

<b>ACTIVITAT</b>
<b>Objectius:</b> <ul style="list-style-type: none"><li>- Saber com definir objectes DAO i com ignorar aquest patró</li></ul>
<b>Instruccions:</b> <ul style="list-style-type: none"><li>- Es tracta d'un treball individual, no s'admet cap tipus de còpia.</li><li>- Responen a l'espai de cada pregunta, si ho feu amb diapositives enganxeu la diapositiva en aquest mateix espai.</li><li>- Es valorarà la cura en la presentació del document i que segueixi l'estructura indicada.</li></ul>
<b>Criteris d'avaluació:</b> <ul style="list-style-type: none"><li>- Cada pregunta té el mateix pes sobre 90%</li><li>- Les metodologies de treball, organització personal i participació contenen un 10%</li></ul>
<b>Entrega:</b> <ul style="list-style-type: none"><li>- Aquest document amb les explicacions i captures necessàries i els arxius adjunts necessaris del codi que es demana</li><li>- El nom dels arxius adjunts a entregar serà: nomicognom-nomicognom.zip</li></ul>

**Noms i Cognoms:** Joel Berzal Álamo

**Materials:**

Necessiteu un entorn de desenvolupament en JAVA

Feu servir Google per buscar els tutorials que us serveixin millor

Teniu més informació [aquí](#)

**Tasques:**

- **Exercici 0** - Crea un programa “Main.java” que implementi la gestió de dades (en arxius .json) d’un apassionat per la programació. La base de dades estarà en una carpeta anomenada ‘dbProgramacio’ i tindrà els següents arxius .json:

- llenguatges.json, conté una llista amb almenys 5 llenguatges de programació amb atributs:
  - id (int, identificador del llenguatge a la base de dades)
  - nom (String, nom del llenguatge)
  - any (int, any en què va aparèixer)
  - dificultat (String dificultat de programar-lo entre fàcil, mitja i difícil)
  - popularitat (int per ordenar-los segons popularitat)

```
dbProgramacio > {} llenguatges.json > ...
1  {
2    {
3      "popularitat": 8,
4      "id": 0,
5      "nom": "C",
6      "any": 1972,
7      "dificultat": "difícil"
8    },
9    {
10     "popularitat": 7,
11     "id": 1,
12     "nom": "C++",
13     "any": 1983,
14     "dificultat": "mitja"
15   },
16   {
17     "popularitat": 2,
18     "id": 2,
19     "nom": "Objective C",
20     "any": 1984,
21     "dificultat": "difícil"
22   },
23   {
24     "popularitat": 6,
25     "id": 3,
26     "nom": "JavaScript",
27     "any": 1996,
28     "dificultat": "fàcil"
29   },
30   {
31     "popularitat": 5,
32     "id": 4,
33     "nom": "Java",
34     "any": 1995,
35     "dificultat": "mitja"
36   }
37 }
```

- eines.json, conté una llista amb almenys 5 eines de programació amb atributs
  - id (int, identificador de la eina la base de dades)
  - nom (String, nom de la eina)
  - any (int, any en què va aparèixer)
  - llenguatges (arraylist amb els identificadors dels llenguatges que accepta)

```
dbProgramacio > {} eines.json > ...
1
2 {
3   "llenguatges": [
4     0,
5     1,
6     2,
7     3,
8     4
9   ],
10  "id": 0,
11  "nom": "Visual Studio Code",
12  "any": 2015
13 },
14 {
15   "llenguatges": [
16     0,
17     1,
18     2
19   ],
20  "id": 1,
21  "nom": "Xcode",
22  "any": 2003
23 },
24 {
25   "llenguatges": [4],
26  "id": 2,
27  "nom": "Eclipse",
28  "any": 2001
29 },
30 {
31   "llenguatges": [
32     0,
33     1,
34     2,
35     3,
36     4
37   ],
38  "id": 3,
39  "nom": "Vim",
40  "any": 1991
41 },
42 {
43   "llenguatges": [
44     0,
45     1,
46     2,
47     3,
48     4
49   ],
50  "id": 4,
51  "nom": "Nano",
52  "any": 1999
53 }
54 }
```

- software.json, conté una llista amb almenys 10 programes (o software) coneguts, desenvolupats amb els llenguatges i les eines anteriors
  - id (int, identificador del software a la base de dades)
  - nom (String, nom del software)
  - any (int, any en què va aparèixer)
  - llenguatges (arraylist amb els identificadors dels llenguatges amb què s'ha desenvolupat)

```
dbProgramacio > {} software.json > ...
1  {}
2  {
3    "llenguatges": [
4      0,
5      1,
6      2,
7      3,
8      4
9    ],
10   "id": 0,
11   "nom": "Visual Studio Code",
12   "any": 2015
13 },
14 {
15   "llenguatges": [
16     0,
17     1,
18     2
19   ],
20   "id": 1,
21   "nom": "Xcode",
22   "any": 2003
23 },
24 {
25   "llenguatges": [4],
26   "id": 2,
27   "nom": "Eclipse",
28   "any": 2001
29 },
30 {
31   "llenguatges": [
32     0,
33     1,
34     2,
35     3,
36     4
37   ],
38   "id": 3,
39   "nom": "Vim",
40   "any": 1991
41 },
42 {
43   "llenguatges": [
44     0,
45     1,
46     2,
47     3,
48     4
49   ],
50   "id": 4,
51   "nom": "Nano",
52   "any": 1999
53 }
54 }
```

Un cop tinguis aquesta base de dades:

- Crea els objectes Java que corresponent a cada tipus d'objecte anterior (ObjEina.java, ObjLlenguatge.java i ObjSoftware.java)

Per poder crear l'objecte "ObjLlenguatge.java", primer he hagut de crear cinc variables, les quals corresponen a l'id, el nom, l'any, la dificultat i la popularitat dels llenguatges de programació.

```
3 public class Objllenguatge {  
4  
5     private int id, any, popularitat;  
6     private String nom, dificultat;
```

A continuació, he hagut de crear un constructor que instancia les cinc variables anteriors.

```
8     public Objllenguatge(int id, String nom, int any, String dificultat, int popularitat) {  
9         this.id = id;  
10        this.nom = nom;  
11        this.any = any;  
12        this.dificultat = dificultat;  
13        this.popularitat = popularitat;  
14    }
```

Després, he hagut de crear els getters i els setters de les cinc variables anteriors.

```
16    public int getId() {  
17        return id;  
18    }  
19  
20    public void setId(int id) {  
21        this.id = id;  
22    }  
23  
24    public String getNom() {  
25        return nom;  
26    }  
27  
28    public void setNom(String nom) {  
29        this.nom = nom;  
30    }  
31  
32    public int getAny() {  
33        return any;  
34    }  
35  
36    public void setAny(int any) {  
37        this.any = any;  
38    }  
39  
40    public String getDificultat() {  
41        return dificultat;  
42    }  
43  
44    public void setDificultat(String dificultat) {  
45        this.dificultat = dificultat;  
46    }  
47  
48    public int getPopularitat() {  
49        return popularitat;  
50    }  
51  
52    public void setPopularitat(int popularitat) {  
53        this.popularitat = popularitat;  
54    }
```

Finalment, he hagut de crear el mètode toString() d'aquest objecte.

```
56    @Override  
57    public String toString() {  
58        return "Llenguatge: " + this.id + " " + this.nom + ", " + this.any + " - " + this.dificultat + "/" + this.popularitat;  
59    }  
60 }
```

Per poder crear l'objecte "ObjEina.java", primer he hagut de crear quatre variables, les quals corresponen a l'id, el nom, l'any, i els llenguatges de programació de les eines.

```
5 public class ObjEina {  
6  
7     private int id, any;  
8     private String nom;  
9     private ArrayList<Integer> llenguatges;
```

A continuació, he hagut de crear un constructor que instancia les quatre variables anteriors.

```
11 public ObjEina(int id, String nom, int any, ArrayList<Integer> llenguatges) {  
12     this.id = id;  
13     this.nom = nom;  
14     this.any = any;  
15 }
```

Després, he hagut de crear els getters i els setters de les quatre variables anteriors.

```
17 public int getId() {  
18     return id;  
19 }  
20  
21 public void setId(int id) {  
22     this.id = id;  
23 }  
24  
25 public String getNom() {  
26     return nom;  
27 }  
28  
29 public void setNom(String nom) {  
30     this.nom = nom;  
31 }  
32  
33 public int getAny() {  
34     return any;  
35 }  
36  
37 public void setAny(int any) {  
38     this.any = any;  
39 }  
40  
41 public ArrayList<Integer> getllenguatges() {  
42     return llenguatges;  
43 }  
44  
45 public void setllenguatges(ArrayList<Integer> llenguatges) {  
46     this.llenguatges = llenguatges;  
47 }
```

Finalment, he hagut de crear el mètode toString() d'aquest objecte.

```
50 @Override  
51 public String toString() {  
52     return "Eina: " + this.id + " " + this.nom + ", " + this.any + " - " + this.llenguatges;  
53 }  
54 }
```

Per poder crear l'objecte "ObjSoftware.java", primer he hagut de crear quatre variables, les quals corresponen a l'id, el nom, l'any i els llenguatges de programació dels softwares.

```
5 public class ObjSoftware {  
6  
7     private int id, any;  
8     private String nom;  
9     private ArrayList<Integer> llenguatges;
```

A continuació, he hagut de crear un constructor que instancia les quatre variables anteriors.

```
11 public ObjSoftware(int id, String nom, int any, ArrayList<Integer> llenguatges) {  
12     this.id = id;  
13     this.nom = nom;  
14     this.any = any;  
15 }
```

Després, he hagut de crear els getters i els setters de les quatre variables anteriors.

```
17 public int getId() {  
18     return id;  
19 }  
20  
21 public void setId(int id) {  
22     this.id = id;  
23 }  
24  
25 public String getNom() {  
26     return nom;  
27 }  
28  
29 public void setNom(String nom) {  
30     this.nom = nom;  
31 }  
32  
33 public int getAny() {  
34     return any;  
35 }  
36  
37 public void setAny(int any) {  
38     this.any = any;  
39 }  
40  
41 public ArrayList<Integer> getllenguatges() {  
42     return llenguatges;  
43 }  
44  
45 public void setllenguatges(ArrayList<Integer> llenguatges) {  
46     this.llenguatges = llenguatges;  
47 }
```

Finalment, he hagut de crear el mètode toString() d'aquest objecte.

```
50 @Override  
51 public String toString() {  
52     return "Software: " + this.id + " " + this.nom + ", " + this.any + " - " + this.llenguatges;  
53 }  
54 }
```

- Per cada un d'aquests objectes JAVA crea el Dao corresponent basat en CRUD (DaoEina.java, DaoLlenguatge.java, DaoSoftware.java)

Per poder crear l'objete "DaoLlenguatge.java", primer he hagut de crear el mètode *writeList*, el qual agafa un ArrayList d'objectes tipus "ObjLlenguatge" i desa els seus elements dins d'un arxiu JSON.

```

13 private void writelist(ArrayList<ObjLlenguatge> llista) {
14     try {
15         JSONArray jsonArray = new JSONArray();
16         for (ObjLlenguatge llenguatge : llista) {
17             JSONObject jsonObject = new JSONObject();
18             jsonObject.put("id", llenguatge.getId());
19             jsonObject.put("nom", llenguatge.getNom());
20             jsonObject.put("any", llenguatge.getAny());
21             jsonObject.put("dificultat", llenguatge.getDificultat());
22             jsonObject.put("popularitat", llenguatge.getPopularitat());
23             jsonArray.put(jsonObject);
24         }

```

Tot seguit, he hagut de crear el mètode *getPosition*, el qual cerca la posició d'un element d'un ArrayList d'objectes del tipus "ObjLlenguatge" en funció del seu id i, en trobar-lo, retorna la posició d'aquest element en l'ArrayList.

```

35 private int getPosition(int id) {
36     int result = -1;
37     ArrayList<ObjLlenguatge> llista = getAll();
38     for (int i = 0; i < llista.size(); i++) {
39         ObjLlenguatge llenguatge = llista.get(i);
40         if (llenguatge.getId() == id) {
41             result = i;
42             break;
43         }
44     }
45     return result;
46 }

```

A continuació, he hagut de crear el mètode *add*, el qual agrega un objecte del tipus "ObjLlenguatge" a un ArrayList d'objectes d'aquest mateix tipus al mateix temps que l'actualitza.

```

48 @Override
49 public void add(ObjLlenguatge llenguatge) {
50     ArrayList<ObjLlenguatge> llista = getAll();
51     ObjLlenguatge item = get(llenguatge.getId());
52     if (item == null) {
53         llista.add(llenguatge);
54         writelist(llista);
55     }
56 }

```

Després, he hagut de crear el mètode *get*, el qual obté un objecte del tipus "ObjLlenguatge" d'un ArrayList d'objectes d'aquest mateix tipus en funció del seu id i, en fer-ho, el retorna.

```

58 @Override
59 public ObjLlenguatge get(int id) {
60     ObjLlenguatge result = null;
61     ArrayList<ObjLlenguatge> llista = getAll();
62     int posicio = getPosition(id);
63     if (posicio != -1) {
64         result = llista.get(posicio);
65     }
66     return result;
67 }

```



Posteriorment, he hagut de crear el mètode *getAll*, el qual recupera un ArrayList d'objectes del tipus “ObjLlenguatge” des d'un arxiu i, en fer-ho, la retorna.

```

69  @Override
70  public ArrayList<ObjLlenguatge> getAll() {
71      ArrayList<ObjLlenguatge> result = new ArrayList<>();
72      try {
73          String content = new String(Files.readAllBytes(Paths.get(Main.llenguatgesPath)));
74          JSONArray jsonArray = new JSONArray(content);
75          for (int i = 0; i < jsonArray.length(); i++) {
76              JSONObject jsonObject = jsonArray.getJSONObject(i);
77              int id = jsonObject.getInt("id");
78              String nom = jsonObject.getString("nom");
79              int any = jsonObject.getInt("any");
80              String dificultat = jsonObject.getString("dificultat");
81              int popularitat = jsonObject.getInt("popularitat");
82              result.add(new ObjLlenguatge(id, nom, any, dificultat, popularitat));
83          }
84      } catch (IOException e) {
85          e.printStackTrace();
86      }
87      return result;
88  }

```

Seguidament, he hagut de crear el mètode *update*, el qual actualitza un objecte del tipus “ObjLlenguatge” en un ArrayList d'objectes d'aquest mateix tipus i, en fer-ho, la desa.

```

90  @Override
91  public void update(int id, ObjLlenguatge llenguatge) {
92      ArrayList<ObjLlenguatge> llista = getAll();
93      int posicio = getPosition(id);
94      if (posicio != -1) {
95          llista.set(posicio, llenguatge);
96          writeList(llista);
97      }
98  }

```

Més tard, he hagut de crear el mètode *delete*, el qual elimina un objecte del tipus “ObjLlenguatge” d'un ArrayList d'objectes d'aquest mateix tipus en funció del seu id i, en fer-ho, la desa.

```

100 @Override
101 public void delete(int id) {
102     ArrayList<ObjLlenguatge> llista = getAll();
103     int posicio = getPosition(id);
104     if (posicio != -1) {
105         llista.remove(posicio);
106         writeList(llista);
107     }
108 }

```

Seguit d'això, he hagut de crear el mètode *print*, el qual imprimeix per pantalla la informació continguda en un ArrayList d'objectes del tipus “ObjLlenguatge”.

```

110 @Override
111 public void print() {
112     ArrayList<ObjLlenguatge> llista = getAll();
113     for (int i = 0; i < llista.size(); i++) {
114         System.out.println(" " + llista.get(i));
115     }
116 }

```

Finalment, he hagut de crear els setters de tots els atributs de “ObjLlenguatge”.

```
118     @Override
119     public void setNom(int id, String nom) {
120         ArrayList<ObjLlenguatge> llista = getAll();
121         int posicio = getPosition(id);
122         if (posicio != -1) {
123             ObjLlenguatge llenguatge = llista.get(posicio);
124             llenguatge.setNom(nom);
125             update(id, llenguatge);
126         }
127     }
128
129     @Override
130     public void setAny(int id, int any) {
131         ArrayList<ObjLlenguatge> llista = getAll();
132         int posicio = getPosition(id);
133         if (posicio != -1) {
134             ObjLlenguatge llenguatge = llista.get(posicio);
135             llenguatge.setAny(any);
136             update(id, llenguatge);
137         }
138     }
139
140     public void setDificultat(int id, String dificultat) {
141         ArrayList<ObjLlenguatge> llista = getAll();
142         int posicio = getPosition(id);
143         if (posicio != -1) {
144             ObjLlenguatge llenguatge = llista.get(posicio);
145             llenguatge.setDificultat(dificultat);
146             update(id, llenguatge);
147         }
148     }
149
150     public void setPopularitat(int id, int popularitat) {
151         ArrayList<ObjLlenguatge> llista = getAll();
152         int posicio = getPosition(id);
153         if (posicio != -1) {
154             ObjLlenguatge llenguatge = llista.get(posicio);
155             llenguatge.setPopularitat(popularitat);
156             update(id, llenguatge);
157         }
158     }
```

Per poder crear l'objecte “DaoEina.java”, primer he hagut de crear el mètode *writeList*, el qual agafa un ArrayList d'objectes tipus “ObjEina” i desa els seus elements dins d'un arxiu JSON.

```

13 private void writeList(ArrayList<ObjEina> llista) {
14     try {
15         JSONArray jsonArray = new JSONArray();
16         for (ObjEina eina : llista) {
17             JSONObject jsonObject = new JSONObject();
18             jsonObject.put("id", eina.getId());
19             jsonObject.put("nom", eina.getNom());
20             jsonObject.put("any", eina.getAny());
21             jsonObject.put("llenguatges", eina.getLlenguatges());
22             jsonArray.put(jsonObject);
23         }
24
25         PrintWriter out = new PrintWriter(Main.einesPath);
26         out.write(jsonArray.toString(4)); // 4 es l'espaiat
27         out.flush();
28         out.close();
29     } catch (FileNotFoundException e) {
30         e.printStackTrace();
31     }
32 }

```

Tot seguit, he hagut de crear el mètode *getPosition*, el qual cerca la posició d'un element d'un ArrayList d'objectes del tipus “ObjEina” en funció del seu id i, en trobar-lo, retorna la posició d'aquest element en l'ArrayList.

```

34 private int getPosition(int id) {
35     int result = -1;
36     ArrayList<ObjEina> llista = getAll();
37     for (int i = 0; i < llista.size(); i++) {
38         ObjEina eina = llista.get(i);
39         if (eina.getId() == id) {
40             result = i;
41             break;
42         }
43     }
44     return result;
45 }

```

A continuació, he hagut de crear el mètode *add*, el qual agrega un objecte del tipus “ObjEina” a un ArrayList d'objectes d'aquest mateix tipus al mateix temps que l'actualitza.

```

47 @Override
48 public void add(ObjEina eina) {
49     ArrayList<ObjEina> llista = getAll();
50     ObjEina item = get(eina.getId());
51     if (item == null) {
52         llista.add(eina);
53         writeList(llista);
54     }
55 }

```

Després, he hagut de crear el mètode *get*, el qual obté un objecte del tipus “ObjEina” d'un ArrayList d'objectes d'aquest mateix tipus en funció del seu id i, en fer-ho, el retorna.

```

57 @Override
58 public ObjEina get(int id) {
59     ObjEina result = null;
60     ArrayList<ObjEina> llista = getAll();
61     int posicio = getPosition(id);
62     if (posicio != -1) {
63         result = llista.get(posicio);
64     }
65     return result;
66 }

```

Posteriorment, he hagut de crear el mètode *getAll*, el qual recupera un ArrayList d'objectes del tipus “ObjEina” des d'un arxiu i, en fer-ho, la retorna.

```
68     @Override
69     public ArrayList<ObjEina> getAll() {
70         ArrayList<ObjEina> result = new ArrayList<>();
71         try {
72             String content = new String(Files.readAllBytes(Paths.get(Main.einesPath)));
73             JSONArray jsonArray = new JSONArray(content);
74             for (int i = 0; i < jsonArray.length(); i++) {
75                 JSONObject jsonObject = jsonArray.getJSONObject(i);
76                 int id = jsonObject.getInt("id");
77                 String nom = jsonObject.getString("nom");
78                 int any = jsonObject.getInt("any");
79                 JSONArray jsonLlenguatges = jsonObject.getJSONArray("llenguatges");
80                 ArrayList<Integer> llenguatges = new ArrayList<>();
81                 for (int j = 0; j < jsonLlenguatges.length(); j++) {
82                     llenguatges.add(jsonLlenguatges.getInt(j));
83                 }
84                 ObjEina eina = new ObjEina(id, nom, any, llenguatges);
85                 result.add(eina);
86             }
87         } catch (IOException e) {
88             e.printStackTrace();
89         }
90         return result;
91     }
```

Seguidament, he hagut de crear el mètode *update*, el qual actualitza un objecte del tipus “ObjEina” en un ArrayList d'objectes d'aquest mateix tipus i, en fer-ho, la desa.

```
93     @Override
94     public void update(int id, ObjEina eina) {
95         ArrayList<ObjEina> llista = getAll();
96         int posicio = getPosition(id);
97         if (posicio != -1) {
98             llista.set(posicio, eina);
99             writelist(llista);
100         }
101     }
```

Més tard, he hagut de crear el mètode *delete*, el qual elimina un objecte del tipus “ObjEina” d'un ArrayList d'objectes d'aquest mateix tipus en funció del seu id i, en fer-ho, la desa.

```
103     @Override
104     public void delete(int id) {
105         ArrayList<ObjEina> llista = getAll();
106         int posicio = getPosition(id);
107         if (posicio != -1) {
108             llista.remove(posicio);
109             writelist(llista);
110         }
111     }
```

Seguit d'això, he hagut de crear el mètode *print*, el qual imprimeix per pantalla la informació continguda en un ArrayList d'objectes del tipus “ObjEina”.

```
113     @Override
114     public void print() {
115         ArrayList<ObjEina> llista = getAll();
116         for (int i = 0; i < llista.size(); i++) {
117             System.out.println("    " + llista.get(i));
118         }
119     }
```

Finalment, he hagut de crear els setters de tots els atributs de “ObjEina”.

```
121 @Override
122 public void setNom(int id, String nom) {
123     ArrayList<ObjEina> llista = getAll();
124     int posicio = getPosition(id);
125     if (posicio != -1) {
126         ObjEina eina = llista.get(posicio);
127         eina.setNom(nom);
128         update(id, eina);
129     }
130 }
131
132 @Override
133 public void setAny(int id, int any) {
134     ArrayList<ObjEina> llista = getAll();
135     int posicio = getPosition(id);
136     if (posicio != -1) {
137         ObjEina eina = llista.get(posicio);
138         eina.setAny(any);
139         update(id, eina);
140     }
141 }
142
143 public void setLlenguatgesAdd(int id, int idLlenguatge) {
144     ArrayList<ObjEina> llista = getAll();
145     int posicio = getPosition(id);
146     if (posicio != -1) {
147         ObjEina eina = llista.get(posicio);
148         ArrayList<Integer> llenguatges = eina.getLlenguatges();
149         llenguatges.add(idLlenguatge);
150         eina.setLlenguatges(llenguatges);
151         update(id, eina);
152     }
153 }
154
155 public void setLlenguatgesDelete(int id, int idLlenguatge) {
156     ArrayList<ObjEina> llista = getAll();
157     int posicio = getPosition(id);
158     if (posicio != -1) {
159         ObjEina eina = llista.get(posicio);
160         ArrayList<Integer> llenguatges = eina.getLlenguatges();
161         llenguatges.remove(idLlenguatge);
162         eina.setLlenguatges(llenguatges);
163         update(id, eina);
164     }
165 }
```

Per poder crear l'objecte "DaoSoftware.java", primer he hagut de crear el mètode *writeList*, el qual agafa un ArrayList d'objectes tipus "ObjSoftware" i desa els seus elements dins d'un arxiu JSON.

```

13 private void writeList(ArrayList<ObjSoftware> llista) {
14     try {
15         JSONArray jsonArray = new JSONArray();
16         for (ObjSoftware software : llista) {
17             JSONObject jsonObject = new JSONObject();
18             jsonObject.put("id", software.getId());
19             jsonObject.put("nom", software.getNom());
20             jsonObject.put("any", software.getAny());
21             jsonObject.put("llenguatges", software.getLlenguatges());
22             jsonArray.put(jsonObject);
23         }
24
25         PrintWriter out = new PrintWriter(Main.softwarePath);
26         out.write(jsonArray.toString(4)); // 4 es l'espaiat
27         out.flush();
28         out.close();
29     } catch (FileNotFoundException e) {
30         e.printStackTrace();
31     }
32 }

```

Tot seguit, he hagut de crear el mètode *getPosition*, el qual cerca la posició d'un element d'un ArrayList d'objectes del tipus "ObjSoftware" en funció del seu id i, en trobar-lo, retorna la posició d'aquest element en l'ArrayList.

```

34 private int getPosition(int id) {
35     int result = -1;
36     ArrayList<ObjSoftware> llista = getAll();
37     for (int i = 0; i < llista.size(); i++) {
38         ObjSoftware software = llista.get(i);
39         if (software.getId() == id) {
40             result = i;
41             break;
42         }
43     }
44     return result;
45 }

```

A continuació, he hagut de crear el mètode *add*, el qual agrega un objecte del tipus "ObjSoftware" a un ArrayList d'objectes d'aquest mateix tipus al mateix temps que l'actualitza.

```

47 @Override
48 public void add(ObjSoftware software) {
49     ArrayList<ObjSoftware> llista = getAll();
50     ObjSoftware item = get(software.getId());
51     if (item == null) {
52         llista.add(software);
53         writeList(llista);
54     }
55 }

```

Després, he hagut de crear el mètode *get*, el qual obté un objecte del tipus "ObjSoftware" d'un ArrayList d'objectes d'aquest mateix tipus en funció del seu id i, en fer-ho, el retorna.

```

57 @Override
58 public ObjSoftware get(int id) {
59     ObjSoftware result = null;
60     ArrayList<ObjSoftware> llista = getAll();
61     int posicio = getPosition(id);
62     if (posicio != -1) {
63         result = llista.get(posicio);
64     }
65     return result;
66 }

```

Posteriorment, he hagut de crear el mètode *getAll*, el qual recupera un ArrayList d'objectes del tipus “ObjSoftware” des d'un arxiu i, en fer-ho, la retorna.

```
68  @Override
69  public ArrayList<ObjSoftware> getAll() {
70      ArrayList<ObjSoftware> result = new ArrayList<>();
71      try {
72          String content = new String(Files.readAllBytes(Paths.get(Main.softwarePath)));
73          JSONArray jsonArray = new JSONArray(content);
74          for (int i = 0; i < jsonArray.length(); i++) {
75              JSONObject jsonObject = jsonArray.getJSONObject(i);
76              int id = jsonObject.getInt("id");
77              String nom = jsonObject.getString("nom");
78              int any = jsonObject.getInt("any");
79              JSONArray jsonLlenguatges = jsonObject.getJSONArray("llenguatges");
80              ArrayList<Integer> llenguatges = new ArrayList<>();
81              for (int j = 0; j < jsonLlenguatges.length(); j++) {
82                  llenguatges.add(jsonLlenguatges.getInt(j));
83              }
84              ObjSoftware software = new ObjSoftware(id, nom, any, llenguatges);
85              result.add(software);
86          }
87      } catch (IOException e) {
88          e.printStackTrace();
89      }
90      return result;
91  }
```

Seguidament, he hagut de crear el mètode *update*, el qual actualitza un objecte del tipus “ObjSoftware” en un ArrayList d'objectes d'aquest mateix tipus i, en fer-ho, la desa.

```
93  @Override
94  public void update(int id, ObjSoftware software) {
95      ArrayList<ObjSoftware> llista = getAll();
96      int posicio = getPosition(id);
97      if (posicio != -1) {
98          llista.set(posicio, software);
99          writeList(llista);
100      }
101  }
```

Més tard, he hagut de crear el mètode *delete*, el qual elimina un objecte del tipus “ObjSoftware” d'un ArrayList d'objectes d'aquest mateix tipus en funció del seu id i, en fer-ho, la desa.

```
103  @Override
104  public void delete(int id) {
105      ArrayList<ObjSoftware> llista = getAll();
106      int posicio = getPosition(id);
107      if (posicio != -1) {
108          llista.remove(posicio);
109          writeList(llista);
110      }
111  }
```

Seguit d'això, he hagut de crear el mètode *print*, el qual imprimeix per pantalla la informació continguda en un ArrayList d'objectes del tipus “ObjSoftware”.

```
113  @Override
114  public void print() {
115      ArrayList<ObjSoftware> llista = getAll();
116      for (int i = 0; i < llista.size(); i++) {
117          System.out.println("    " + llista.get(i));
118      }
119  }
```

Finalment, he hagut de crear els setters de tots els atributs de “ObjSoftware”.

```

121  @Override
122  public void setNom(int id, String nom) {
123      ArrayList<ObjSoftware> llista = getAll();
124      int posicio = getPosition(id);
125      if (posicio != -1) {
126          ObjSoftware software = llista.get(posicio);
127          software.setNom(nom);
128          update(id, software);
129      }
130  }
131
132  @Override
133  public void setAny(int id, int any) {
134      ArrayList<ObjSoftware> llista = getAll();
135      int posicio = getPosition(id);
136      if (posicio != -1) {
137          ObjSoftware software = llista.get(posicio);
138          software.setAny(any);
139          update(id, software);
140      }
141  }
142
143  public void setllenguatgesAdd(int id, int idllenguatge) {
144      ArrayList<ObjSoftware> llista = getAll();
145      int posicio = getPosition(id);
146      if (posicio != -1) {
147          ObjSoftware software = llista.get(posicio);
148          ArrayList<Integer> llenguatges = software.getllenguatges();
149          llenguatges.add(idllenguatge);
150          software.setllenguatges(llenguatges);
151          update(id, software);
152      }
153  }
154
155  public void setllenguatgesDelete(int id, int idllenguatge) {
156      ArrayList<ObjSoftware> llista = getAll();
157      int posicio = getPosition(id);
158      if (posicio != -1) {
159          ObjSoftware software = llista.get(posicio);
160          ArrayList<Integer> llenguatges = software.getllenguatges();
161          llenguatges.remove(idllenguatge);
162          software.setllenguatges(llenguatges);
163          update(id, software);
164      }
165  }

```

- Fes funcions Update específiques per cada atribut, al seu Dao corresponent, és a dir:

Dao tindrà add, get, getAll, update, delete, print, setNom, setAny

Dao tindrà setNom(int id, String nom) i setAny(int id, int any)

```

src > main > java > com > project > Dao.java
1  package com.project;
2
3  import java.util.ArrayList;
4
5  public interface Dao<T> {
6
7      void add(T t); // Equival a Create
8
9      T get(int id); // Equival a Read
10
11     ArrayList<T> getAll();
12
13     void update(int id, T t);
14
15     void delete(int id);
16
17     void print();
18
19     void setNom(int id, String nom);
20
21     void setAny(int id, int any);
22 }

```



DaoLlenguatge a més tindrà  
 setDificultat(int id, String dificultat)  
 setPopularitat(int id, int popularitat)

```

118 public void setDificultat(int id, String dificultat) {
119     ArrayList<ObjLlenguatge> llista = getAll();
120     int posicio = getPosition(id);
121     if (posicio != 1) {
122         ObjLlenguatge llenguatge = llista.get(posicio);
123         llenguatge.setPopularitat(popularitat);
124         update(id, llenguatge);
125     }
126 }
127
128 public void setPopularitat(int id, int popularitat) {
129     ArrayList<ObjLlenguatge> llista = getAll();
130     int posicio = getPosition(id);
131     if (posicio != 1) {
132         ObjLlenguatge llenguatge = llista.get(posicio);
133         llenguatge.setPopularitat(popularitat);
134         update(id, llenguatge);
135     }
136 }

```

DaoEina i DaoSoftware a més tindran:  
 setLlenguatgesAdd(int id, int idLlenguatge) Per afegir un id a la llista  
 setLlenguatgesDelete(id, int idLlenguatge) per treure un id de la llista

```

121 public void setLlenguatgesAdd(int id, int idLlenguatge) {
122     ArrayList<ObjEina> llista = getAll();
123     int posicio = getPosition(id);
124     if (posicio != -1) {
125         ObjEina eina = llista.get(posicio);
126         ArrayList<Integer> llenguatges = eina.getLlenguatges();
127         llenguatges.add(idLlenguatge);
128         eina.setLlenguatges(llenguatges);
129         update(id, eina);
130     }
131 }
132
133 public void setLlenguatgesDelete(int id, int idLlenguatge) {
134     ArrayList<ObjEina> llista = getAll();
135     int posicio = getPosition(id);
136     if (posicio != -1) {
137         ObjEina eina = llista.get(posicio);
138         ArrayList<Integer> llenguatges = eina.getLlenguatges();
139         llenguatges.remove(idLlenguatge);
140         eina.setLlenguatges(llenguatges);
141         update(id, eina);
142     }
143 }

```

```

121 public void setLlenguatgesAdd(int id, int idLlenguatge) {
122     ArrayList<ObjSoftware> llista = getAll();
123     int posicio = getPosition(id);
124     if (posicio != -1) {
125         ObjEina eina = llista.get(posicio);
126         ArrayList<Integer> llenguatges = eina.getLlenguatges();
127         llenguatges.add(idLlenguatge);
128         eina.setLlenguatges(llenguatges);
129         update(id, eina);
130     }
131 }
132
133 public void setLlenguatgesDelete(int id, int idLlenguatge) {
134     ArrayList<ObjSoftware> llista = getAll();
135     int posicio = getPosition(id);
136     if (posicio != -1) {
137         ObjEina eina = llista.get(posicio);
138         ArrayList<Integer> llenguatges = eina.getLlenguatges();
139         llenguatges.remove(idLlenguatge);
140         eina.setLlenguatges(llenguatges);
141         update(id, eina);
142     }
143 }

```

Fes que el programa Main.java tingui aquest codi i mostri aquest resultat:

```
import java.util.ArrayList;

public class Main {
    public static String basePath = System.getProperty("user.dir") + "/";
    public static String llenguatgesPath = basePath + "./dbProgramacio/llenguatges.json";
    public static String einesPath = basePath + "./dbProgramacio/eines.json";
    public static String softwarePath = basePath + "./dbProgramacio/software.json";

    public static void main(String[] args) {

        DaoEina daoEina = new DaoEina();
        DaoLlenguatge daoLlenguatge = new DaoLlenguatge();
        DaoSoftware daoSoftware = new DaoSoftware();

        // Afegir una eina
        ArrayList<Integer> alLenguatges0 = new ArrayList<>();
        alLenguatges0.add(0);
        alLenguatges0.add(1);
        ObjEina objEina0 = new ObjEina (5, "Text", 2000, alLenguatges0);
        daoEina.add(objEina0);

        // Modificar una eina
        ArrayList<Integer> alLenguatges1 = new ArrayList<>();
        alLenguatges1.add(0);
        alLenguatges1.add(1);
        alLenguatges1.add(2);
        ObjEina objEina1 = new ObjEina (5, "TextEdit", 2001, alLenguatges1);
        daoEina.update(5, objEina1);

        // Afegir llenguatge a eina
        daoEina.setLlenguatgesAdd(5, 3);

        // Treure llenguatge a eina
        daoEina.setLlenguatgesDelete(5, 2);

        // Llistar les eines
        daoEina.print();

        // Esborrar eina amb id 5
        daoEina.delete(5);

        // Afegir un llenguatge
        ObjLlenguatge objLlenguatge0 = new ObjLlenguatge(5, "Dart", 2011, "facil", 8);
        daoLlenguatge.add(objLlenguatge0);

        // Canviar el nom
        daoLlenguatge.setNom(5, "Dart+Flutter");

        // Canviar l'any
        daoLlenguatge.setAny(5, 2018);

        // Canviar dificultat
        daoLlenguatge.setDificultat(5, "mitja");

        // Canviar popularitat
        daoLlenguatge.setPopularitat(5, 9);

        // Llistar els llenguatges
        daoLlenguatge.print();

        // Esborrar llenguatge amb id 5
        daoLlenguatge.delete(5);
```

```

// Afegir un software
ArrayList<Integer> allenguatges2 = new ArrayList<>();
allenguatges2.add(3);
ObjSoftware objSoftware = new ObjSoftware(10, "webTool", 2022, allenguatges2);
daoSoftware.add(objSoftware);

// Llistar software
daoSoftware.print();

// Esborrar llenguatge amb id 5
daoSoftware.delete(10);
}
}

```

```

Eina: 0 Visual Studio Code, 2015 - [0, 1, 2, 3, 4]
Eina: 1 Xcode, 2003 - [0, 1, 2]
Eina: 2 Eclipse, 2001 - [4]
Eina: 3 Vim, 1991 - [0, 1, 2, 3, 4]
Eina: 4 Nano, 1999 - [0, 1, 2, 3, 4]
Eina: 5 TextEdit, 2001 - [0, 1, 3]
Llenguatge: 0 C, 1972 - dificil/8
Llenguatge: 1 C++, 1983 - mitja/7
Llenguatge: 2 Objective C, 1984 - dificil/2
Llenguatge: 3 JavaScript, 1996 - facil/6
Llenguatge: 4 Java, 1995 - mitja/5
Llenguatge: 5 Dart+Flutter, 2018 - mitja/9
Software: 0 Visual Studio Code, 2015 - [0, 1, 2, 3, 4]
Software: 1 Xcode, 2003 - [0, 1, 2]
Software: 2 Eclipse, 2001 - [4]
Software: 3 Vim, 1991 - [0, 1, 2, 3, 4]
Software: 4 Nano, 1999 - [0, 1, 2, 3, 4]
Software: 10 webTool, 2022 - [3]

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Eina: 0 Visual Studio Code, 2015 - [0, 1, 2, 3, 4]
Eina: 1 Xcode, 2003 - [0, 1, 2]
Eina: 2 Eclipse, 2001 - [4]
Eina: 3 Vim, 1991 - [0, 1, 2, 3, 4]
Eina: 4 Nano, 1999 - [0, 1, 2, 3, 4]
Eina: 5 TextEdit, 2001 - [0, 1, 3]
Llenguatge: 0 C, 1972 - dificil/8
Llenguatge: 1 C++, 1983 - mitja/7
Llenguatge: 2 Objective C, 1984 - dificil/2
Llenguatge: 3 JavaScript, 1996 - facil/6
Llenguatge: 4 Java, 1995 - mitja/5
Llenguatge: 5 Dart+Flutter, 2018 - mitja/9
Software: 0 Visual Studio Code, 2015 - [0, 1, 2, 3, 4]
Software: 1 Xcode, 2003 - [0, 1, 2]
Software: 2 Eclipse, 2001 - [4]
Software: 3 Vim, 1991 - [0, 1, 2, 3, 4]
Software: 4 Nano, 1999 - [0, 1, 2, 3, 4]
Software: 10 webTool, 2022 - [3]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.036 s
[INFO] Finished at: 2023-11-20T22:24:22+01:00
[INFO] -----

```