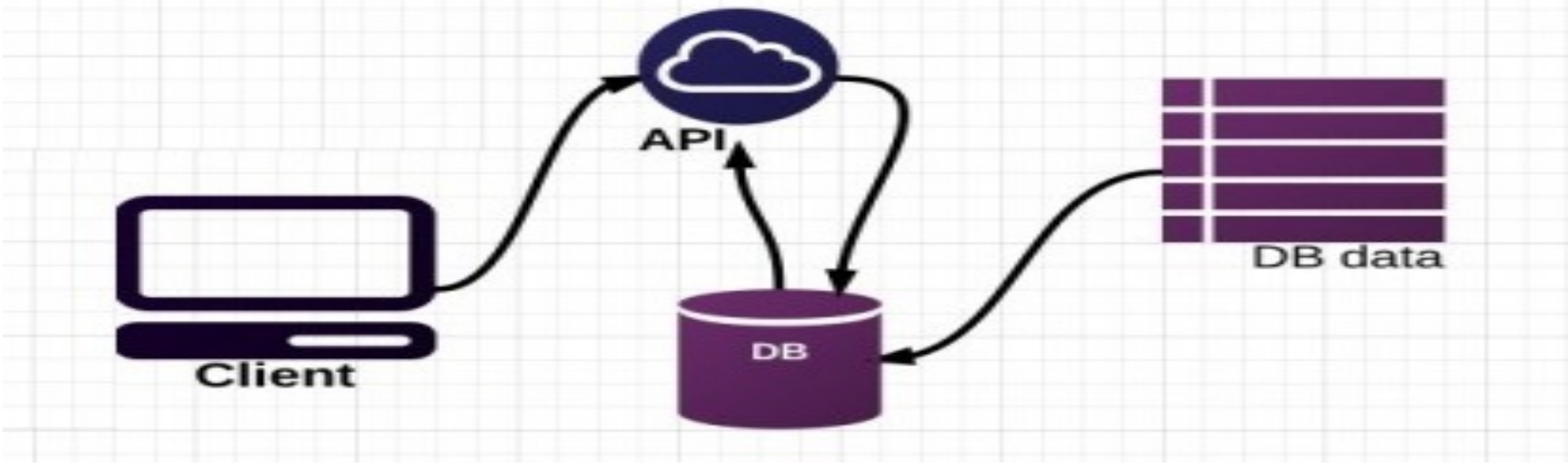


# Web Services



## Gui3n de Pr3cticas 5: API REST

S. Alonso ([zerjioi@ugr.es](mailto:zerjioi@ugr.es)) y J.M. Guirao ([jmguirao@ugr.es](mailto:jmguirao@ugr.es))

Entrega: 06 - diciembre

### Resumen

En esta pr3ctica haremos la misma funcionalidad de la pr3ctica anterior (CRUD sobre la BD), pero usando una API RESTfull. Lo haremos siguiendo las indicaciones de [What is a REST API?](#), y las directrices de [RESTful](#):

- Usar los m3todos de HTTP expl3citamente
- Sin estado, cacheable
- Exponer urls estilo path, interface uniforme
- Responder JSON o XML

### End points

Lo primero ser3 planificar que urls, con que verbos, y que respuesta tendr3n. Si tengo la colecci3n de movies:

- **GET /api/movies**, devolver3 una lista con todos los registros
- **GET /api/movies?year=2014**, devolver3 una lista esta b3squeda
- **POST /api/movies**, crear3 un registro nuevo a partir de los par3metros que enviemos con el POST, y devolver3 el **id** del registro creado, y sus datos
- **PUT /api/movies/tt1155056**, modificar3 el registro con **id** tt1155056 a partir de los par3metros enviados en el POST y devolver3 el **id** del registro modificado y sus datos
- **DELETE /api/movies/tt1155056**, borrar3 el registro con **id** tt1155056 y devolver3 el **id** del registro borrado

En su caso, devolver3 los mensajes de error correspondientes. El c3digo ser3a:

```
#!/app/app.py
from flask import request, jsonify
from bson import ObjectId
...
# para devolver una lista (GET), o a1adir (POST)
@app.route('/api/movies', methods=['GET', 'POST'])
def api_1():
    if request.method == 'GET':
        lista = []
        movies = db.video_movies.find().sort('year')
        for movie in movies:
            lista.append({
                'id': str(movie.get('_id')), # pasa a string el ObjectId
                'title': movie.get('title'),
                'year': movie.get('year'),
                'imdb': movie.get('imdb')
            })
        return jsonify(lista)

    if request.method == 'POST':
        ...

# para devolver una, modificar o borrar
@app.route('/api/movies/<id>', methods=['GET', 'PUT', 'DELETE'])
def api_2(id):
    if request.method == 'GET':
        try:
            movie = db.video_movies.find_one({'_id': ObjectId(id)})
            return jsonify({
                'id': id,
                'title': movie.get('title'),
                'year': movie.get('year'),
                'imdb': movie.get('imdb')
            })
        except:
            return jsonify({'error': 'Not found'}), 404
    ...
```

Para probarlo podemos usar las utilidades [curl](#) o [httpie](#)

#### Para nota

Hacer otra versi3n del api usando la librer3a [flask-restfull](#), como en [Python REST API Tutorial - Building a Flask REST API](#)