CRUD

- Llegir

Al metode llegirEquip inicio la connexio i uso el QBE per buscar los equips que hi han a l'arxiu equips.yap i dins del for inserto el valor del equip a la coleccio dades.

```
}

public void llegirEquip() throws IOException {
    File file1 = new File("equips.yap");
    String fullPath = file1.getAbsolutePath();
    db = Db4oEmbedded.openFile(fullPath);
    ObjectSet<Equip> os = db.queryByExample(new Equip(null, 0, 0, 0, 0, 0, 0, 0));
    Equip obj;
    for (Equip p : os) {
        System.out.println(p.get1_nom());
        Model.<Equip>insertar(p, Model.getDades());
    }

}
```

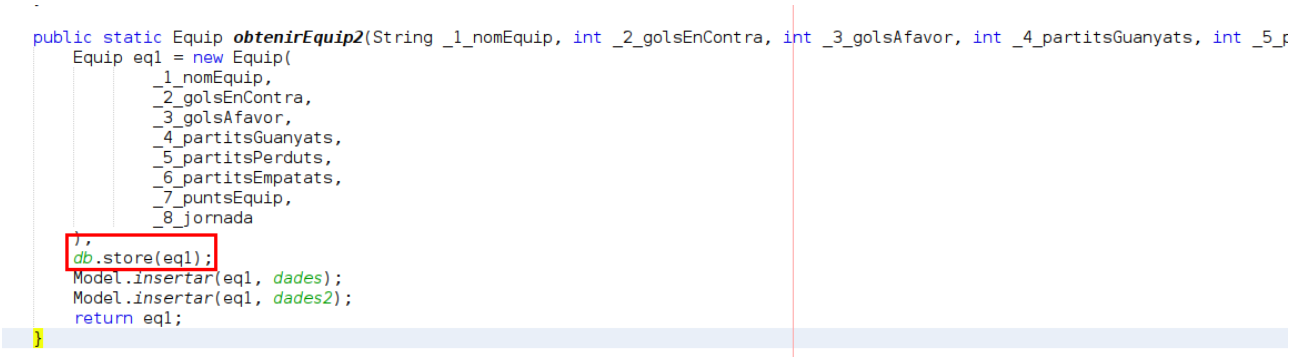
Al metode llegirJugador uso tambe el QBE per buscar els jugadors que hi han a l'arxiu equips.yap i dins del for inserto el valor del jugador a la coleccio dades.

```
public void llegirJugador() throws IOException {
        ObjectSet<Jugador> os = db.queryByExample(new Jugador(null, null, null, 0, 0));
        Jugador obj;
        for (Jugador p : os) {
            if (p.get2_equip() != null) {
                for (Equip value : Model.dades) {
                    if (value.get1_nom().equals(p.get2_equip().get1_nom())) {
                        value._9_jug.add(p);
                    }
                }
            }
            Model.<Jugador>insertar(p, Model.getDadesJugador());
        }
}
```

- Crear

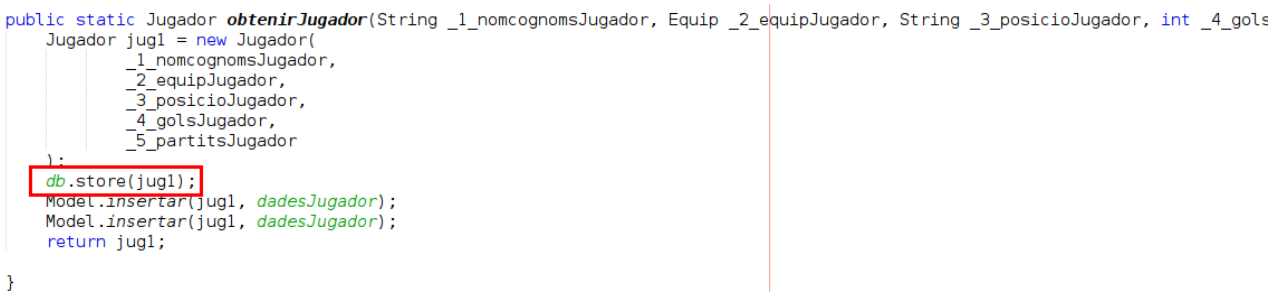
Equip:

Al metode obtenirEquip2 ens passen el valor de tots els camps d'equip que introdueix l'usuari i amb aquestes dades fem un Equip eq1 = new Equip(dades que ha introduit l'usuari); aixi que simplement fem db.store(eq1) per guardar l'equip a la base de dades.



Jugador:

Al metode obtenirJugador ens passen el valor de tots els camps de jugador que introdueix l'usuari i amb aquestes dades fem un Jugador jug1 = new Jugador(dades que ha introduit l'usuari); aixi que simplement fem db.store(jug1) per guardar el jugador a la base de dades.



- Borrar

Tinc 2 metodes un que es diu borrarEquip i l'altre que es diu borrarJugador aquí el que faig es em passen el valor de l'equip o jugador al que borrar i amb aquest valor simplement faig un db.delete(objecte).

```
public static void borrarEquip(Equip eq1) {
    db.delete(eq1);
    for (Jugador j : eq1.get9_jug()) {
        j.set2_equip(null);
        db.store(j);
    }
    dades.remove(eq1);
    dades2.remove(eq1);
}

public static void borrarJugador(Jugador j1) {
    db.delete(j1);
    if (j1.get2_equip() != null) {
        j1.get2_equip().get9_jug().remove(j1);
    }
    dadesJugador.remove(j1);
    dadesJugador2.remove(j1);
}
```

- Editar

Aqui el que fem es ens passen el valor del jugador o equip que vol editar l'usuari i fem un objecte.setCamp1 seguidament fem db.store(objecte) despres fem objecte.setCamp2 despres fem un db.store(objecte) i aixi amb tots els camps que tinguessim de jugador o del equip.

```
public static void updateJugador(Jugador j1, Equip eq1, String nom, String nomActualitzat, String nomEquip, String posicio, in
    j1.set1_nomcognoms(nomActualitzat);
    db.store(j1);
    j1.set2_equip(eq1);
    db.store(j1);
    j1.set3_posicio(posicio);
    db.store(j1);
    j1.set4_gols(gols);
    db.store(j1);
    j1.set5_partits(partits);
    db.store(j1);

}
```

```
public static void updateEquip(Equip obj, String nomActualitzat, int gols_en_contra, int gols_afavor, int partits_guanyats, in
    obj.set1_nom(nomActualitzat);
    db.store(obj);
    obj.set2_golsEnContra(gols_en_contra);
    db.store(obj);
    obj.set3_golsAfavor(gols_afavor);
    db.store(obj);
    obj.set4_partitsGuanyats(partits_guanyats);
    db.store(obj);
    obj.set5_partitsPerduts(partits_perduts);
    db.store(obj);
    obj.set6_partitsEmpatats(partits_empatats);
    db.store(obj);
    obj.set7_punts(punts);
    db.store(obj);
    obj.set8_jornada(jornada);
    db.store(obj);
}
```

Tancar Connexio

La connexio amb la base de dades la obric al principi i solament la tanco quant donen clic a la X per tancar el programa. Quant tanquen el programa automaticament s'executa aquest metode.

```
public static void tancarConn() {
    db.close();
}
```

```
view.addWindowListener(new java.awt.event.WindowAdapter() {
    @Override
    public void windowClosing(java.awt.event.WindowEvent e) {
        Runtime.getRuntime().addShutdownHook(new Thread() {
            @Override
            public void run() {
                model.tancarConn();
            }
        });
    }
});
```

QBE

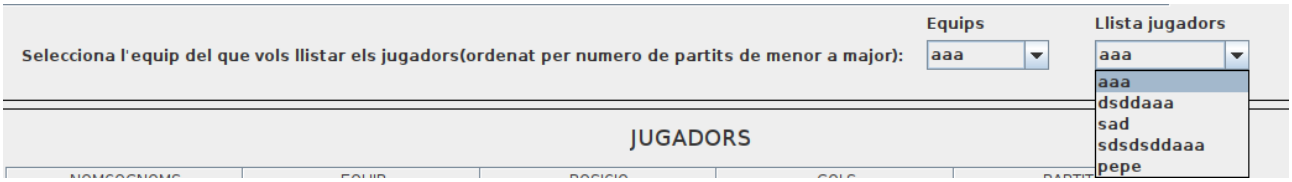
Com has pogut veure als metodes de llegir he usat el QBE. Ha sigut el unic tipus de Query que ha sigut necessaria usar per poder fer el crud.

```
}

public void llegirEquip() throws IOException {
    File file1 = new File("equips.yap");
    String fullPath = file1.getAbsolutePath();
    db = Db4oEmbedded.openFile(fullPath);
    ObjectSet<Equip> os = db.queryByExample(new Equip(null, 0, 0, 0, 0, 0, 0, 0));
    Equip obj;
    for (Equip p : os) {
        System.out.println(p.get1_nom());
        Model.<Equip>insertar(p, Model.getDades());
    }

}
```

```
public void llegirJugador() throws IOException {
        ObjectSet<Jugador> os = db.queryByExample(new Jugador(null, null, null, 0, 0));
        Jugador obj;
        for (Jugador p : os) {
            if (p.get2_equip() != null) {
                for (Equip value : Model.dades) {
                    if (value.get1_nom().equals(p.get2_equip().get1_nom())) {
                        value._9_jug.add(p);
                    }
                }
            }
            Model.<Jugador>insertar(p, Model.getDadesJugador());
        }
}
```

Natives

He ficat dos combobox a la vista i si selecciones un equip del combobox d'equips automaticament es carregara el combobox del costat amb tots els jugadors d'aquell equip.



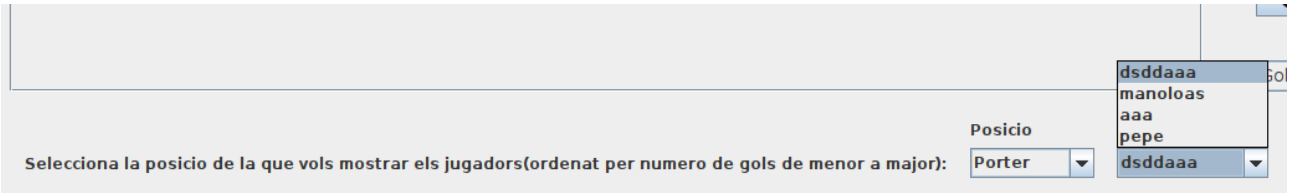
Per a que aixo funcioni he creat el metode sqlNative al Model on em passen el valor de l'equip que selecciona l'usuari i fent una cerca nativa inserta els jugadors d'aquell equip en concret en una coleccio que he creat especialment per fer aixo que es diu dadesSql. I al controlador el que faig es mirar quant em seleccionen algo al combobox d'equips i declaro una variable d'equip i li dono el valor que te el que esta seleccionat i seguidament uso el metode sqlNative i li passo el valor de l'equip i despres que la coleccio ja estara carregada amb el valor dels jugadors executo el loadCombo per carregar el combobox amb els jugadors d'aquell equip.

```
view.getjComboBox3().addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Equip x = (Equip) view.getjComboBox3().getSelectedItem();
        Model.sqlNative(x);
        Utils.<Jugador>loadCombo(model.getDadesSql(), view.getjComboBox4());
    }
});
```

```
public static void sqlNative(Equip eq1) {
    dadesSql.clear();
    ObjectSet<Jugador> os = db.query(new Predicate<Jugador>() {
        @Override
        public boolean match(Jugador et) {
            return et.get2_equip().toString().equals(eq1.get1_nom());
        }
    },
            new JugadorOrdena2()
    );
    for (Jugador p : os) {
        System.out.println(p);
        Model.insertar(p, dadesSql);
    }
}
```

SODA

He fet un combobox amb les posicions i al seleccionar una posicio automaticament t'ompli el combobox amb els jugadors d'aquella posicio. Aixo ho he fet usant SQL de tipus SODA busco els jugadors que tenen aquella posicio i els guardo en un objectSet i despres amb un for recorro el resultat i inserto les dades en una coleccio que he fet expressament.




```
public static void sqlSoda(String posicio) {
    dadesSqlSoda.clear();
    Comparator<Jugador> cmp= new Comparator<Jugador>(){
      public int compare(Jugador o1, Jugador o2) {
    return o1.get5_partits()-o2.get5_partits();
}
    };
    Query query = db.query();
    query.constrain(Jugador.class);
    query.descend("_3_posicio").constrain(posicio);
    query.sortBy(cmp);
    ObjectSet<Jugador> result = query.execute();

    for (Jugador p : result) {
        System.out.println(p);
        Model.insertar(p, dadesSqlSoda);
```

```
view.getjComboBox2().addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String x = view.getjComboBox2().getSelectedItem().toString();
        Model.sqlSoda(x);
        Utils.<Jugador>loadCombo(model.getDadesSqlSoda(), view.getjComboBox5());
    }
});
```