

```
elif _operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
elif _operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True  
  
#selection at the end -add back the deselected mirror modifier object  
mirror_ob.select= 1  
modifier_ob.select=1  
bpy.context.scene.objects.active = modifier_ob  
print("Selected" + str(modifier_ob)) # modifier ob is the active ob  
#mirror_ob.select = 0  
#one = bpy.context.selected_objects[0]
```

FILTRADO COLABORATIVO

PRÁCTICA 2
Gestión de Información en la Web



Álvaro de la Flor Bonilla

```
username;  
password;  
$database;  
$charset;  
  
public function connect()  
{  
    $link = null;  
    if (!$link = mysql_connect($database, $username, $password))  
    {  
        throw new MySQLException("Error al conectar a la base de datos");  
    }  
    mysql_query("SET CHARACTER SET UTF8");  
    mysql_query("SET NAMES UTF8");  
    mysql_query("USE $database");  
    return $link;  
}
```



CONFIGURACIÓN

LENGUAJE



BASE DE DATOS





DATASET

MOVIELENS 100K DATASET

100K Dataset

ings. Stable benchmark datas

MB, [checksum](#))

[S](#)

<https://grouplens.org/datasets/movielens/100k/>



Desarrollo

MONGODB, RECOMENDADOR



Lectura y guardado

```
6 def read_u_user():
7     users = []
8     with open(get_url_u_user()) as f:
9         lines = f.readlines()
10        for line in lines:
11            x = line.split('|')
12            user_id = x[0].strip()
13            age = x[1].strip()
14            gender = x[2].strip()
15            occupation = x[3].strip()
16            zip_code = x[4].strip()
17            user = User(user_id=user_id, age=age, gender=gender, occupation=occupation, zip_code=zip_code)
18            users.append(user)
19
20    return users
21
22
23 def save_users(rates):
24     db = get_mongo_db()
25     users_mongodb = db.users
26     users_mongodb.drop()
27     user_dict = []
28     for rate in rates:
29         user_dict.append(json.loads(rate.to_json()))
30     users_mongodb.insert_many(user_dict)
```

giw_db.movies

COLLECTION SIZE: 412.85KB TOTAL DOCUMENTS: 1682 INDEXES TOTAL SIZE: 32KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

FILTER {"filter": "example"}

QUERY RESULTS 1-20 OF MANY

```
_id: ObjectId("607086cbeb38f9a28fe2354e")
✓ genres: Array
  0: "Animation"
  1: "Children s"
  2: "Comedy"
imdb_url: "http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)"
movie_id: "1"
release_date: "1995-01-01 00:00:00"
title: "Toy Story (1995)"
video_release_date: ""
```

```
_id: ObjectId("607086cbeb38f9a28fe2354f")
✓ genres: Array
  0: "Action"
  1: "Adventure"
  2: "Thriller"
imdb_url: "http://us.imdb.com/M/title-exact?GoldenEye%20(1995)"
movie_id: "2"
```

Lectura y guardado

giw_db.recommendation

COLLECTION SIZE: 3.65KB TOTAL DOCUMENTS: 1 INDEXES

Find

Indexes

Schema Anti-Patterns 0

FILTER {"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("607337054f5d94798a2111a2")
11/04/2021 19:51:01: Array
  0: Array
    "{
      "item_id": "22",
      "rating": "2",
      "timestamp": 1618163439...."
    }
    "{
      "item_id": "37",
      "rating": "1",
      "timestamp": 1618163440...."
    }
    "{
      "item_id": "67",
      "rating": "2",
      "timestamp": 1618163441...."
    }
    "{
      "item_id": "235",
      "rating": "2",
      "timestamp": 1618163442...."
```

giw_db.recommendation

COLLECTION SIZE: 3.65KB TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 24KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

FILTER {"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
11/04/2021 19:51:01: Array
  0: Array
    0: Object
      title: "Flirt (1995)"
      id: "1495"
      rate: 4.95014794292052
    1: Object
      title: "Braindead (1992)"
      id: "853"
      rate: 4.928501038011899
    2: Object
      title: "Dead Man (1995)"
      id: "922"
      rate: 4.841355190497069
    3: Object
      title: "Wallace & Gromit: The Best of Aardman Animation (1996)"
      id: "114"
      rate: 4.82066384254473
```

Recomendación

```
1  from math import sqrt
2      from utilities.rate_util import *
3      from utilities.constants import *
4  import shelve
5
6
7  def load_dict(rates_add):...
40
41
42  def get_pearson(dict_user, user_1, user_2):...
84
85
86  def top_matches(dict_user, person, similarity=get_pearson):...
98
99
100 def get_recommendations(dict_user, person, similarity=get_pearson):...
137
138
139 def transform_dict_user(dict_user):...
154
```




Desarrollo

DEMO

1. w
2. w
3. w
w
w
w
o
o
o

¡GRACIAS!

¿ALGUNA DUDA?

IV
DAS

Video module

H1-Headline

Menu

|||

|||