

Università degli Studi di Salerno
Corso di Basi di Dati 2

MongolInsects

Nicolapio Gagliarde



Cos'è MongolInsects



CERCA

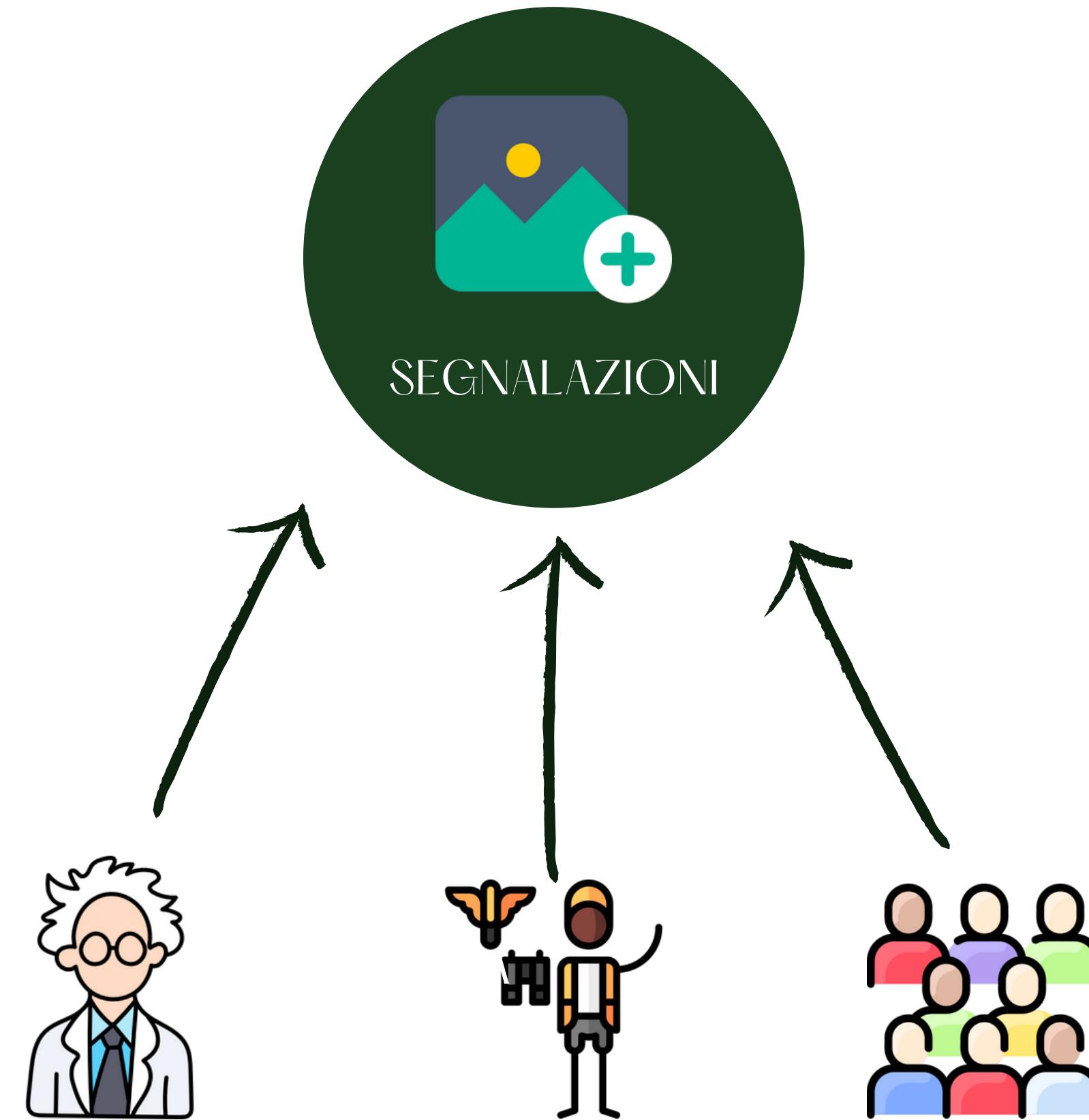


FOTOGRAFA

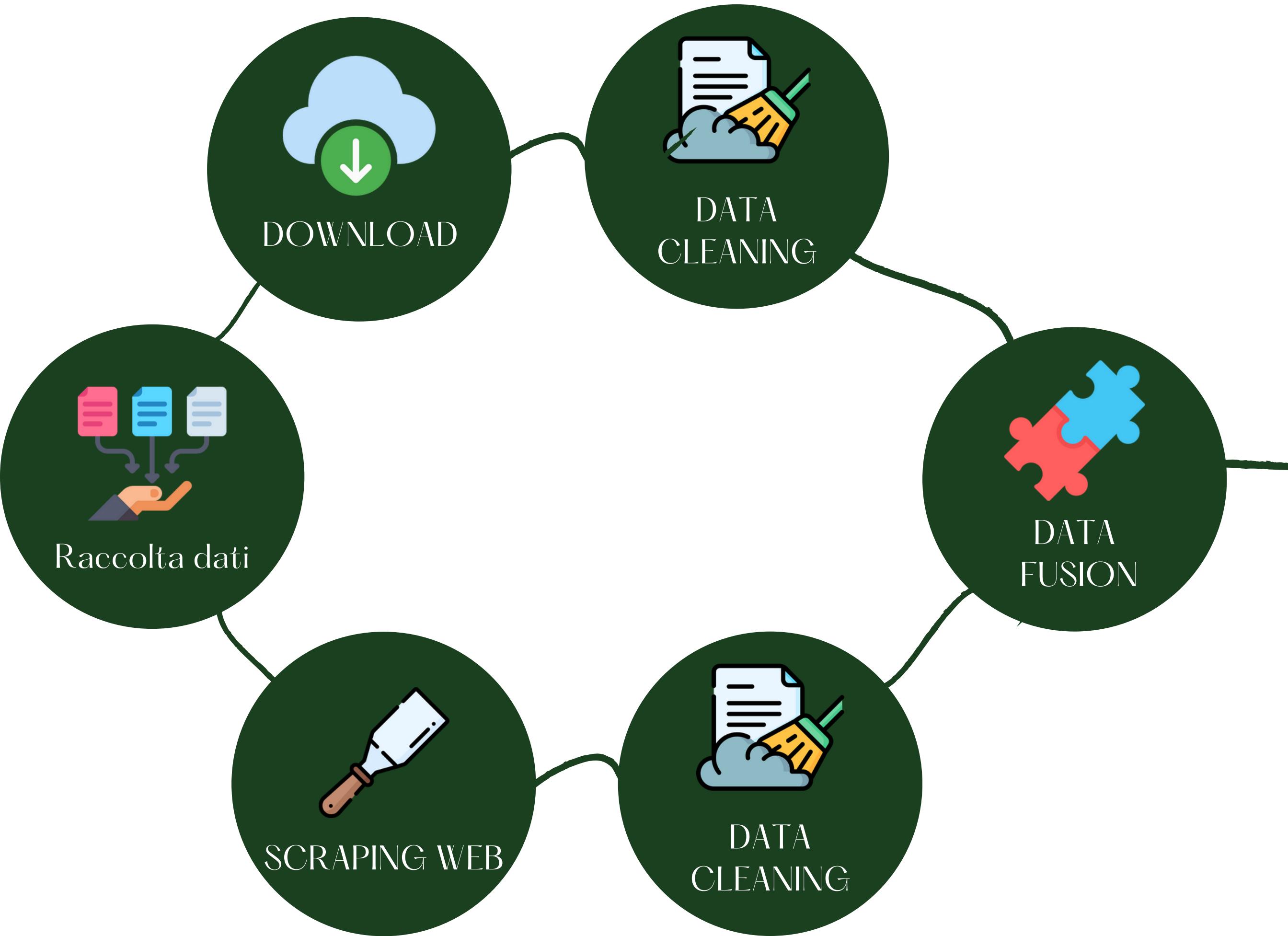


SEGNALA

Cos'è MongolInsects



Realizzazione



MongoDB®
+
Flask

Raccolta dati

iNaturalist

Download DB

11 806 segnalazioni
39 proprietà
Dati omogenei e strutturati



InsectIdentification
for the casual observer

Scraping web

1593 segnalazioni
Proprietà diverse per ogni segnalazione
Dati eterogenei e non strutturati

Data Cleaning

- Labeling delle caratteristiche fisiche e delle capacità dell'animale segnalato
- Estrazione dei dati relativi agli utenti
- Rimozione delle proprietà inutili
- Rimozione dei record errati
- Parsing dei campi errati
- Conversione delle date in formato GG/MM/AAAA



Data Cleaning

- Labeling delle caratteristiche fisiche e delle capacità dell'animale segnalato
- Estrazione dei dati relativi agli utenti
- Rimozione delle proprietà inutili
- Rimozione dei record errati
- Parsing dei campi errati
- Conversione delle date in formato GG/MM/AAAA

Dizionario con 599 capacità e caratteristiche generato ad hoc

Codice Python con regex

Dataset finale e scelta del DBMS



Dati eterogenei e
senza schemi fissi.
Collezioni definite
dal dataset

Database MongoDB

utenti

```
{  
  _id: <ObjectId>,  
  username: <string>,  
  password: <string>,  
  
  isAdmin: <bool>  
CAMPO OPZIONALE  
}
```

Insecta

```
{  
  _id: <ObjectId>,  
  id_utente: <ObjectId>,  
  data_osservazione: <Date>,  
  image_url: <pattern_url>,  
  latitudine: <double>,  
  longitudine: <double>,  
  
  ...  
  CAMPI DEFINITI DALL'UTENTE E  
  DALLO SPECIFICO DATASET  
}
```

Arachnida

```
{  
  _id: <ObjectId>,  
  id_utente: <ObjectId>,  
  data_osservazione: <Date>,  
  image_url: <pattern_url>,  
  latitudine: <double>,  
  longitudine: <double>,  
  
  ...  
  CAMPI DEFINITI DALL'UTENTE E  
  DALLO SPECIFICO DATASET  
}
```

In totale: 2066 utenti, 6612 insetti, 5177 aracnidi

Popolazione del database

```
inserted_id = db.get_collection("utenti").insert_one(  
    {"username": obj["user_name"], "password": "utente"}).inserted_id  
  
osservazione = {"id_utente": inserted_id["_id"],  
                "data_osservazione": datetime.strptime(obj['observed_on'], "%Y-%m-%d"),  
                "image_url": obj['image_url'], "latitudine": float(obj["latitude"]),  
                "longitudine": float(obj["longitude"])),  
                "scientific_name": obj["scientific_name"]}  
  
db.get_collection(obj["iconic_taxon_name"]).insert_one(osservazione)  
  
scientific_name = re.sub(r'Ã|\xa0', ' ', obj["scientific_name"]).strip().replace(" ", " ")  
db.get_collection(collection).update_many({"scientific_name": scientific_name},  
                                         {"$set": {"nome_comune": obj["common-name"],  
                                                 "dimensioni": obj["Size (Adult; Length)"]}})  
db.get_collection(collection).update_many({"scientific_name": scientific_name},  
                                         {"$set": creaColori(obj["Colors"])})  
  
if db.get_collection(collection).update_many({"scientific_name": scientific_name}, {  
    "$set": attributi(obj["Descriptors"])}).modified_count > 0:  
    print("aggiornato" + scientific_name)
```

Inserimento di una segnalazione

```
ossevazione = {
    'image_url': image_url,
    'scientific_name': nome_scientifico,
    'nome_comune': nome_comune,
    'data_osservazione': datetime.strptime(request.form.get("dataosser"), "%Y-%m-%d"),
    'latitudine': float(request.form.get("latitudine")),
    'longitudine':float(request.form.get("longitudine")),
}

# Iterare sui parametri aggiuntivi e aggiungerli come proprietà dell'oggetto JSON
for chiave, valore in request.form.items():
    if chiave not in ['image', 'category', 'scientificName', 'commonName',
                      'usernameNascosta', 'dataosser', 'latitudine', 'longitudine']:
        valori = valore.split(":")
        ossevazione[valori[0]] = valori[1]

id_utente= db["utenti"].find_one({"username":request.form["usernameNascosta"]})["_id"]
ossevazione["id_utente"]=ObjectId(id_utente)
print(categoria)
print(request.form["usernameNascosta"])
db.get_collection(categoria).insert_one(ossevazione)
```

Aggiungi una nuova segnalazione

Immagine:

Nessun file selezionato

Categoria:

Nome scientifico:

Nome comune:

Data Osservazione:

Latitudine:

Longitudine:

Cancellazione e modifica di una segnalazione

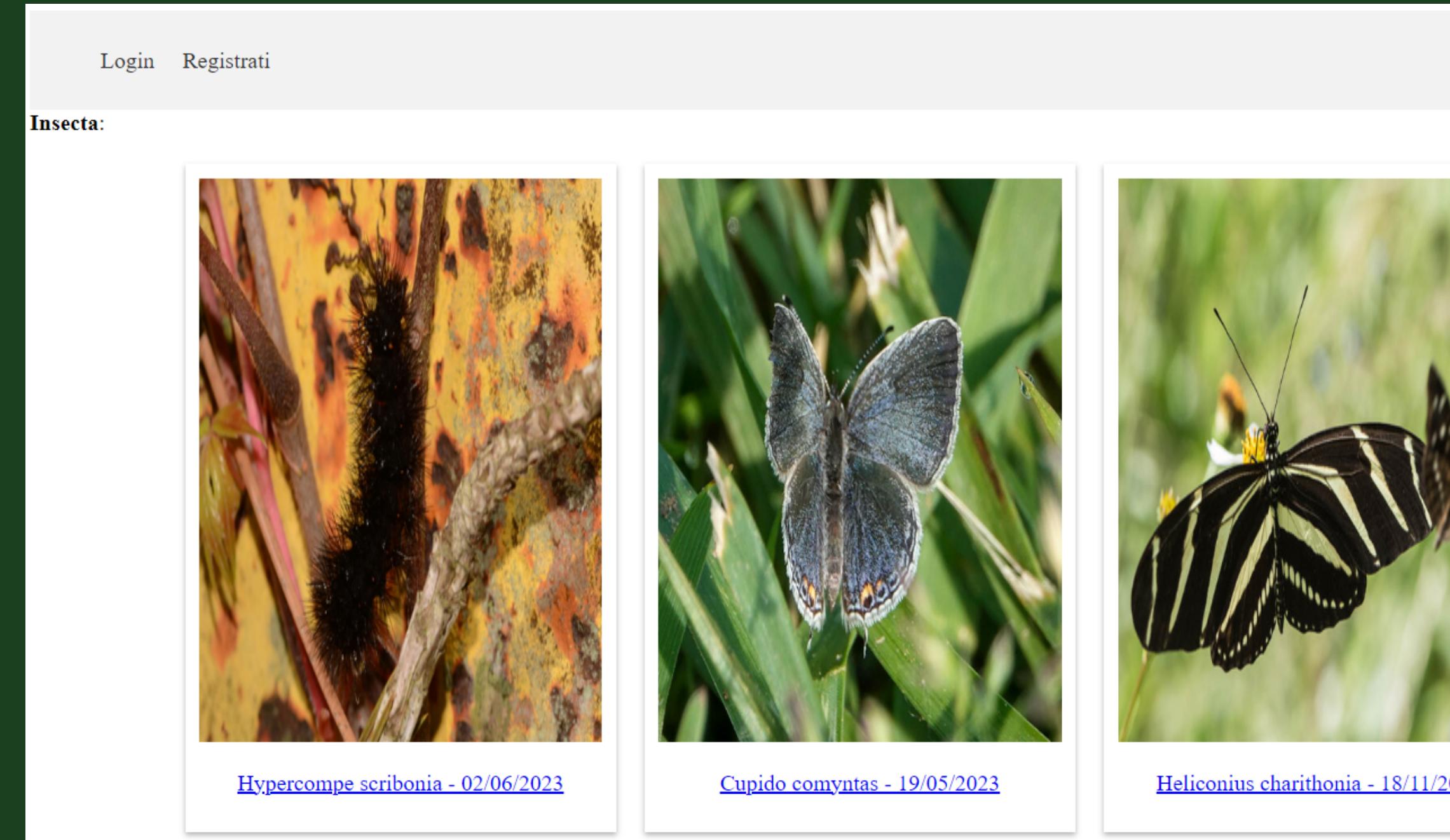
```
for collection in getCollections(db):
    result = db[collection].delete_one({'_id': ObjectId(id_documento)})
    if result.deleted_count == 1:
        return render_template('index.html', documents=getRandomDocuments(), benvenuto="admin", cancellato=True)
```

```
document_id = request.form['_id']
document_fields = request.form.to_dict()
print(document_fields)
document_fields.pop('_id') # Rimuovi l'ID dal dizionario dei campi
document_fields.pop('id_utente')
document_fields['data_osservazione'] = datetime.strptime(document_fields['data_osservazione'], "%Y-%m-%d %H:%M:%S")
document_fields['latitudine'] = float(document_fields['latitudine'])
document_fields['longitudine'] = float(document_fields['longitudine'])

for collection in getCollections(db):
    db[collection].update_one({'_id': ObjectId(document_id)}, {'$set': document_fields})
return render_template('index.html', documents=getRandomDocuments(), benvenuto="admin", modificato=True)
```

Query per la Home

```
.aggregate([
  {
    '$match': {
      '$expr': {
        '$gt': [{$size:
          {$objectToArray: '$$ROOT'}}, 10]
      }
    }
  },
  {
    '$sample': {
      'size': 8
    }
  }
])
])
```



Query per visualizzare una segnalazione

```
db.get_collection(collezione).aggregate([
```

```
    {'$match': {  
        '_id': ObjectId(documento_id)  
    }},  
    {'$lookup': {  
        'from': 'utenti',  
        'localField': 'id_utente',  
        'foreignField': '_id',  
        'as': 'utente'  
    }},  
    {  
        '$project': {  
            '_id': 0,  
            'utente.password': 0,  
            'utente._id': 0,  
            'id_utente': 0  
        }  
    }  
]
```



data_osservazione: 23/03/2023
latitudine: 33.0638020669
longitudine: -96.9752818584
scientific_name: Cicindela sexguttata
dimensioni: 10mm to 15mm
nome_comune: Six-spotted Tiger Beetle
Colore principale: green
Colore traslucido: black
colore/forma: green
forma: shiny
pattern: spots
utente:Kerry Carloy

Query per cercare delle segnalazioni

```
{  
  'data_osservazione': {  
    '$gte': datetime.datetime(2022, 12, 9, 0, 0),  
    '$lte': datetime.datetime(2022, 12, 11, 0, 0)  
  },  
  'latitudine': {  
    {'$gte': 23.0617730624, '$lte': 43.0617730624},  
  'longitudine': {  
    {'$gte': -106.9628690815, '$lte': -86.9628690815},  
  'scientific_name': {  
    {'$regex': re.compile('Periplaneta\\ americana',  
      re.IGNORECASE)}  
  }  
}
```

Cerca una segnalazione

Inizio Periodo Data Osservazione:

09/12/2022

Fine Periodo Data Osservazione:

11/12/2022

Latitudine:

33.0617730624

Longitudine:

-96.9628690815

Nome Scientifico:

Periplaneta americana

Username utente:

Pluto

Cerca

Chiudi

Query per cercare delle segnalazioni di uno specifico utente

```
{"$lookup": {  
    "from": "utenti",  
    "localField": "id_utente",  
    "foreignField": "_id",  
    "as": "utente"  
},  
{$match": {  
    "utente.username": request.form["usernameUtente"]  
},  
{$project": {  
    "_id": 0,  
    "id_utente":0,  
    "utente.password":0,  
    "utente._id":0  
}  
}
```

Username utente:

Segnalazioni trovate

Arachnida



data_osservazione: 2023-05-01 00:00:00
latitudine: 33.075603
longitudine: -96.99855
scientific_name: Bagheera prosper

Università degli Studi di Salerno
Corso di Basi di Dati 2

Grazie

Nicolapio Gagliarde