



# IPG

Politécnico  
da Guarda

Escola Superior  
de Tecnologia e Gestão

---

## SISTEMAS DISTRIBUÍDOS

<b>Curso(s):</b>	Desenvolvimento de Aplicações Informáticas
<b>Unidade(s) Curricular(s):</b>	Sistemas Distribuídos
<b>Ano Letivo:</b>	2018/2019
<b>Docente:</b>	Pedro Pinto
<b>Estudantes:</b>	Micael Costa Pires N°1700011 Pedro Matos N°1700575 Rafael Santos N°1700039
<b>Data:</b>	13-01-2019

# Índice

Desenvolvimento da api e interface .....	5
Criação de subpastas.....	6
Elaboração da Interface .....	9
Ligação do servidor nodejs e interface à mblab e google cloud .....	12
Virtualização do projeto .....	13
Publicação da API .....	14

---

## DESENVOLVIMENTO DA API E INTERFACE

---

Para o desenvolvimento da API foram utilizados os seguintes programas:

**Node.js**

**MongoDB**

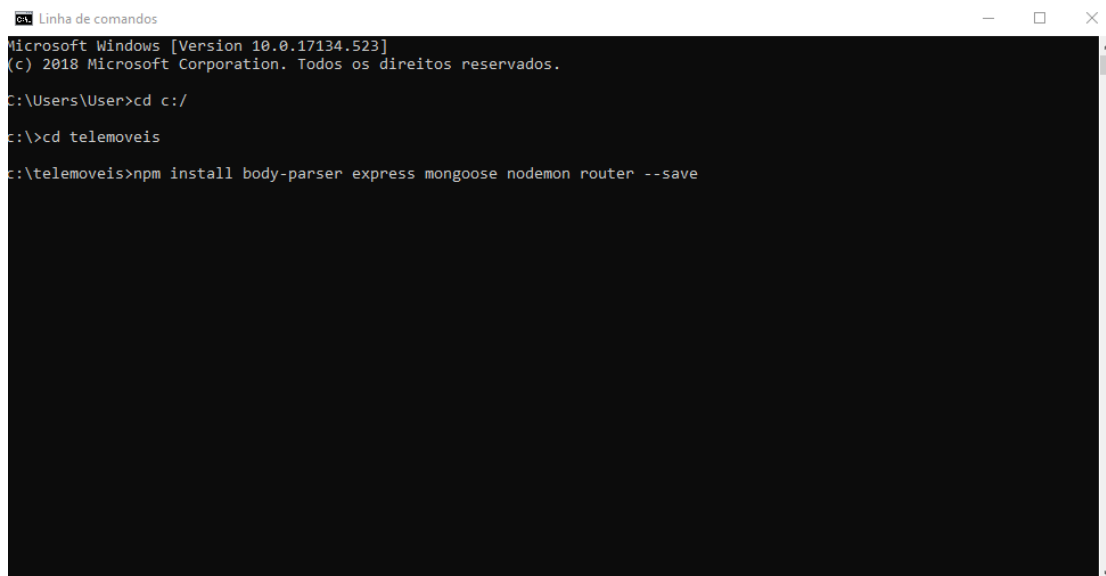
**Microsoft Visual Studio Code**

**Postman**

O primeiro passo para o desenvolvimento da API foi a instalação do node.js com o propósito de ajudar o servidor a interpretar o código do lado cliente.

De seguida iniciou-se a instalação do **MongoDB** para armazenar e organizar os dados.

Após a instalação do **Node.js** e **MongoDB**, foi criada uma pasta onde foram armazenados todos arquivos do projeto, de seguida através da Linha de Comandos executamos o comando “**npm init**” no diretório do nosso projeto onde foram criados os arquivos base, posteriormente instalámos as dependências necessárias para a elaboração da mesma que foram “**body-parser**”, “**express**”, “**mongoose**”, “**nodemon**” e a “**router**”.



```
Linha de comandos
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\Users\User>cd c:/

C:\>cd telemoveis

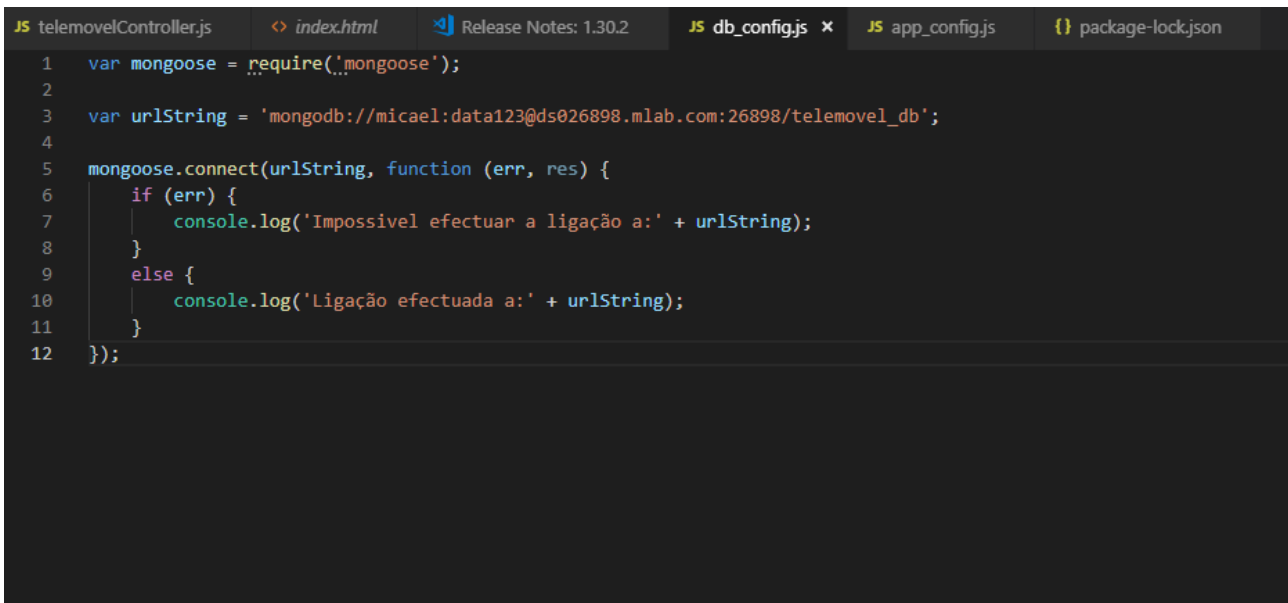
C:\telemoveis>npm install body-parser express mongoose nodemon router --save
```

## Criação de subpastas

Depois criámos as respetivas subpastas:

- Na subpasta “**config**” criou-se dois ficheiros:

“**bd\_config**” - que contem todas as configurações necessárias para a mesma a poder operar, como a porta e permissões de controlo,

A screenshot of a code editor with a dark theme. The editor has several tabs at the top: 'JS telemovelController.js', '<> index.html', 'Release Notes: 1.30.2', 'JS db\_config.js' (which is the active tab), 'JS app\_config.js', and '{} package-lock.json'. The code in the active tab is as follows:

```
1  var mongoose = require('mongoose');
2
3  var urlString = 'mongodb://michael:data123@ds026898.mlab.com:26898/telemovel_db';
4
5  mongoose.connect(urlString, function (err, res) {
6    if (err) {
7      console.log('Impossivel efectuar a ligação a:' + urlString);
8    }
9    else {
10     console.log('Ligação efectuada a:' + urlString);
11    }
12  });
```

“app\_config” onde colocamos as configurações da “app”

```
JS telemovelController.js  <> index.html  Release Notes: 1.30.2  JS db_config.js  JS app_config.js x  {} package-lock.json

1  const express = require('express');
2  var bodyParser = require('body-parser');
3  var port = '8888';
4  var app = module.exports = express();
5  const mongoose = require('mongoose');
6
7  mongoose.connect('mongodb://michael:data123@ds026898.mlab.com:26898/telemovel_db');
8  mongoose.connection.once('open', () => {
9    console.log('Ligação efectuada com sucesso a Base de Dados');
10 });
11
12 app.listen(port);
13
14 app.use(bodyParser.urlencoded({ extended: true }));
15 app.use(bodyParser.json());
16
17 app.use(function (req, res, next) {
18   res.setHeader('Access-Control-Allow-Origin', '*');
19   res.setHeader('Access-Control-Allow-Methods', 'GET,POST,PUT,DELETE');
20   res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With,content-type,Authorization');
21   next();
22 })
```

- Na subpasta “**controllers**” colocaram-se as funções que pretendíamos que a nossa API utilizasse.

```
JS telemovelController.js x  <> index.html  Release Notes: 1.30.2  JS db_config.js  JS app_config.js  {} package-lock.json

1  var telemovel = require('../models/telemovel');
2
3  exports.save = function (nome, ram, armazenamento, resolucao, camera, preco, callback) {
4    new telemovel({
5      'nome': nome,
6      'ram': ram,
7      'armazenamento': armazenamento,
8      'resolucao': resolucao,
9      'camera': camera,
10     'preco': preco,
11   }).save(function (error, telemovel) {
12     if (error) {
13       callback({ error: 'Nao foi possivel guardar' });
14     } else {
15       callback(telemovel);
16     }
17   });
18 }
19
20 exports.list = function (callback) {
21   telemovel.find({}, function (error, telemovel) {
22     if (error) {
23       callback({ error: 'Não foi possível encontrar nenhum resultado !' });
24     } else {
25       callback(telemovel);
26     }
27   });
28 }
29
30
31 exports.delete = function (id, callback) {
32   telemovel.findById(id, function (error, telemovel) {
33     if (error) {
34       callback({ error: 'Não foi possível excluir o ficheiro !' });
35     } else {
36       telemovel.remove(function (error) {
37         if (!error) {
38           callback({ resposta: "Telemovel excluido com sucesso !" });
39         }
40       });
41     }
42   });
43 }
44 }
```

- Na subpasta “**models**” definimos os campos e qual a informação recolhia e entregava

```
JS telemovelController.js JS telemovel.js x Release Notes: 1.30.2 JS db_config.js JS app_config.js {} package-lock.json
1 var mongoose = require('mongoose');
2 var Schema = mongoose.Schema;
3
4 var TelemovelSchema = new Schema({
5   nome: String,
6   ram: String,
7   armazenamento: String,
8   resolucao: String,
9   camera: String,
10  preco: String,
11 });
12
13 module.exports = mongoose.model('telemovel', TelemovelSchema);
```

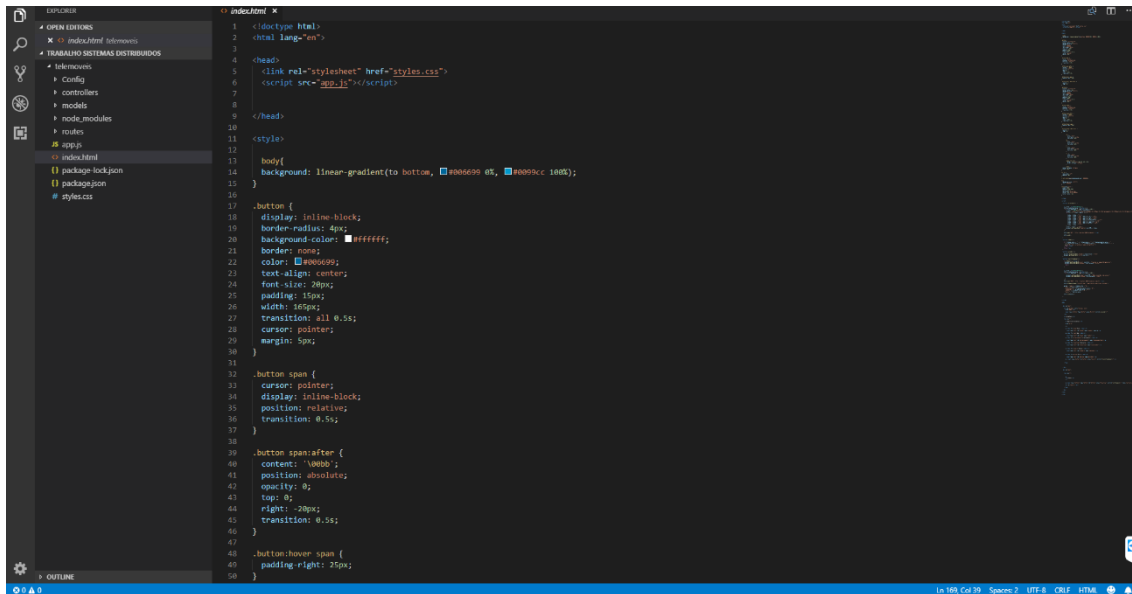
- Na subpasta “**routes**” definimos as rotas e funções como:

“GET”,  
“POST”  
“DELETE”

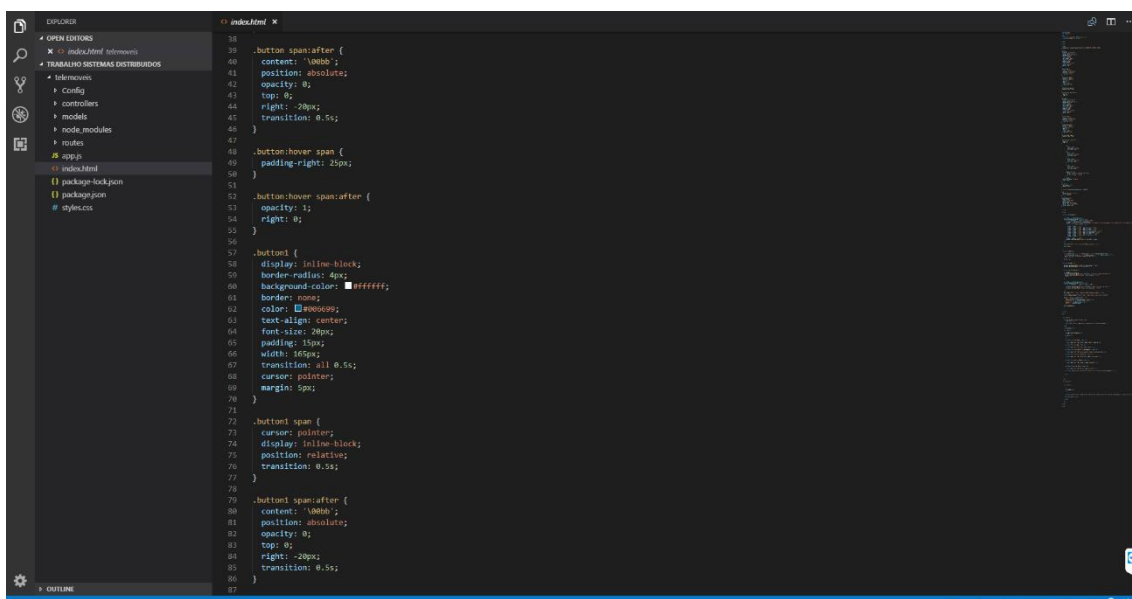
```
JS telemovelController.js JS telemovelRoutes.js x Release Notes: 1.30.2 JS db_config.js JS app_config.js {} package-lock.json
1 var express = require('express');
2 var router = express.Router();
3 var telemovelController = require('../controllers/telemovelController');
4
5
6 router.get('/', function (req, res) {
7   telemovelController.list(function (resp) {
8     res.json(resp);
9   });
10 });
11
12
13 router.post('/registo', function (req, res) {
14   var nome = req.body.nome;
15   var ram = req.body.ram;
16   var armazenamento = req.body.armazenamento;
17   var resolucao = req.body.resolucao;
18   var camera = req.body.camera;
19   var preco = req.body.preco;
20
21   telemovelController.save(nome, ram, armazenamento, resolucao, camera, preco, function (resp) {
22     res.json(resp);
23   });
24 });
25
26 router.delete('/apagar/:id', function (req, res) {
27   var id = req.params.id;
28
29   telemovelController.delete(id, function (resp) {
30     res.json(resp);
31   });
32 });
33
34 module.exports = router;
```

# Elaboração da Interface

Em seguida elaboramos uma interface em **HTML** para a apresentação da **API** no editor **Microsoft Visual Studio Code**.



```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <link rel="stylesheet" href="styles.css">
6   <script src="app.js"></script>
7 </head>
8
9 <body>
10
11 <style>
12
13   body {
14     background: linear-gradient(to bottom, #000000 45%, #000000 45% 55%, #000000 55% 100%);
15   }
16
17   .button {
18     display: inline-block;
19     border-radius: 4px;
20     background-color: #ffffff;
21     border: none;
22     color: #000000;
23     text-align: center;
24     font-size: 20px;
25     padding: 10px;
26     width: 100px;
27     transition: all 0.5s;
28     cursor: pointer;
29     margin: 5px;
30   }
31
32   .button span {
33     cursor: pointer;
34     display: inline-block;
35     position: relative;
36     transition: 0.5s;
37   }
38
39   .button span:after {
40     content: "\000020";
41     position: absolute;
42     opacity: 0;
43     top: 0;
44     right: -20px;
45     transition: 0.5s;
46   }
47
48   .button:hover span {
49     padding-right: 25px;
50 }
```



```
38
39 .button span:after {
40   content: "\000020";
41   position: absolute;
42   opacity: 0;
43   top: 0;
44   right: -20px;
45   transition: 0.5s;
46 }
47
48 .button:hover span {
49   padding-right: 25px;
50 }
51
52 .button1 span:after {
53   opacity: 1;
54   right: 0;
55 }
56
57 .button1 {
58   display: inline-block;
59   border-radius: 4px;
60   background-color: #ffffff;
61   border: none;
62   color: #000000;
63   text-align: center;
64   font-size: 20px;
65   padding: 10px;
66   width: 100px;
67   transition: all 0.5s;
68   cursor: pointer;
69   margin: 5px;
70 }
71
72 .button1 span {
73   cursor: pointer;
74   display: inline-block;
75   position: relative;
76   transition: 0.5s;
77 }
78
79 .button1 span:after {
80   content: "\000020";
81   position: absolute;
82   opacity: 0;
83   top: 0;
84   right: -20px;
85   transition: 0.5s;
86 }
87
```

The screenshot displays a web development environment with a dark theme. On the left, a file explorer shows the project structure, including folders like 'controllers', 'models', 'routes', and 'views', and files like 'package-lock.json', 'package.json', and 'styles.css'. The main editor area shows the CSS code for 'index.html'. The code defines a table with a fixed width of 350px, a collapsed border, and a header with a solid blue background. The table has two columns: 'th' and 'td'. The 'th' column is centered, and the 'td' column is left-aligned. The table is styled with a white background and a blue border. The code is as follows:

```
91
92
93 button:hover span{after {
94   opacity: 1;
95   right: 0;
96 }
97
98 h1 {
99   color: #white;
100   font-family: lucida;
101   font-size: 150%;
102 }
103
104 h2 {
105   color: #white;
106   font-family: lucida;
107   font-size: 125%;
108 }
109
110 p {
111   color: #white;
112   font-family: lucida;
113   font-size: 100%;
114 }
115
116 tr {
117   color: #white;
118   font-family: lucida;
119   font-size: 100%;
120 }
121
122 table, th, td {
123   border: 1px solid #rgb(255, 255, 255);
124   border-collapse: collapse;
125 }
126
127 table {
128   border-collapse: collapse;
129   width: 350px;
130 }
131
132 th, td {
133   text-align: left;
134   padding: 5px;
135 }
136
137 tr:nth-child(even){background-color: #e0e0e0}
138
139 th {
140   background-color: #ffffff;
141   color: #e0e0e0;
142 }
```

The status bar at the bottom indicates the file is 'index.html' in the 'src' directory, with a line number of 99, column 31, and a UTF-8 encoding.

```

143 padding: 12px 20px;
144 margin: 8px 0;
145 box-sizing: border-box;
146 border: 2px solid #000080;
147 border-radius: 4px;
148 }
149
150
151 </style>
152
153 <script>
154
155     function vertelemovei() {
156
157         var xmlhttp = new XMLHttpRequest();
158         xmlhttp.onreadystatechange = function () {
159             if (this.readyState == 4 && this.status == 200) {
160                 var myAres = JSON.parse(this.responseText);
161                 linhas = "<table><tr><th>id</th><th>nome</th><th>Arma</th><th>ArmaResoluc</th><th>thCamera</th><th>thPreco</th></tr>";
162                 for (i = 0; i < myAres.length; i++) {
163
164                     linhas = linhas + "<tr>";
165                     linhas = linhas + "<td>" + myAres[i].id + "</td>";
166                     linhas = linhas + "<td>" + myAres[i].nome + "</td>";
167                     linhas = linhas + "<td>" + myAres[i].Arma + "</td>";
168                     linhas = linhas + "<td>" + myAres[i].ArmaResoluc + "</td>";
169                     linhas = linhas + "<td>" + myAres[i].resoluc + "</td>";
170                     linhas = linhas + "<td>" + myAres[i].camera + "</td>";
171                     linhas = linhas + "<td>" + myAres[i].preco + "</td>";
172                     linhas = linhas + "</tr>";
173                 }
174                 linhas = linhas + "</table>";
175                 document.getElementById("tabela").innerHTML = linhas;
176             }
177         };
178
179         xmlhttp.open("GET", "http://localhost:8080/telemovei", true);
180         xmlhttp.send();
181     }
182
183     function isEmpty() {
184
185         if (f.idnome.value == "" || f.idArma.value == "" || f.idArmaResoluc.value == ""
186             || f.idResoluc.value == "" || f.idCamera.value == "" || f.idPreco.value == "") {
187             alert("Por favor, preencha os campos obrigatórios!");
188             return false;
189         }
190     }
191
192

```

```

1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
267
```



```
235
236
237 </script>
238
239 <body>
240
241
242 <div id="box1">
243
244 <div id="divdemo" style="display: none">
245 <p id="demo"></p>
246
247 <input class="button" type="button" value="Ocultar" onclick="esconder()">
248
249 </div>
250
251 <h1>Telemóveis</h1>
252
253 <form id="f">
254
255 <h2>Registo de telemóveis</h2>
256
257 <p id="erro">
258
259 </p>
260
261 <p><label for="nome">Nome:</label><br>
262
263 <input type="text" id="idnome" name="nomebox" required:</p>
264
265 <p><label for="ram">Ram:</label><br>
266
267 <input type="text" id="idram" name="rambox"></p>
268
269 <p><label for="armazenamento">Armazenamento:</label><br>
270
271 <input type="text" id="idarmazenamento" name="armazenamentobox"></p>
272
273 <p><label for="resolucao">Resolucao:</label><br>
274
275 <input type="text" id="idresolucao" name="resolucaobox"></p>
276
277
278 <p><label for="camera">Cámara:</label><br>
279
280 <input type="text" id="idcamera" name="camerabox"></p>
281
282
283 <p><label for="preco">Preço:</label><br>
284
285
```

```
286 <input type="text" id="idcamera" name="camerabox"></p>
287
288 <p><label for="preco">Preço:</label><br>
289
290 <input type="text" id="idpreco" name="precbbox"></p>
291
292 <p><input class="button" id="button" value="Inserir" onclick="inserirtelemovel()"></p>
293
294 </form>
295
296 </div>
297
298 <div id="box2">
299
300 <form id="f">
301
302 <p>
303 <h2>Dados</h2>
304
305 </p>
306
307 <p><input class="button" type="button" id="button1" value="Visualizar" onclick="vertelemovel()" style="vertical-align:middle;"></p>
308
309 <div id="tabela"></div>
310
311 </form>
312
313 </div>
314
315 </body>
316
317 </html>
```

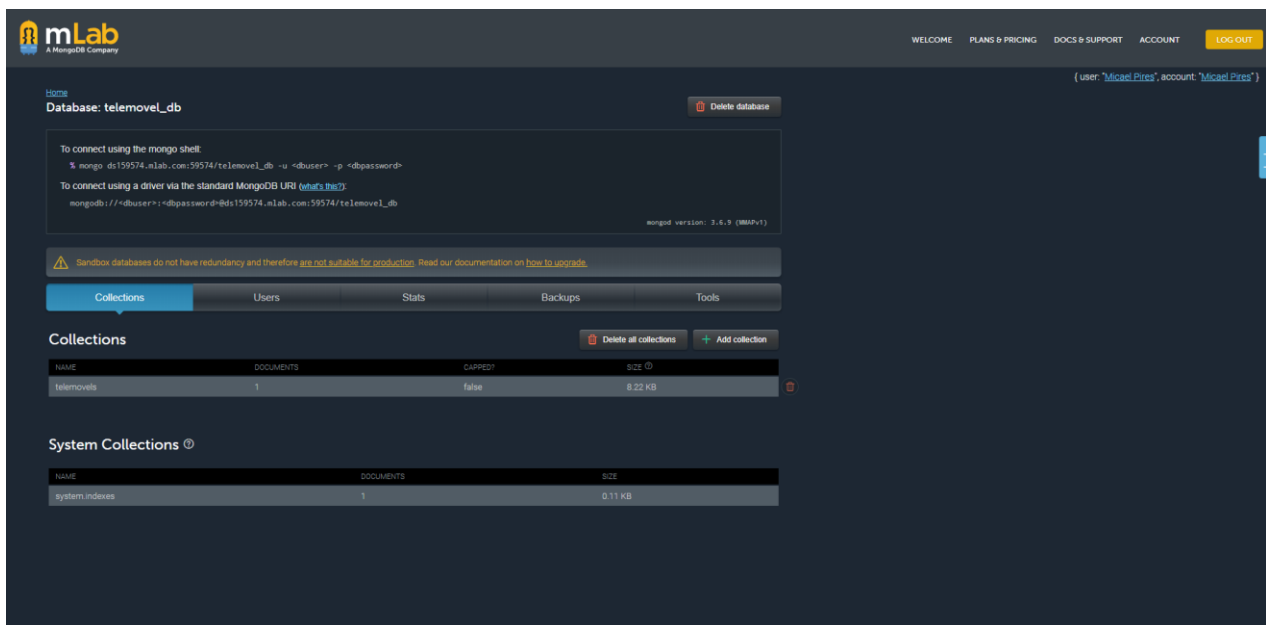
---

# LIGAÇÃO DO SERVIDOR NODEJS E INTERFACE À MBLAB E GOOGLE CLOUD

---

Após a API finalizada e com todas as funcionalidades em pleno funcionamento no **localhost**, passamos então à implementação da base de dados online (neste caso utilizamos o **mLab**).

Criamos uma conta no website e designamos um nome para a base de dados, depois criamos um utilizador na mesma e por fim implementamos o endereço fornecido para a ligação com a **API**, após este processo efetuamos um teste inserindo um dado na base de dados e obtivemos a criação de uma coleção e de um documento que era o esperado.



The screenshot displays the mLab MongoDB web interface. At the top, the mLab logo and navigation links (WELCOME, PLANS & PRICING, DOCS & SUPPORT, ACCOUNT, LOG OUT) are visible. The user is logged in as 'Miguel Pires'. The main section shows the 'Database: telemovel\_db' with a 'Delete database' button. Below this, there are instructions on how to connect using the mongo shell or a standard MongoDB URI. A warning message states: 'Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on how to upgrade.' The 'Collections' tab is selected, showing a table with one collection: 'telemovels' with 1 document, capped at false, and a size of 8.22 KB. There is also a 'System Collections' section showing 'system.indexes' with 1 document and a size of 0.11 KB.

NAME	DOCUMENTS	CAPPED	SIZE
telemovels	1	false	8.22 KB

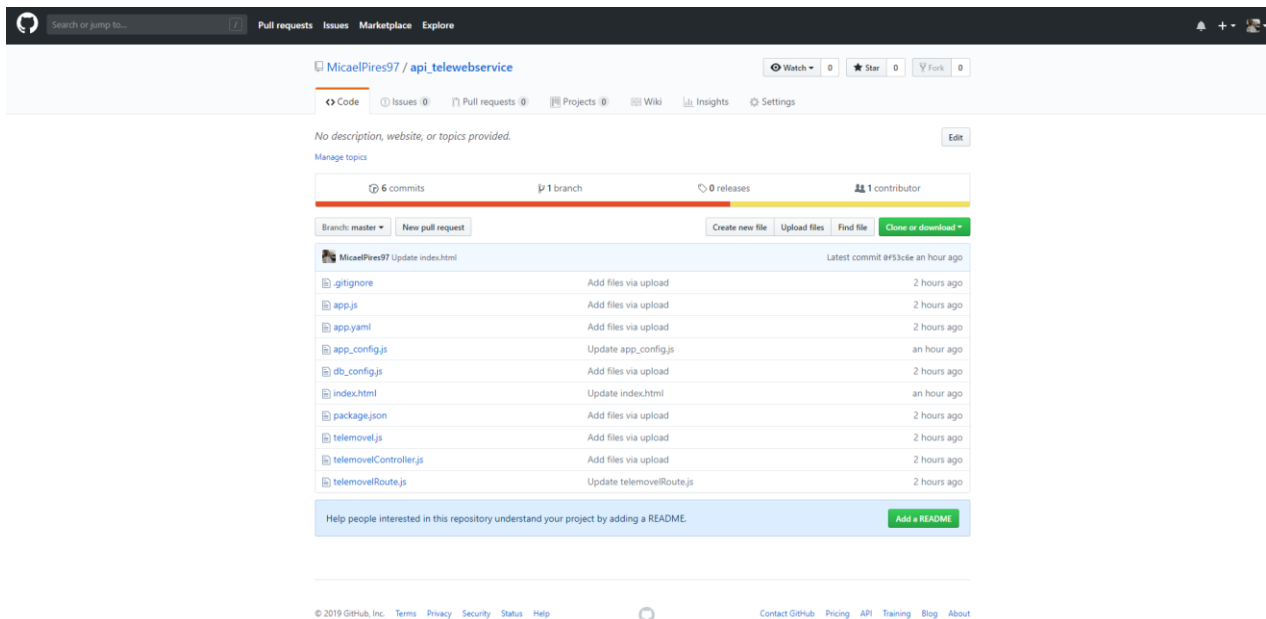
  

NAME	DOCUMENTS	SIZE
system.indexes	1	0.11 KB

# Virtualização do projeto

Para fazermos a virtualização do projeto utilizamos o **GitHub** e a **Google Cloud**.

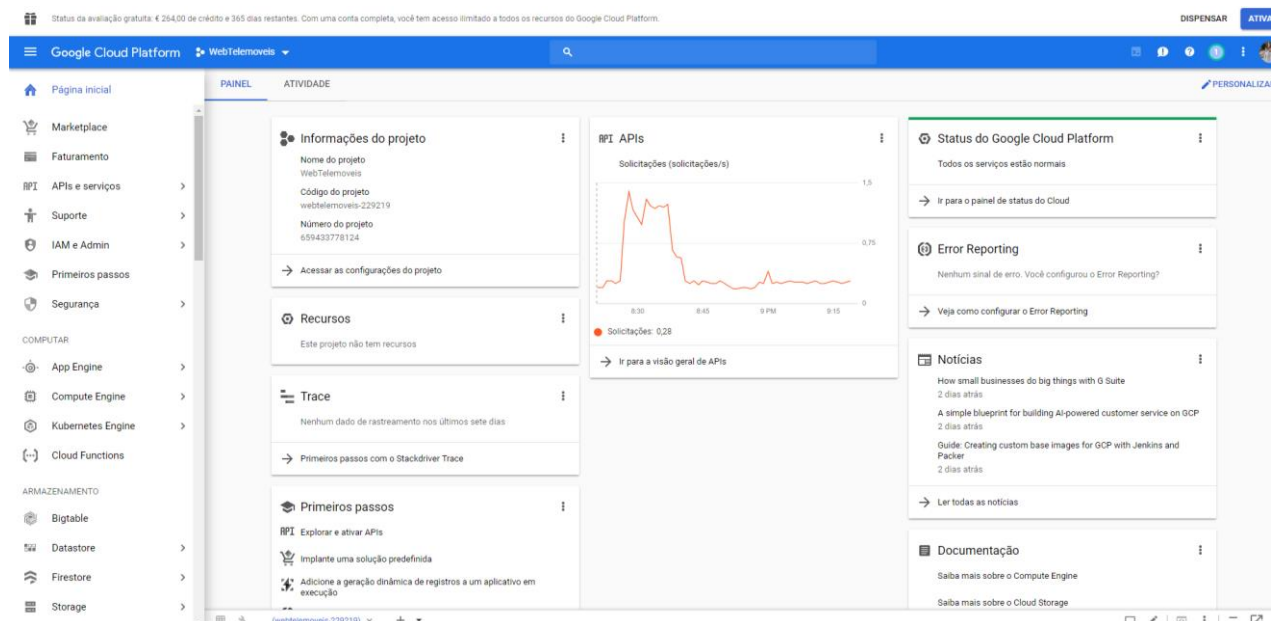
De seguida criamos um repositório para ficheiros necessários da API e adicionamos os ficheiros ao mesmo



Os seguintes ficheiros (app.yaml e .gitignore) foram criados de raiz, o ficheiro app.yaml é necessário para o correto funcionamento da ferramenta App engine do google cloud e o .gitignore tem como objetivo ignorar a necessidade dos módulos proveniente dos ficheiros, visto que o **GitHub** já os fornece.

Já no ficheiro package.json fazemos uma pequena alteração no script acrescentando (**"start": "node app.js"**).

Passando para o **Google Cloud**, criamos um projeto e dentro da **Home Page** abrimos o **Cloud Shell**.



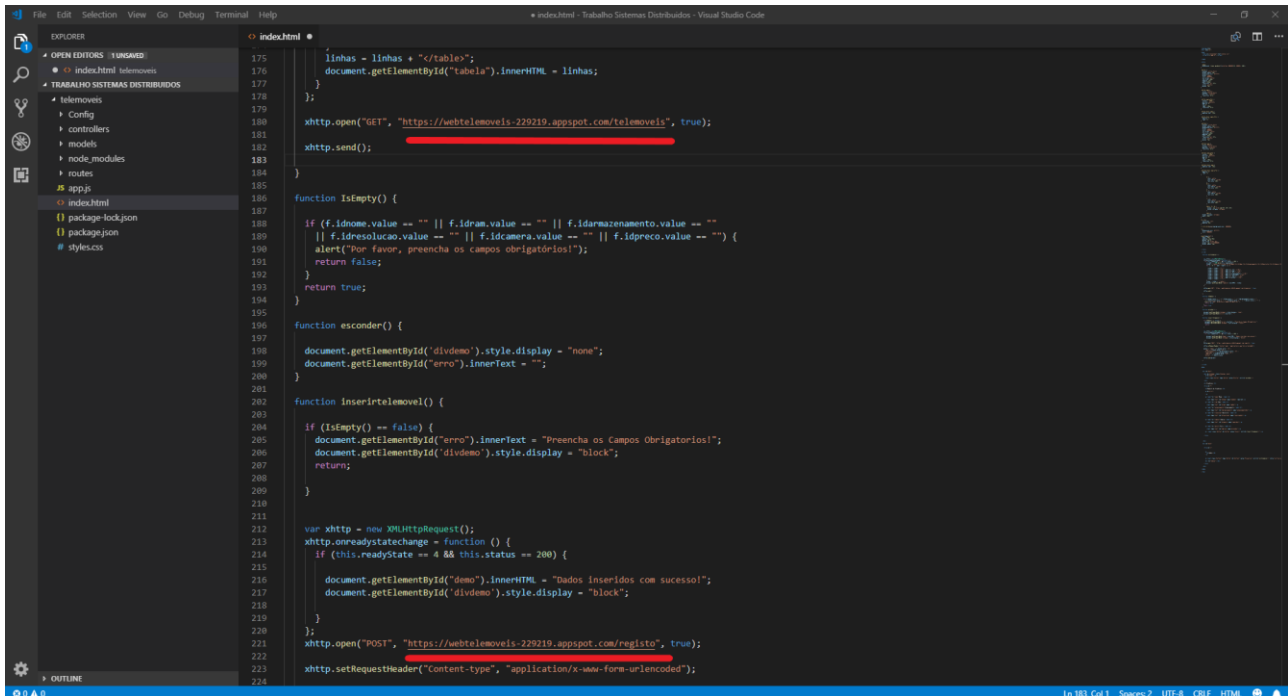
Dentro do cloud Shell fizemos um clone do repositório do **GitHub** através do comando

**“git clone [https://github.com/MicaelPires97/api\\_telewebsevice](https://github.com/MicaelPires97/api_telewebsevice)”,** e depois acedemos ao repositório **cd api\_telewebsevice** e efetuamos o deploy para a app engine através do comando **“gcloud app deploy”**.

## Publicação da API

Após este processo fizemos a publicação da app através do comando **“gcloud app browse”**, se todo o processo foi feito corretamente será nos fornecido um link onde fora publicada.

Após obtermos o link vamos substituí-lo pelo **localhost:8080** nas functions **GET** e **POST** no ficheiro **index.html** dentro do repositório do **GitHub**



```
175     linhas = linhas + "</table>";
176     document.getElementById("tabela").innerHTML = linhas;
177 }
178
179 xhttp.open("GET", "https://webtelemovels-229219.appspot.com/telemovels", true);
180 xhttp.send();
181
182
183
184
185
186 function isEmpty() {
187
188     if (f.idnome.value == "" || f.idrm.value == "" || f.idrazonamento.value == ""
189     || f.idresolucao.value == "" || f.idcamera.value == "" || f.idpreco.value == "") {
190         alert("Por favor, preencha os campos obrigatórios!");
191         return false;
192     }
193     return true;
194 }
195
196 function esconder() {
197     document.getElementById("divdemo").style.display = "none";
198     document.getElementById("erro").innerText = "";
199 }
200
201
202 function inserirTelemovel() {
203
204     if (isEmpty() == false) {
205         document.getElementById("erro").innerText = "Preencha os Campos Obrigatórios!";
206         document.getElementById("divdemo").style.display = "block";
207         return;
208     }
209
210
211
212     var xhttp = new XMLHttpRequest();
213     xhttp.onreadystatechange = function () {
214         if (this.readyState == 4 && this.status == 200) {
215
216             document.getElementById("demo").innerHTML = "Dados inseridos com sucesso!";
217             document.getElementById("divdemo").style.display = "block";
218
219         }
220     };
221     xhttp.open("POST", "https://webtelemovels-229219.appspot.com/registo", true);
222     xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
223 }
```

E de seguida saímos dos repositório através do comando **"cd.."** e executamos o comando **"rm -rf api\_telewebsevice"** e voltamos a refazer o processo de deploy repetindo os comandos

**"git clone [https://github.com/MicaelPires97/api\\_telewebsevice](https://github.com/MicaelPires97/api_telewebsevice)"**,

**"gcloud app deploy"**

**"gcloud app browse"**

Por fim o projeto estará totalmente funcional.