
          db.getTransaction().begin();
          us1.setWallet(wallet-t.getMoney());
          Movement move= new Movement(t.getDesc(), t.getMoney());
          db.getTransaction().commit();
          this.addMovement(us1, move);
       }
       return true;
    }
}
```

RemoveFollow metodoan branch kopuruak murriztu egin ditugu. Kasu honetan azpi metodo batzuk eginda. Horrela metodo nagusiari konplexutasuna kentzen diogukentzen diogu. Kasu honetan removeFollow2 eta removeFollow3

RemoveFollow metodoa lehen:

```
public void removeFollow(User jarraitua, User jarraitzaile) {
    User us1 = db.find(User.class, jarraitua);
    User us2 = db.find(User.class, jarraitzaile);
    db.getTransaction().begin();
    Iterator<User> j1 = us1.getJarraitzaileak().iterator();
    while (j1.hasNext()) {
      User u = j1.next();
      if (u.equals(us2)) {
        j1.remove();
      }
    }
    Iterator<User> j2= us2.getJarraituak().iterator();
    while (j2.hasNext()) {
      User u2 = j2.next();
      if (u2.equals(us1)) {
        j2.remove();
      }
    }
    db.getTransaction().commit();
  }
```

Orain:

```
public void removeFollow(User jarraitua, User jarraitzaile) {
    User us1 = db.find(User.class, jarraitua);
    User us2 = db.find(User.class, jarraitzaile);
    db.getTransaction().begin();
    this.removeFollow2(us1, us2);
    this.removeFollow3(us1, us2);
    db.getTransaction().commit();
  }

  protected void removeFollow2(User us1, User us2){
    Iterator<User> j1 = us1.getJarraitzaileak().iterator();
    while (j1.hasNext()) {
      User u = j1.next();
      if (u.equals(us2)) {
        j1.remove();
      }
    }
  }
  protected void removeFollow3(User us1, User us2) {
    Iterator<User> j2= us2.getJarraituak().iterator();
    while (j2.hasNext()) {
      User u2 = j2.next();
      if (u2.equals(us1)) {
        j2.remove();
      }
    }
  }
```

Kodigoan kopiak murrizteko, DataAccess klasean errepikatutako hitz batzuk aldagaiak bihurtu ditugu

```
String Lokala = "Lokala";

qq11=q1.addQuote(Lokala, 1.5);
qq12=q1.addQuote(Bisitaria, 2);
qq21=q2.addQuote("Lokalak", 2.5);
qq22=q2.addQuote("Bisitariak", 1);
qq31=q3.addQuote(Lokala, 3);
qq32=q3.addQuote(Bisitaria, 1);
qq41=q4.addQuote("1", 1.5);
qq42=q4.addQuote("2", 2);
qq51=q5.addQuote(Lokala, 1.5);
qq52=q5.addQuote(Bisitaria, 2);
qq61=q6.addQuote("Bai", 1.5);
qq62=q6.addQuote("Ez", 2);
```

Lokala idatzita zegoen leku guztietan Lokala aldagai batekin ordeztu ditugu etenero “Lokala” idatzita eduki beharrean, horrela edozein gauza aldatuko beharko bagenu, aldagaiaren edukia aldatzen nahikoa izango zen

makeWinner metodoan 15 lerro baino gehiago geneukan, beraz bad smell hau zuzentzeko metodo laguntzaile baten bitartez (galtzaileak metodoa) for begizta bat exekutatzen dugu.

MakeWinner metodoa lehen:

```
public boolean makeWinner(Question quest, Quote q) {
Refactor this method to reduce its Cognitive Complexity from 17 to the 15 allowed.
      System.out.println("hola");
      Quote q1= db.find(Quote.class, q);
      Question quest2 = db.find(Question.class, quest);

if (quest2 == null
|| q1==null) {

if(quest2 == null) {
         System.out.println("no llega quest ");
         }

if(q1 ==null) {
            System.out.println("no llega quote");
         }
         return false;
      }
else {
         System.out.println("llega aqui");

if (q1.isWinner()==1) {
            return false;
         }
else {
            db.getTransaction().begin();
            q1.makeWinner();
            //System.out.println("makeWinner true");


for(Quote qq: quest2.getQuotes()) {
               Quote qq2 = db.find(Quote.class, qq);

if(!qq2.getQuoteNumber().equals(q1.getQuoteNumber()))
                  qq2.makeLoser();
            }
            //System.out.println("is Winner: " + q1.isWinner());
            db.getTransaction().commit();
            return true;
         }
      }
   }
```

Orain:

```
protected void Galtzaileak(Question galdera, Quote ql) {
    for (Quote qq : galdera.getQuotes()) {
        Quote qq2 = db.find(Quote.class, qq);
        if (!qq2.getQuoteNumber().equals(ql.getQuoteNumber()))
            qq2.makeLoser();
    }
}

public boolean makeWinner(Question quest, Quote q) {
    System.out.println("hola");
    Quote ql = db.find(Quote.class, q);
    Question quest2 = db.find(Question.class, quest);
    if (quest2 == null || ql == null) {
        if (quest2 == null) {
            System.out.println("no llega quest ");
        }
        if (ql == null) {
            System.out.println("no llega quote");
        }
        return false;
    } else {
        System.out.println("llega aqui");
        if (ql.isWinner() == 1) {
            return false;
        } else {
            db.getTransaction().begin();
            ql.makeWinner();
            // System.out.println("makeWinner true");
            Galtzaileak(quest2, ql);
            // System.out.println("is Winner: " + ql.isWinner());
            db.getTransaction().commit();
            return true;
        }
    }
}
```

```
public RegisterGUI() {

    this.eraikitzaileak1();
    this.eraikitzaileak2();
    this.eraikitzaileak3();
    this.eraikitzaileak4();
    this.eraikitzaileak5();
    this.eraikitzaileak6();

    SignInButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if (NameField.getText().length() == 0 || AgeField.getText().length() == 0
                    || EmailField.getText().length() == 0 || UserNameField.getText().length() == 0
                    || passwordField.getPassword().length == 0 || PassConfField.getPassword().length == 0) {
                lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error1"));
            } else {
                try {
                    eraikitzaileak8();
                } catch (NumberFormatException execption) {
                    lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error5"));
                }
            }
        }
    });

    contentPane.setLayout(null);
    contentPane.add(SignInButton);

}
```

2.Mikel

Register metodoan parametroen kantitatea murriztu dugu. Enuntziatuak 4 parametro edo gehioko metodoak aldatzea eskatzen zigun. Kasu honetan String username, String password, String Izena eta Int age pasa ordez, BLFacadeInplemetation-en sortutako user bat pasatzen diogu.

Hau egiteko BLFacadeInplementation-en dagoen register metodoa ere aldatu behar izan dugu.

Register metodoa lehen:

```
public boolean register( String username, String password, String izena, int age) {
      UserAbstract us= db.find(UserAbstract.class, username);
      if (us== null) {
         db.getTransaction().begin();
         System.out.println(username);
         db.persist((UserAbstract)new User(username, password, izena, age));
         //db.persist((UserAbstract)new UserAdmin(username, password));
         db.getTransaction().commit();
         System.out.println("Erregistratu da");
         return true;
      }else {
         return false;
      }
   }
```

Orain:

```
  public boolean register(User us) {
     UserAbstract us1= db.find(UserAbstract.class, us.getUsername());
     if (us1== null) {
        db.getTransaction().begin();

        db.persist((UserAbstract)us);
        //db.persist((UserAbstract)new UserAdmin(username, password));
        db.getTransaction().commit();

        return true;
     }else {
        return false;
     }
  }
```

```
public boolean register( String username, String password, String izena, int age) {
    dbManager.open(false);
    User us = new User(username,password, izena,age);
    boolean b=dbManager.register(us);
    dbManager.close();
    return b;
 }
```

CopyEvent metodoan branch kopuruak murriztu egin ditugu. Kasu honetan azpi metodo batzuk eginda. Horrela metodo printzipalari karga kentzen diogu. Kasu honetan copyEvent2

CopyEvent metodoa lehen:

```
public boolean copyEvent(Event ev, Date date) {
   Event evdb = db.find(Event.class, ev);
   if (evdb == null) {
      return false;
   } else {
      this.createEvent(evdb.getDescription(), date);
      TypedQuery<Event> query1 = db.createQuery("SELECT ev FROM Event ev WHERE ev.eventDate=?1 AND ev.description=?2", Event.class);
      query1.setParameter(1, date);
      query1.setParameter(2, evdb.getDescription());
      for (Question q: evdb.getQuestions()) {
         try {
         Question q2 = this.createQuestion(query1.getSingleResult(), q.getQuestion(), q.getBetMinimum());
         Question qdb = db.find(Question.class, q2);
         for (Quote qq : q.getQuotes()) {
            this.createQuote(qdb, qq.getQuote(), qq.getMulti());
         }

         } catch (QuestionAlreadyExist e) {
            e.printStackTrace();
         }
      }
      return true;
   }
```

Orain:

```
public boolean copyEvent(Event ev, Date date) {
   Event evdb = db.find(Event.class, ev);
   if (evdb == null) {
      return false;
   } else {
      this.createEvent(evdb.getDescription(), date);
      TypedQuery<Event> query1 = db.createQuery("SELECT ev FROM Event ev WHERE ev.eventDate=?1 AND ev.description=?2", Event.class);
      query1.setParameter(1, date);
      query1.setParameter(2, evdb.getDescription());

      return true;
   }
}
protected void copyEvent2(Event evdb, TypedQuery<Event> query1) {
   for (Question q: evdb.getQuestions()) {
      try {
      Question q2 = this.createQuestion(query1.getSingleResult(), q.getQuestion(), q.getBetMinimum());
      Question qdb = db.find(Question.class, q2);
      for (Quote qq : q.getQuotes()) {
         this.createQuote(qdb, qq.getQuote(), qq.getMulti());
      }

      } catch (QuestionAlreadyExist e) {
         e.printStackTrace();
      }
   }
}
```

Kodigoan kopiak murrizteko, DataAccess klasean errepikatutako hitz batzuk aldagaiak bihurtu ditugu

```
String Bisitaria = "Bisitaria";

qq11=q1.addQuote(Lokala, 1.5);
qq12=q1.addQuote(Bisitaria, 2);
qq21=q2.addQuote("Lokalak", 2.5);
qq22=q2.addQuote("Bisitariak", 1);
qq31=q3.addQuote(Lokala, 3);
qq32=q3.addQuote(Bisitaria, 1);
qq41=q4.addQuote("1", 1.5);
qq42=q4.addQuote("2", 2);
qq51=q5.addQuote(Lokala, 1.5);
qq52=q5.addQuote(Bisitaria, 2);
qq61=q6.addQuote("Bai", 1.5);
qq62=q6.addQuote("Ez", 2);
```

Bisitaria idatzita zegoen leku guztietan Bisitaria aldagai batekin ordeztu ditugu etenero “Bisitaria” idatzita eduki beharrean, horrela edozein gauza aldatuko beharko bagenu, aldagaiaren edukia aldatzen nahikoa izango zen

RegisterGUI-en lerroak murrizteko eraikitzaileak metodoak sortu ditugu eta bertan objektuak deklaratzen ditugu:

RegisterGUI metodoa lehen:

```
public RegisterGUI() {
		this.setTitle(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_sign"));
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 679, 517);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);

		SignInButton = new JButton(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_sign"));
		SignInButton.setBounds(285, 419, 140, 48);
		SignInButton.setForeground(Color.WHITE);
		SignInButton.setBackground(new Color(0, 0, 0));


		JLabel lblError = new JLabel("");
		lblError.setHorizontalAlignment(SwingConstants.CENTER);
		lblError.setBounds(150, 369, 357, 13);
		contentPane.add(lblError);




		SignInButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				if(NameField.getText().length() == 0 || AgeField.getText().length() == 0 || EmailField.getText().length() == 0 || UserNameField.getText().length() == 0 || passwordFie
					lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error1"));
				} else {
					try {
						int age = Integer.parseInt(AgeField.getText());
						if(age < 18)
							lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error2"));
						else if(!arrToString(passwordField.getPassword()).equals(arrToString(PassConfField.getPassword())))
							lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error3"));
						else {
							boolean b = MainGUI.getBusinessLogic().register(UserNameField.getText(), arrToString(passwordField.getPassword()), NameField.getText(), age);
							if (!b) {
								User us=(User) MainGUI.getBusinessLogic().getUserByName(UserNameField.getText());
								//MainGUI.getBusinessLogic().setUser(us);
								lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error4"));
							} else {
								UserAbstract us = MainGUI.getBusinessLogic().getUserByName(UserNameField.getText());
								MainGUI.setUser(us);
								MenuGUI m = new MenuGUI();
								m.setVisible(true);
								RegisterGUI.this.setVisible(false);
							}
						}
					} catch (NumberFormatException execption) {
						lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error5"));
					}
				}
			}
```

```
                }
        }
});


contentPane.setLayout(null);
contentPane.add(SignInButton);

BackButton = new JButton(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_back"));
BackButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                MainGUI main=new MainGUI();
                main.setVisible(true);
                RegisterGUI.this.setVisible(false);
        }
});
BackButton.setBackground(new Color(255, 255, 255));
BackButton.setBounds(466, 419, 140, 48);
contentPane.add(BackButton);

NameField = new JTextField();
NameField.setBounds(150, 24, 478, 38);
contentPane.add(NameField);
NameField.setColumns(10);

NameLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_name"));
NameLabel.setBounds(10, 24, 130, 38);
contentPane.add(NameLabel);

AgeField = new JTextField();
AgeField.setColumns(10);
AgeField.setBounds(150, 73, 478, 38);
contentPane.add(AgeField);

AgeLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_age"));
AgeLabel.setBounds(10, 73, 130, 38);
contentPane.add(AgeLabel);

EmailField = new JTextField();
EmailField.setColumns(10);
EmailField.setBounds(150, 131, 478, 38);
contentPane.add(EmailField);

UserNameField = new JTextField();
UserNameField.setColumns(10);
UserNameField.setBounds(150, 186, 478, 38);
contentPane.add(UserNameField);

EmailLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_email"));
EmailLabel.setBounds(10, 130, 130, 38);
contentPane.add(EmailLabel);

UserNameLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_username"));
```

```
        EmailLabel.setBounds(10, 130, 130, 38);
        contentPane.add(EmailLabel);

        UserNameLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_username"));
        UserNameLabel.setBounds(10, 185, 130, 38);
        contentPane.add(UserNameLabel);

        PasswordLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_password"));
        PasswordLabel.setBounds(10, 243, 130, 38);
        contentPane.add(PasswordLabel);

        PassConfLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_confPassword"));
        PassConfLabel.setBounds(10, 302, 130, 38);
        contentPane.add(PassConfLabel);

        passwordField = new JPasswordField();
        passwordField.setToolTipText("");
        passwordField.setBounds(150, 244, 478, 38);
        contentPane.add(passwordField);

        PassConfField = new JPasswordField();
        PassConfField.setBounds(150, 303, 478, 38);
        contentPane.add(PassConfField);

        Login = new JButton(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_login"));
        Login.setBackground(new Color(255, 255, 255));
        Login.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        LoginGUI login=new LoginGUI();
                        login.setVisible(true);
                        RegisterGUI.this.setVisible(false);
                }
        });
        Login.setBounds(105, 419, 140, 48);
        contentPane.add(Login);

    }
```

Orain:

```
public RegisterGUI() {

    this.eraikitzaileak1();
    this.eraikitzaileak2();
    this.eraikitzaileak3();
    this.eraikitzaileak4();
    this.eraikitzaileak5();
    this.eraikitzaileak6();

    SignInButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if (NameField.getText().length() == 0 || AgeField.getText().length() == 0
                    || EmailField.getText().length() == 0 || UserNameField.getText().length() == 0
                    || passwordField.getPassword().length == 0 || PassConfField.getPassword().length == 0) {
                lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error1"));
            } else {
                try {
                    eraikitzaileak8();
                } catch (NumberFormatException execption) {
                    lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error5"));
                }
            }
        }
    });

    contentPane.setLayout(null);
    contentPane.add(SignInButton);

}
```

```
protected void eraikitzaileak1() {

    SignInButton = new JButton(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_sign"));
    SignInButton.setBounds(285, 419, 140, 48);
    SignInButton.setForeground(Color.WHITE);
    SignInButton.setBackground(new Color(0, 0, 0));
    BackButton = new JButton(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_back"));
    BackButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            MainGUI main = new MainGUI();
            main.setVisible(true);
            RegisterGUI.this.setVisible(false);
        }
    });
    BackButton.setBackground(new Color(255, 255, 255));
    BackButton.setBounds(466, 419, 140, 48);
    contentPane.add(BackButton);

    AgeLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_age"));
    AgeLabel.setBounds(10, 73, 130, 38);
    contentPane.add(AgeLabel);

}

protected void eraikitzaileak2() {
    NameField = new JTextField();
    NameField.setBounds(150, 24, 478, 38);
    contentPane.add(NameField);
    NameField.setColumns(10);

    NameLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_name"));
    NameLabel.setBounds(10, 24, 130, 38);
    contentPane.add(NameLabel);

    AgeField = new JTextField();
    AgeField.setColumns(10);
    AgeField.setBounds(150, 73, 478, 38);
    contentPane.add(AgeField);

}
```

```
protected void eraikitzaileak3() {

    EmailField = new JTextField();
    EmailField.setColumns(10);
    EmailField.setBounds(150, 131, 478, 38);
    contentPane.add(EmailField);

    UserNameField = new JTextField();
    UserNameField.setColumns(10);
    UserNameField.setBounds(150, 186, 478, 38);
    contentPane.add(UserNameField);

    EmailLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_email"));
    EmailLabel.setBounds(10, 130, 130, 38);
    contentPane.add(EmailLabel);

    UserNameLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_username"));
    UserNameLabel.setBounds(10, 185, 130, 38);
    contentPane.add(UserNameLabel);
}

protected void eraikitzaileak4() {
    PasswordLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_password"));
    PasswordLabel.setBounds(10, 243, 130, 38);
    contentPane.add(PasswordLabel);

    PassConfLabel = new JLabel(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_confPassword"));
    PassConfLabel.setBounds(10, 302, 130, 38);
    contentPane.add(PassConfLabel);

    passwordField = new JPasswordField();
    passwordField.setToolTipText("");
    passwordField.setBounds(150, 244, 478, 38);
    contentPane.add(passwordField);

    PassConfField = new JPasswordField();
    PassConfField.setBounds(150, 303, 478, 38);
    contentPane.add(PassConfField);
}
```

```
protected void eraikitzaileak5() {
    this.setTitle(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_sign"));
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 679, 517);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);

    lblError = new JLabel("");
    lblError.setHorizontalAlignment(SwingConstants.CENTER);
    lblError.setBounds(150, 369, 357, 13);
    contentPane.add(lblError);
}

protected void eraikitzaileak6() {

    Login = new JButton(ResourceBundle.getBundle("Etiquetas").getString("LoginGUI_login"));
    Login.setBackground(new Color(255, 255, 255));
    Login.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            LoginGUI login = new LoginGUI();
            login.setVisible(true);
            RegisterGUI.this.setVisible(false);
        }
    });
    Login.setBounds(105, 419, 140, 48);
    contentPane.add(Login);
}

protected void eraikitzaileak7() {
    UserAbstract us = MainGUI.getBusinessLogic().getUserByName(UserNameField.getText());
    MainGUI.setUser(us);
    MenuGUI m = new MenuGUI();
    m.setVisible(true);
    RegisterGUI.this.setVisible(false);
}
```

```
protected void eraikitzaileak8() {
    int age = Integer.parseInt(AgeField.getText());
    if (age < 18)
        lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error2"));
    else if (!arrToString(passwordField.getPassword()).equals(arrToString(PassConfField.getPassword())))
        lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error3"));
    else {
        boolean b = MainGUI.getBusinessLogic().register(UserNameField.getText(),
                arrToString(passwordField.getPassword()), NameField.getText(), age);
        if (!b) {
            User us = (User) MainGUI.getBusinessLogic().getUserByName(UserNameField.getText());
            // MainGUI.getBusinessLogic().setUser(us);
            lblError.setText(ResourceBundle.getBundle("Etiquetas").getString("RegisterGUI_error4"));
        } else {
            eraikitzaileak7();
        }
    }
}
```

3.Unax

PutMoney metodoan parametroen kantitatea murriztu dugu. Enuntziatuak 4 parametro edo gehioko metodoak aldatzea eskatzen zigun, baina ez geneuzkan askorik. Beraz hirukoak ere murriztea erabaki dugu. Kasu honetan float money, User us eta String desc aldatzen ditugu TakeMoneyClass t baten ordez. Klase horretan tratatzen ditugu lehen behar genituen parametroak.

Hau egiteko BLFacadeInplementation-en dagoen putMoney metodoa ere aldatu behar izan dugu.

PutMoney metodoa lehen:

```
public boolean putMoney(float money, User us, String desc) {
   User us1 = db.find(User.class, us);
   if (us1 == null) {
      return false;
   }else {
      db.getTransaction().begin();
      us1.setWallet(us1.getWallet()+ money);
      db.getTransaction().commit();
      Movement move= new Movement(desc, money);
      this.addMovement(us1, move);
      return true;
   }
}
```

```
public boolean putMoney(float money, User us, String desc) {
   dbManager.open(false);
   boolean b = dbManager.putMoney(money, us, desc);
   dbManager.close();
   return b;
}
```

Orain:

```
public boolean putMoney(TakeMoneyClass t) {
   User us1 = db.find(User.class, t.getUs());
   if (us1 == null) {
      return false;
   }else {
      db.getTransaction().begin();
      us1.setWallet(us1.getWallet()+ t.getMoney());
      db.getTransaction().commit();
      Movement move= new Movement(t.getDesc(), t.getMoney());
      this.addMovement(us1, move);
      return true;
   }
}
```

```
  public boolean putMoney(float money, User us, String desc) {
     dbManager.open(false);
     TakeMoneyClass t= new TakeMoneyClass(money,us, desc);
     boolean b = dbManager.putMoney(t);
     dbManager.close();
     return b;
  }
```

GetUnfollows metodoan branch kopuruak murriztu egin ditugu. Kasu honetan azpi metodo batzuk eginda. Horrela metodo printzipalari karga kentzen diogu. Kasu honetan getUnfollows2 metodoarekin.

GetUnfollows metodoa lehen:

```
public ArrayList<User> getUnfollows(User us){
   User us1 = db.find(User.class, us);
   ArrayList<User> nofollowing = new ArrayList<User>();
   TypedQuery<User> guztiak = db.createQuery("SELECT DISTINCT us FROM User us", User.class);
   if (!guztiak.getResultList().isEmpty()) {
      for (User ez : guztiak.getResultList()) {
         if (!us1.getJarraituak().contains(ez) && !us1.equals(ez)) {
            nofollowing.add(ez);
         }
      }
   }
      return nofollowing;
}
```

Orain:

```
public ArrayList<User> getUnfollows(User us){
      User us1 = db.find(User.class, us);
      ArrayList<User> nofollowing = new ArrayList<User>();
      TypedQuery<User> guztiak = db.createQuery("SELECT DISTINCT us FROM User us", User.class);
      if (!guztiak.getResultList().isEmpty()) {

      }
         return nofollowing;
   }
   protected void getUnfollows2(ArrayList<User> nofollowing, TypedQuery<User> guztiak, User us1) {
      for (User ez : guztiak.getResultList()) {
         if (!us1.getJarraituak().contains(ez) && !us1.equals(ez)) {
            nofollowing.add(ez);
         }
      }
   }
```

Kodigoan kopiak murrizteko, DataAccess klasean errepikatutako hitz batzuk aldagaiak bihurtu ditugu

```
String Local ="Local";
    qq11= q1.addQuote(Local, 1.5);
    qq12= q1.addQuote("Visitante", 2);
    qq21=q2.addQuote(Local, 2.5);
    qq22= q2.addQuote("Visitante", 1);
    qq31= q3.addQuote(Local, 3);
    qq32=q3.addQuote("Visitante", 1);
    qq41= q4.addQuote("1", 1.5);
    qq42=q4.addQuote("2", 2);
    qq51=q5.addQuote(Local, 1.5);
    qq52=q5.addQuote("Visitante", 2);
    qq61=q6.addQuote("Si", 1.5);
    qq62=q6.addQuote("No", 2);

}
else if (Locale.getDefault().equals(new
    q1=ev1.addQuestion("Who will win th
    q2=ev1.addQuestion("Who will score
    q3=ev11.addQuestion("Who will win t
    q4=ev11.addQuestion("How many goals
    q5=ev17.addQuestion("Who will win t
    q6=ev17.addQuestion("Will there be

    qq11= q1.addQuote(Local, 1.5);
    qq12= q1.addQuote("Visitor", 2);
    qq21=q2.addQuote(Local, 2.5);
    qq22= q2.addQuote("Visitor", 1);
    qq31= q3.addQuote(Local, 3);
    qq32=q3.addQuote("Visitor", 1);
    qq41= q4.addQuote("1", 1.5);
    qq42=q4.addQuote("2", 2);
    qq51=q5.addQuote(Local, 1.5);
    qq52=q5.addQuote("Visitor", 2);
    qq61=q6.addQuote("Yes", 1.5);
    qq62=q6.addQuote("No", 2);
}
```

Local idatzita zegoen leku guztietan Local aldagai batekin ordeztu ditugu etenero “Local” idatzita eduki beharrean, horrela edozein gauza aldatuko beharko bagenu, aldagaiaren edukia aldatzen nahikoa izango zen

addResults metodoen lerroak murrizteko addMoneyFor eta addStatistics metodo laguntzaileak sortu ditugu:

addResults metodoa lehen:

```
@WebMethod
public void addResult(Question q, Quote qq) {
    boolean irabazle = true;
    this.makeWinner(q, qq);
    dbManager.open(false);
    Quote dbq = dbManager.getQuote(qq);
    Question quest = dbManager.getQuestion(q.getQuestionNumber());
    dbManager.close();
    for (Bet b : dbq.getBets()) {
        User u = (User) this.getUserByName(b.getUser().getIzena());
        for (Quote ql : b.getQuotes()) {
            if (ql.isWinner() != 1) {
                irabazle = false;
            }
        }

        float moneywin = b.getKop();
        if (irabazle) {
            for (Quote ql : b.getQuotes()) {
                moneywin *= ql.getMulti();
            }
            this.putMoney(moneywin, u, "ApustuaIrabazi");
        }

        dbManager.open(false);
        dbManager.updateStatistics(u, moneywin, irabazle);
        dbManager.close();
    }
    for (Quote qql : quest.getQuotes()) {
        dbManager.open(false);
        Quote qq2 = dbManager.getQuote(qql);
        dbManager.close();
        if (!qq2.getQuoteNumber().equals(qql.getQuoteNumber())) {
            for (Bet bl : qq2.getBets()) {
                User us = (User) this.getUserByName(bl.getUser().getIzena());
                dbManager.open(false);
                dbManager.updateStatistics(us, 0, false);
                dbManager.close();
            }
        }
    }
}
```

Orain:

```java
@WebMethod
public void addResult(Question q, Quote qq) {
    boolean irabazle = true;
    this.makeWinner(q, qq);
    dbManager.open(false);
    Quote dbq = dbManager.getQuote(qq);
    Question quest = dbManager.getQuestion(q.getQuestionNumber());
    dbManager.close();
    for (Bet b : dbq.getBets()) {
        irabazle = this.addMoneyFor(b, irabazle);
    }
    for (Quote qq1 : quest.getQuotes()) {
        this.AddStatistics(qq1, qq);
    }
}

protected void AddStatistics(Quote qq1, Quote qwin) {
    dbManager.open(false);
    Quote qq2 = dbManager.getQuote(qq1);
    dbManager.close();
    if(!qq2.getQuoteNumber().equals(qwin.getQuoteNumber())) {
        for (Bet b1: qq2.getBets()) {
            User us = (User) this.getUserByName(b1.getUser().getIzena());
            dbManager.open(false);
            dbManager.updateStatistics(us, 0, false);
            dbManager.close();
        }
    }
}
```

```java
protected boolean addMoneyFor(Bet b, boolean irabazle) {
    User u = (User) this.getUserByName(b.getUser().getIzena());
    for (Quote ql : b.getQuotes()) {
        if (ql.isWinner() != 1) {
            irabazle = false;
        }
    }
```