

## SAN FERMIN



EGILEAK:

Leire Oiden  
Xenki Etxaburu  
Ander Beitia  
Unai Guenaga

## **AURKIBIDEA**

Praktikaren aurkezpena.....	3
EKE.....	4
Domeinuaren eredia.....	5
Interfaze grafikoaren zirriborroak .....	6
Sekuentzia diagramak .....	14
GitHub-eko URL .....	17
Datu basearen eskema .....	18
Kodea .....	18
Junit-.....	19
Orduak.....	25
Ondorioak.....	25
Dokumentazioa.....	25

## **PRAKTIKAREN AURKEZPENA**

Praktika honen helburua San Ferminak kudeatzeko aplikazio bat sortzea da. Horretarako ikasgaian zehar ikasten joan garen harramintez baliatu gara. Kurtsoa hasiaz batera praktikan erabili beharreko tresnak ikasten hasi ginen; erabilpen kasuak, domeinuaren eredua, interfaze grafikoen zirriborroak, swing erabili, datu base bat java lengoaiaz kudeatu... Gai hauetako batzuk san ferminetako praktikarekin lotura zuzena izan dute (adibidez domeinuaren eredua praktikatzeko praktika honen enuntziatua erabili genuen).

Praktika 3 zatitan bana daitekela esan genezake. Lehenengo zatia Kautotu litzateke, hemen erabiltzaileak programara sartzerakoan bere erabiltzaile izena (Administratzailea izan edo ganadutegi bateko arduraduna) eta pasahitza sartuz aplikazioa erabiltzeko eskubidea lortzen du. Pasahitza edo erabiltzailea okerra balitz, pertsona honek ez luke gure aplikazioa erabiltzeko eskubiderik izango. Gainera lehenengo zati honetan erabiltzaileek bere pasahitza aldatzeko aukera ere izango dute non berain erabiltzaile izena eta pasahitza zaharra sartuz, pasahitza berria idatziz ahaliko duten. Aurretik aipatu bezala gure aplikazioan bi erabiltzaile mota egongo dira: alde batetik administratzailea eta besetik ganadutegietako arduradunak. Hauek dira zehazki praktika honetako beste bi zatiak. Administratzaileari dagokionez bere eskubideen artean Ganadutegiak kudeatzea (hau da, aplikaziora ganadutegi berriak sartu eta ezabatu, hauei datuak aldatu, erabiltzaile berriak sortu...).

Azken zatia erabiltzaileena da (ganadutegietako gerenteak alegia). Hauek beraien erabiltzaile eta pasahitza egokiak sartu ondoren erabiltzaileen lehioa pantailaratuko da, hainbat aukera edukiko dituelarik. Hasteko bere ganadutegian abereak gehitu eta kentzeko aukera. Gainera, ganadutegi horrek esleituta dituen entzierrora eraman beharreko zezenak ere kudeatu ahal izango ditu (zezenak kenduz eta gehituz). Izan daiteke abereak tratamendu batean egotea, hortaz tratamenduak kudeatzeko aukera ere izango du. Bertan animaliei botikak esleitu ahaliko zaie.

Enuntziatua ulertu eta tresna guztiak eskura izanda, lanari ekin genion. MySQL Workbench-en laguntzaz datu basea eraiki genuen. Bertan beharrezkoak ziren taulak sortuz, hauen arteko lotura guztiak bermatzeko beharrezkoak genituen taulak sortuz. Horrela ba, datu basea behin bukatuta java lengoaiarekin lotu behar genuen. Aurretik eginiko domeinuaren ereduez baliatuta java klaseak sortu genituen, hauek pakete ezberdinetan sailkatuta (administratzailea, erabiltzailea eta orokorra).

Javan programatzeko taldea bitan banatu genuen: Ander eta Xenki datu basearen kudeaketaz arduratu ziren (SQL aginduak, datu baseko datuak gure tauletan kargatu...) eta bitartean Leire eta Unai alde grafikoarekin arduratu dira. Swing eta Java erabiliz lehio guztiak sortu genituen. Nahiz eta hemen arazo batzuk izan (ez baitgenituen Swing-ek eskaintzen dizkigun harraminta guztiak ezagutzen) internet eta Juanan en laguntzarekin lortu genuen lehio guztiak ganoraz egitea.

## EKE

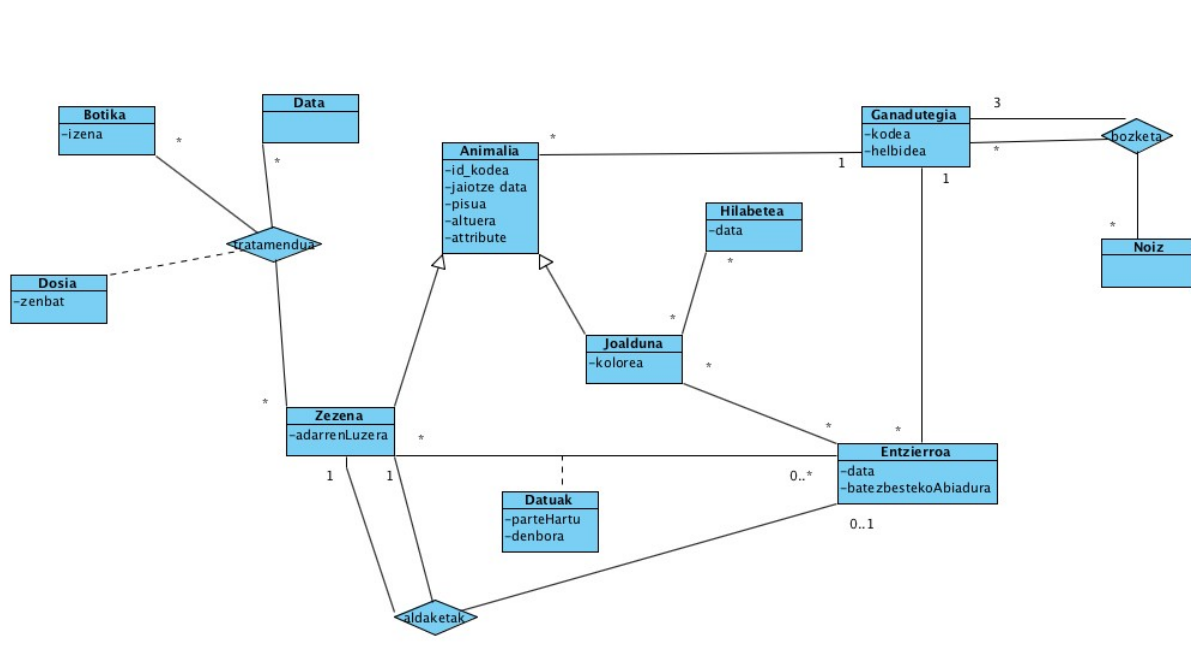
Gure aplikazioaren erabilpen kasuak ondorengo diagrama honetan azaltzen dira:

Visual Paradigm Standard Edition (Universidad del País Vasco)



## DOMEINUAREN EREDUA

Visual Paradigm v11.1 programaz baliatu gara domeinuaren eredua marrazteko. Enuntziatua jarraituz behar ziren klase guztiak sortu ditugu eta bertan eskatzen zituen lotura guztia marraztu ditugu. Ondoren aurkezten den irudian ikus daiteko domeinuaren ereduaren eskema:



Gero eskema hau tauletara pasa beharra dago datu basea sortu ahal izateko. Lehenengo entitate arruntak (Joalduna, Zezena, Botika, Ganadutegia eta Entzierroa). Gero 1:N erlazioak (Animali bakoitzari ganadutegiaren kodea izango den foreign key-a esleitu). N:M erlazioak zeuden bakoitzean taula berri bat sortu dugu eta azkenik N-tar erlazioak (tratamendua, aldaketak eta bozketa). Jarraian datu basea sortzeko sorturiko taulak aurkezten dira:

zezena( id , izenaZezen, jaiotzeData, pisua, altuera, adarrenLuzera, fk\_ganadutegia)

joalduna( id , jaiotzeData, pisua, altuera, kolorea, fk\_ganadutegia)

botika( kodea , izena)

entzierroa( id , luzera, fk\_ganadutegia)

ganadutegia( id , helbidea, izena, arduraduna, tlf)

aldaketak(fk\_aldata, fk\_ordezkoa, fk\_entzierroa)

botoak( fk\_emaila , fk\_hartzailea , data)

erabiltzailea( iderabiltzailea , pasahitza , administratzailea)

jaiak( data , fk\_joalduna)

tratamendua( data , fk\_botika , fk\_zezena , dosia)

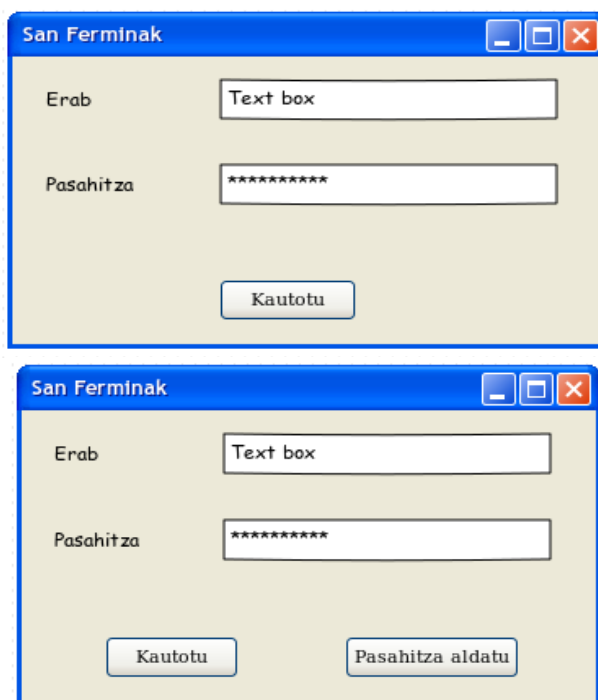
zezenaEntzierroa( fk\_zezena , fk\_entzierroa , parteHartu, denbora)

Hasieran genuen domeinuaren ereduari atributu berri batzuk gehitu dizkiogu eta gainera 'erabiltzailea' taula ere sortu dugu bertan guk ditugun erabiltzaile guztiak gorde ahal izateko.

## **INTERFAZE GRAFIKOAREN ZIRRIBORROA**

Interfaze grafikoaren zirriborroari dagokionez klaseko praktika baten hasi eta gero gure kabuz bukatu dugu Pencil programaz baliatuz. Bertan gero programan beharko ditugun lehio guztiak sortu ditugu bakoitzak beharko dituen botoi, taula, zerrenda... eta guzti. Hau izan da Interfaze grafikoaren zirriborroaren emaitza (esan beharra dago hasiera batean egindako zirriborroak ez zigula guztiz balio izan gauza batzuk falta batziren).

Hasiera batean Kautotu botoia genuen soilik, geroago Pasahitza aldatzeko botoia gehitu genion.



## Administratzailearen lehioak:

San Ferminen kudeaketa ADMINISTRATZAILEA:leire

Menua Erabiltzaileak

Entzierroak **Ganadutegiak** Botikak

Izena	Arduraduna	Telefono Zenbakia
-------	------------	-------------------

Sortu ganadutegi berria

Autatutakoa Ezabatu

Aldaketak Gorde

Sortu Ganadutegi Berria

Izena:

Helbidea:

Telefonoa:

Gorde

**Gehitu Ganadutegia**

Kodea:

Izena:

Helbidea:

Arduraduna:

Telefona:

**Gehitu**

Ganadutegiak   Entzierroak   Botikak

Noiz   Ganadutegia


Sortu entzierro berria

Datuak aldatu

Aldaketak gorde

**Sortu entzierro berria**

Noiz   Ganadutegia esleitu

<<	<	[Close]	>	>>		
October 2008						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Selection...

Gorde



Ganadutegiak

Entzierroak

Botikak

Animali Id

Botika

Alta

Alta Eman

Baja Eman

Aldaketak

Erabiltzailearen lehioak:

San Fermin kudeaketa ERABILTZAILEA:

Abereak

Entzierroko zezenak kudeatu

Botikak

Bozkak

Kodea	Jaiotze data	Pisua	Altuera	Adar Luzera
101	2014/05/15	500kg	1.3m	50cm
102	2014/07/15	520kg	1.5m	60cm

Abereak ezabatu

Aberea gehitu

San Ferminen kudeaketa ERABILTZAILEA: unai

Menua

Abereak

Entzierroko Zezenak Kudeatu

Tratamenduak

Bozkak

Kodea	Izena	Jaiotze data	Pisua	Altuera	Adar luzera
1000	Islero	2013-09-10	250	2.3	1.02
1001	Raton	2013-10-15	256	2.4	1.52
1002	Atrevido	2013-11-06	302	2.1	1.06
1003	Belador	2012-01-26	223	2.56	1.32
1004	Sedoso	2013-10-10	285	2.4	0.98
1005	Cimbareto	2013-02-20	260	2.24	1.1
1006	Destefido	2013-03-10	280	2.01	1.25

Aberea Ezabatu

Aberea Gehitu

Kodea	Jaiotze data	Pisua	Altuera	Kolorea
100	2013-12-10	250	2.3	gorria
101	2013-03-15	256	2.4	beltza
102	2013-10-06	302	2.1	marroia
103	2012-09-26	223	2.56	zuria
104	2013-02-10	285	2.4	beltza
105	2013-07-20	260	2.24	marroia

**Gehitu aberea**

**Gehitu zezena**

Kodea:

Jaiotze data:

Pisua:

Altuera:

Adar Luzera:

**Gehitu zezena**

Kodea:

Izena:

Adar luzeera:

Pisua:

Altuera:

JaiotzeData:
 

December 2014
 

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
49		1	2	3	4	5	6
50	7	8	9	10	11	12	13
51	14	15	16	17	18	19	20
52	21	22	23	24	25	26	27
01	28	29	30	31			

Gehitu joalduna

Kodea:

Jaiotze data:

Pisua:

Altuera:

Kolorea:

Gorde

Gehitu joalduna

Kodea:

Izena:

Kolorea:

Pisua:

Altuera:

JaiotzeData:

December

2014

Sun

Mon

Tue

Wed

Thu

Fri

Sat

49

1

2

3

4

5

6

50

7

8

9

10

11

12

13

51

14

15

16

17

18

19

20

52

21

22

23

24

25

26

27

01

28

29

30

31

Gehitu

San Fermin kudeaketa ERABILTZAILEA:

Botikak Abereak Entzierroko zezenak kudeatu Bozkak

Data	Botika	Dosia	zezena
2014/05/15	Ibuprofeno	1gr	Betxantziñoi
2014/05/28	Aspirina	3gr	Gorri

Botikak kudeatu

San Ferminen kudeaketa ERABILTZAILEA: unai

Menua

Abereak Entzierroko Zezenak Kudeatu Tratamenduak Bozkak

Botika	Dosia	Zezena	Data
Clembuterol		1 Sedoso	Apr 21, 2013
Aspirina		2 Raton	Dec 12, 2013
Paracetamol		7 Belador	Jan 7, 2014
Antitripsina		2 Cimbareto	Mar 3, 2014
Antitripsina		6 Destefido	May 9, 2014
Paracetamol		5 Raton	May 12, 2014
Crotoxina		3 Sedoso	Jun 15, 2014
Tiroxina		5 Sedoso	Jul 2, 2014

Botikak Kudeatu

Tratamendu berria gehitu

Botika: Aspirina

Zezena: Isidro

Dosia:

Noiz:

December 2014

Sun	Mon	Tue	Wed	Thu	Fri	Sat
49	1	2	3	4	5	6
50	7	8	9	10	11	12
51	14	15	16	17	18	19
52	21	22	23	24	25	26
01	28	29	30	31		

Onartu

Tratamendu berria gehitu

Botika: **Paracetamol** ▼

Noiz:

December		2014				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
49		1	2	3	4	5
50	7	8	9	10	11	12
51	14	15	16	17	18	19
52	21	22	23	24	25	26
01	28	29	30	31		

Zezena: **Belador** ▼

Dosia:

**Onartu**

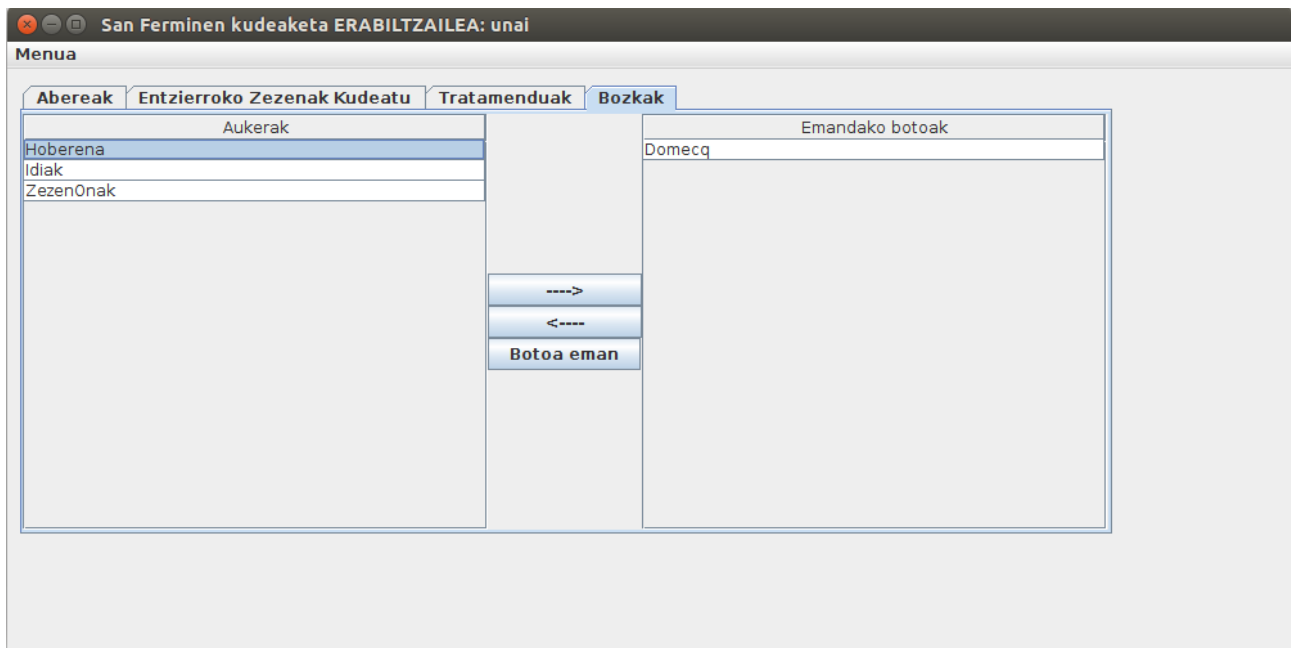
San Fermin kudeaketa ERABILTZAILEA:

Bozkak Abereak Entzieroko zezenak kudeatu Botikak

Autagaiak	Zuk autatutakoak
Iruña	Unaiak
Betimuak	Betizuak

**Botoa gorde**

**Botoa eman**



Zirriborro hauek ikusiz eta gero programa abiaraziz ikus daiteke zirriborroak ez direla programa finaleko berdinak. Proiektuarekin aurrera gindoazen einean aldaketak egin behar izan ditugu. Ez dira oso aldaketa handiak baina erraz ikus daitezke. Esan beharra dago aldaketa asko hoberako izan direla, adibidez menu barra bat gehitu diegu bai administratzaile eta bai erabiltzaileen lehioei bertan aukera batzuk gehituz

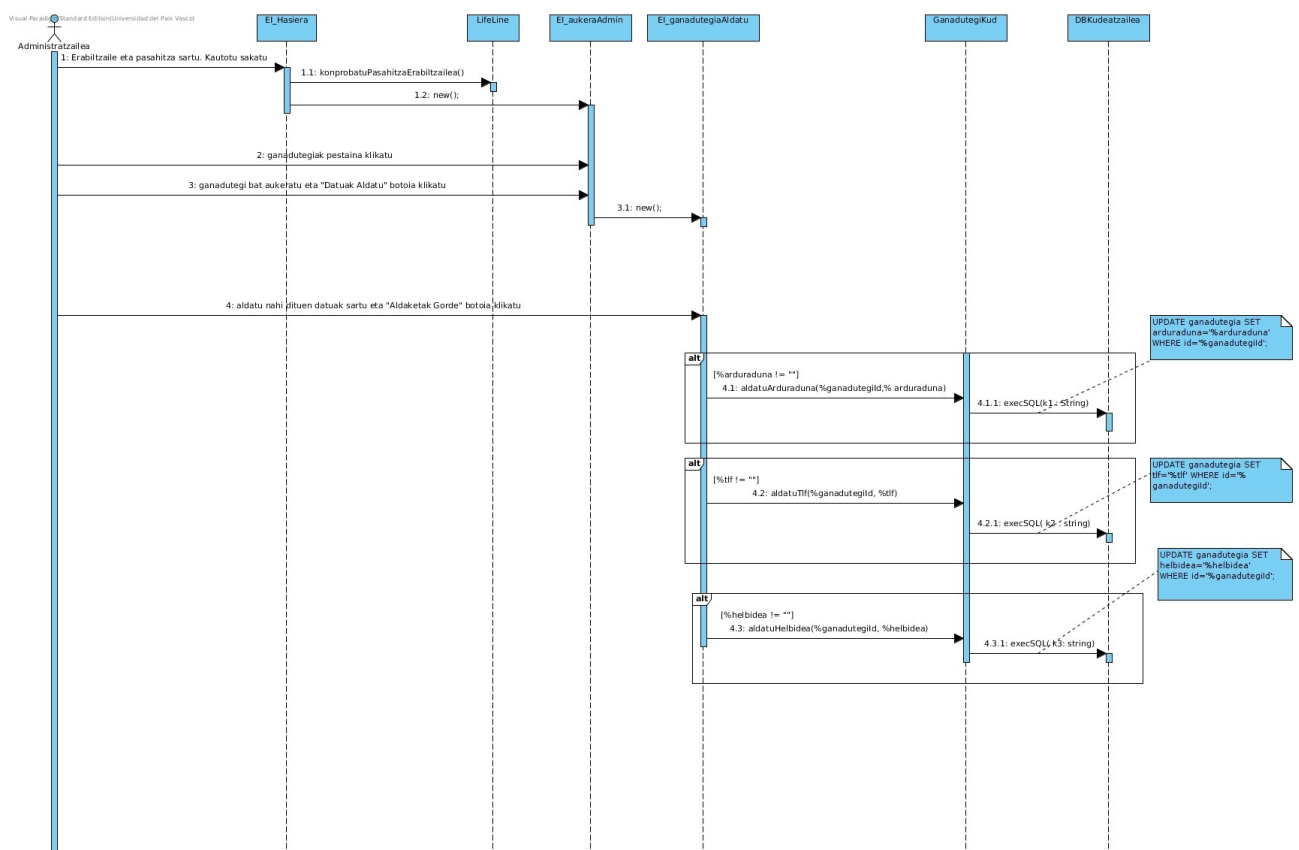
## **SEKUENTZIA DIAGRAMAK**

Jarraian metodo esanguratsuenen sekuentzia diagramak azaltzen dira. Bertan erabiltzailea edo administratzailea kautotzen denetik eta metodo bat bukatu arte exekutatzen duenerainoko sekuentzia azaltzen da.

Entzierro berria sortu:

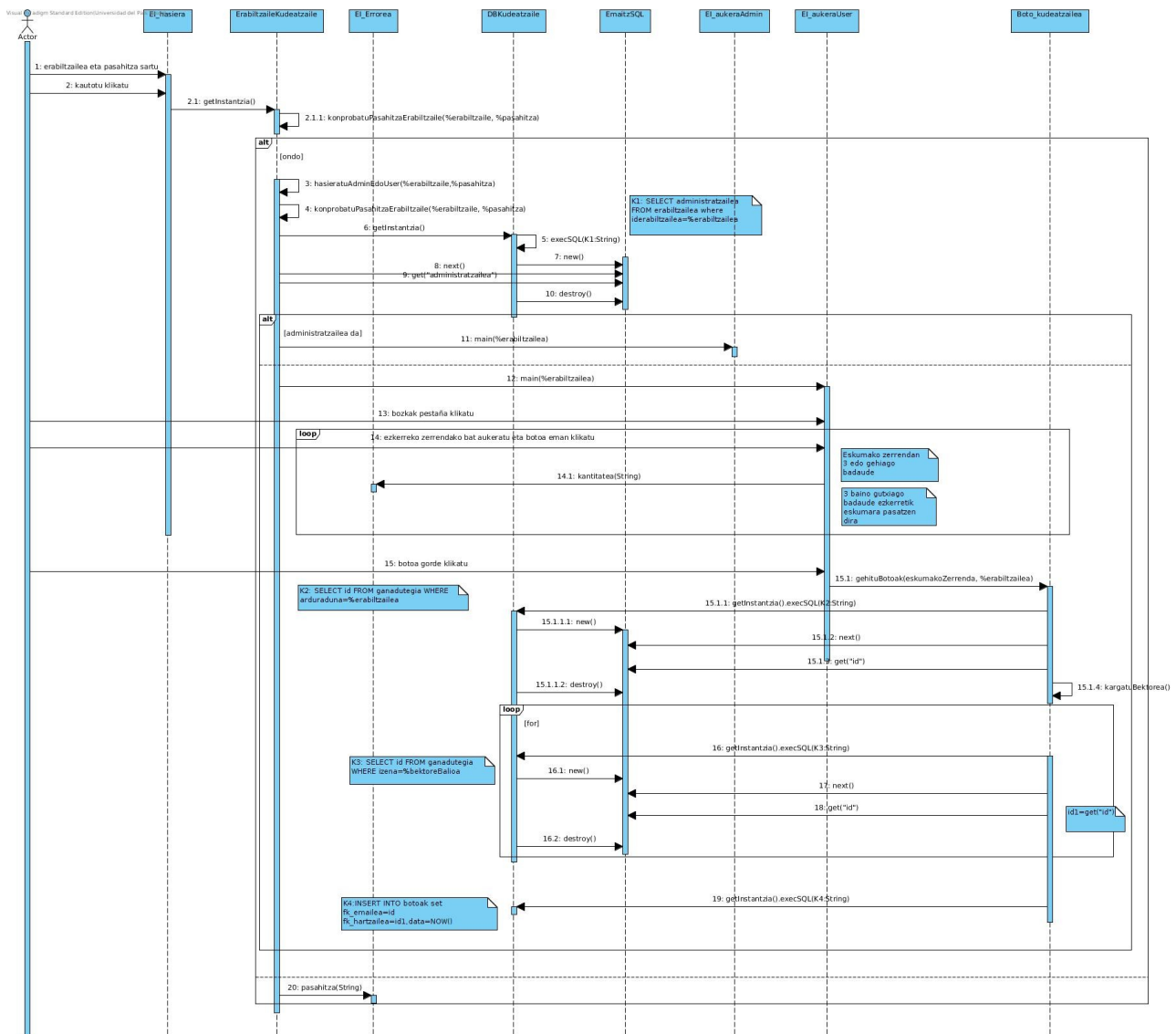
## KonprobatuPasahitzaErabiltzailea()

## ganadutegikoDatuakAldatu()



botoaEman();



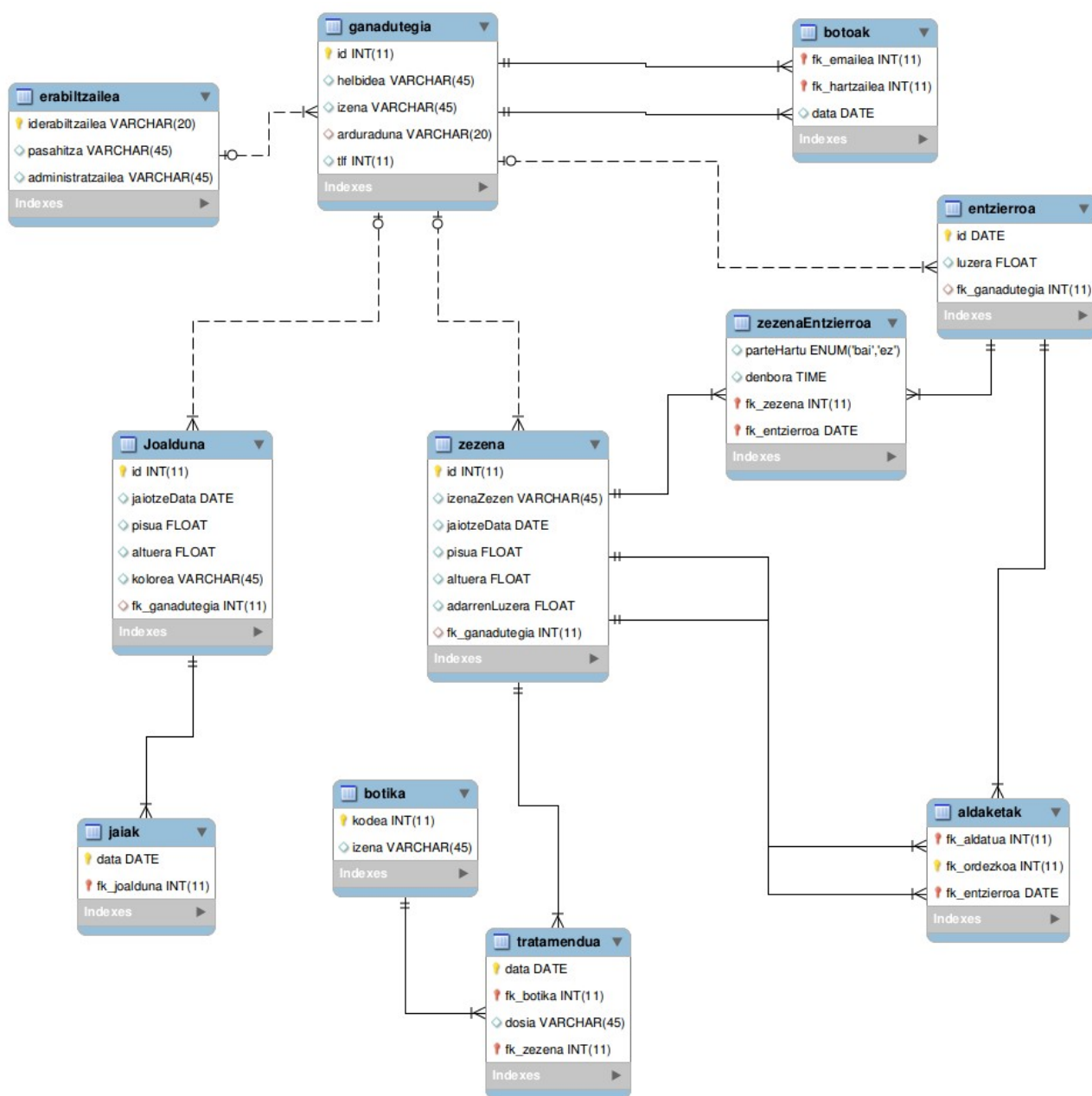


## GITHUB-eko URL-a

Gure praktika egiteko github erabili dugu. Nahiz eta hasiera batean arazoak izan github erabiltzen ikasi dugu denborarekin eta azkenean probetxugarria iruditu zaigu guztioi. Hona hemen gure praktikaren GitHub-eko URL-a:

<https://github.com/unaiguenaga/sanfermin2.0>

## DATU BASEAREN ESKEMA



## KODEA

GitHub-eko URL-an aurkituko duzu aplikazioaren kodea.

## Junit-ak

### PasahitzaAldatuTest

```
package junitak;

import static org.junit.Assert.*;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import Logika.DBKudeatzaile;
import Logika.ErabiltaileKudeatzailea;
import Logika.PasahitzaAldatu;

public class PasahitzaAldatuTest {
    DBKudeatzaile dbk;
    PasahitzaAldatu pa;

    String erab1;
    String pzaharra1;
    String pberri1;
    String pberrikonpr1;
    String kontsulta1;
    String pasahitza1;

    String erab2;
    String pzaharra2;
    String pberri2;
    String pberrikonpr2;
    String kontsulta2;
    String pasahitza2;

    String erab3;
    String pzaharra3;
    String pberri3;
    String pberrikonpr3;
    String kontsulta3;
    String pasahitza3;

    String erab4;
    String pzaharra4;
    String pberri4;
    String pberrikonpr4;
    String kontsulta4;
    String pasahitza4;

    @Before
    public void setUp() throws Exception {
        pa = new PasahitzaAldatu();
        dbk = DBKudeatzaile.getInstantzia();

        erab1 = "unai";
        pzaharra1 = "unai";
```

```

        pberri1 = "guenaga";
        pberrikonpr1 = "guenaga";
        kontsulta1 = "SELECT pasahitza FROM erabiltzailea WHERE
                        iderabiltzailea=" + "'" + erab1 + "'";
        pasahitza1 = null;

        erab2 = "unai";
        pzaharra2 = "guenaga";
        pberri2 = "unai";
        pberrikonpr2 = "unai";
        kontsulta2 = "SELECT pasahitza FROM erabiltzailea WHERE
                        iderabiltzailea=" + "'" + erab2 + "'";
        pasahitza2 = null;

        erab3 = "unai";
        pzaharra3 = "unai";
        pberri3 = "guenaga";
        pberrikonpr3 = "aitor";
        kontsulta3 = "SELECT pasahitza FROM erabiltzailea WHERE
                        iderabiltzailea=" + "'" + erab3 + "'";
        pasahitza3 = null;

        erab4 = "unai";
        pzaharra4 = "txarto";
        pberri4 = "guenaga";
        pberrikonpr4 = "guenaga";
        kontsulta4 = "SELECT pasahitza FROM erabiltzailea WHERE
                        iderabiltzailea=" + "'" + erab1 + "'";
        pasahitza4 = null;
    }

    @After
    public void tearDown() throws Exception {

        erab1 = null;
        pzaharra1 = null;
        pberri1 = null;
        pberrikonpr1 = null;

        erab2 = null;
        pzaharra2 = null;
        pberri2 = null;
        pberrikonpr2 = null;

        erab3 = null;
        pzaharra3 = null;
        pberri3 = null;
        pberrikonpr3 = null;
    }

    @Test
    public void test() {

        // password changed from unai to guenaga.
        pa.pasahitzaAldatu(pberri1, pberrikonpr1, erab1, pzaharra1);
        ResultSet rs = dbk.execSQL(kontsulta1);

        try {

```

```

        rs.next();
        pasahitza1 = rs.getString("pasahitza");
    } catch (SQLException e) {
    }

    assertEquals("guenaga", pasahitza1);
    assertEquals("unai", pasahitza1);

    // password changed from guenaga to unai
    pa.pasahitzaAldatu(pberri2, pberrikonpr2, erab2, pzaharra2);
    rs = dbk.execSQL(kontsulta2);

    try {
        rs.next();
        pasahitza2 = rs.getString("pasahitza");
    } catch (SQLException e) {
    }

    assertEquals("unai", pasahitza2);
    assertEquals("guenaga", pasahitza2);

    // pasahitza and pberrikonpr not same.
    pa.pasahitzaAldatu(pberri3, pberrikonpr3, erab3, pzaharra3);
    rs = dbk.execSQL(kontsulta3);

    try {
        rs.next();
        pasahitza3 = rs.getString("pasahitza");
    } catch (SQLException e) {
    }

    assertEquals("unai", pasahitza3);
    assertEquals("guenaga", pasahitza3);

    // user and password don't match
    pa.pasahitzaAldatu(pberri4, pberrikonpr4, erab4, pzaharra4);
    rs = dbk.execSQL(kontsulta4);

    try {
        rs.next();
        pasahitza4 = rs.getString("pasahitza");
    } catch (SQLException e) {
    }

    assertEquals("unai", pasahitza4);
    assertEquals("guenaga", pasahitza4);
}

}

```

## SortuErabiltzaileaTest

```
package junitak;
```

```

import static org.junit.Assert.*;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import administratzailea.ErabiltzaileBerria;
import Logika.DBKudeatzaile;

public class sortuErabiltzaileTest {

    ErabiltzaileBerria eb;
    DBKudeatzaile dbk;

    String erab;
    String pasahitza;
    String pasahitzaKonpr;
    String kontsulta;
    String ezabatzeko;
    String lag;

    String erab2;
    String pasahitza2;
    String pasahitzaKonpr2;
    String kontsulta2;
    String lag2;

    @Before
    public void setUp() throws Exception {
        eb = new ErabiltzaileBerria();
        dbk = DBKudeatzaile.getInstantzia();
        erab = "periko";
        pasahitza = "palotes";
        pasahitzaKonpr = "palotes";
        kontsulta = "Select pasahitza from erabiltzailea where
                                iderabiltzailea='" + erab + "'";
        ezabatzeko = "delete from erabiltzailea where iderabiltzailea='" +
                                erab + "'";
        lag = null;

        erab2 = "mikel";
        pasahitza2 = "ondo";
        pasahitzaKonpr2 = "txarto";
        kontsulta2 = "select pasahitza from erabiltzailea where
                                iderabiltzailea='" + erab2 + "'";
        lag2 = null;
    }

    @After
    public void tearDown() throws Exception {
        eb = null;
        dbk = null;
        erab = null;
        pasahitza = null;
        pasahitzaKonpr = null;
    }
}

```

```

        kontsulta = null;
        lag = null;
    }

    @Test
    public void test() {
        eb.sortu(erab, pasahitzaKonpr, pasahitzaKonpr);
        ResultSet rs = dbk.execSQL(kontsulta);
        try {
            rs.next();
            lag = rs.getString("pasahitza");
        } catch (SQLException e) {
        }
        assertEquals("palotes", lag);
        dbk.execSQL(ezabatzeko);

        eb.sortu(erab2, pasahitza2, pasahitzaKonpr2);
        rs = dbk.execSQL(kontsulta2);
        try {
            rs.next();
            lag2 = rs.getString("pasahitza");
        } catch (SQLException e) {
        }
        assertEquals(null, lag2);
    }
}

```

#### ZezenBerriaSortu

```

import static org.junit.Assert.*;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

import Logika.DBKudeatzaile;
import user.ZezenKud;

public class SortuZezenaTest {
    private DBKudeatzaile dbk;
    private ZezenKud zk;
    private Integer kodea;
    private String izena;
    private String jaiotzeData;
    private String adarLuzera;
    private String pisua;
    private String altuera;
    private String ganadutegiKod;
    private String kontsulta;
    private String konprobatzeko;
    private String ezabatzeko;

    @Before
    public void setUp() throws Exception {

```

```

        zk = ZezenKud.getInstantzia();
        dbk = DBKudeatzaile.getInstantzia();
        kodea = 2050;
        izena = "Toothless";
        jaiotzeData="2012-05-6";
        adarLuzera = "0.0";
        pisua = "400.0";
        altuera = "1.2";
        ganadutegiKod = "5";
        kontsulta = "select izena from zezena where fk_ganadutegia='"+5+"'";
        ezabatzeko = "delete from zezena where id = '" + kodea + "'";
    }

    @After
    public void tearDown() throws Exception {
        zk=null;
        dbk=null;
        kodea = null;
        izena = null;
        adarLuzera = null;
        pisua = null;
        altuera = null;
        ganadutegiKod = null;
        kontsulta = null;
    }

    @Test
    public void test() {
        zk.gehitu(kodea, izena, jaiotzeData, pisua, altuera, adarLuzera,
ganadutegiKod);
        ResultSet rs = dbk.execSQL(kontsulta);

        try {
            rs.next();
            konprobatzeko = rs.getString("izena");
        } catch (SQLException e) {
        }

        assertEquals("Toothless", konprobatzeko);
        dbk.execSQL(ezabatzeko);
    }
}

```



## **LANEAN EMANDAKO ORDUAK**

Proiektu hau egiteko sartu ditugun orduak azaltzeko, kontuan izan behar dugu bakarkako lana eta taldean batu garenean egin dugun lana. Abenduaren 1ean hasi ginen lanarekin (gutxi gora behera) eta ordutik eta proiektua bukatu harte gutxienez astean 4 aldiz batu gara guztion artean lan egiteko.

Leire eta Unai astelehenera klasea bukatu eta gero 14:00etatik 16:30ak arte egoten ginen gutxi gora behera. Asteartetik 14:00etatik 17:30ak arte. Asteazkenean 14:00etatik 17:30etara. Ostegunetik berriz 12:00etatik 16:30ak arte. Ostiralean Xenki eta Leire 12:00etatik 14:00etara geratzen ziren lanean. Esan genezake astean gutxienez 16 ordu ematen ditugula talde lanean. Esan beharra dago ahal genuen guztietan denok batzen ginelako ordu hauetan, baina askotan beste arloetako lanak, klase orduak... zirela medio ez zaigu posible izan.

Honi bakarkako lana gehitu behar diogu, hau da, etxera heldutakoan bakoitzak bere kabuz aurreratu duen lana. Kalkulatzen dugu, beti ere gutxi gora-behera, bakoitzak 10 ordu egin dituela lanean bere kabuz astean zehar. Gainera abenduaren 9an (fakultateko jai eguna) Unaien etxean batu ginen guztiok goizeko 10etatik arratsaldeko lauak arte.

Proiektuarekin 3 aste eman ditugu eta 78 orduko lana suposatzen digu (kontuan izan 42 ordu talde lanean izan direla eta 36 ordu bakoitzak bere kabuz, eta guztia gutxi gora-beherakoa dela).

## **ONDORIOAK**

Proiektu hau erabilgarria iruditu zaigu etorkizunerako. Gure lanean erabiliko ditugun harrimintak ikasi ditugula iruditzen zaigu. Proiektu polita izan da, gainera gradua sartzean beti pentsatzen dugu aplikazio bat egitearekin, eta oraingoan bat egitea posible izan da. Nahiz eta lanarekin berandu hasi ginen, proiektuarekin emandako 3 asteetan gogotsu aritu gara. Taldean arazoak izan ditugu, haserre txikiak ere egon dira... kostata baina bukatu dugu.

## **ERABILITAKO DOKUMENTAZIOA**

Proiektu hau aurrera eramateko erabili ditugun baliabideak ondokoak dira:

Java Platform, Standard Edition 7 API Specification (<http://docs.oracle.com/javase/7/docs/api/>). Hemen java hizkuntzan programatzerakoan izan ditugun arazoei irtenbideak aurkitu ditugu. Erabitzen genituen Java klaseak erabili ahal izateko informazioa eskuragarri izan dugu web orrialde honen bitartez.

Honetaz gain <https://egela.ehu.es/> -n daukagun Informazio Sistemen Analisia eta Diseinua atalean irakasleak igotako gardenkietaz ere baliatu gara. Kontzeptu nagusiak aurkitu ditugu bertan. Gainera praktikarako erabili ditugun programa asko bertatik eskuratu ditugu (Visual Paradigm, github erabiltzeko tutoriala...)