

# OFICINA DE INTERNET DAS COISAS



## A OFICINA



IMERSÃO NO  
**COTIDIANO**  
DO LAB

**TECNOLOGIAS**  
PARA INTERNET  
DAS COISAS

**PROGRAMAÇÃO**  
DE SISTEMAS  
EMBARCADOS

MONTAGEM DE  
**PROTÓTIPOS**

## RESULTADOS

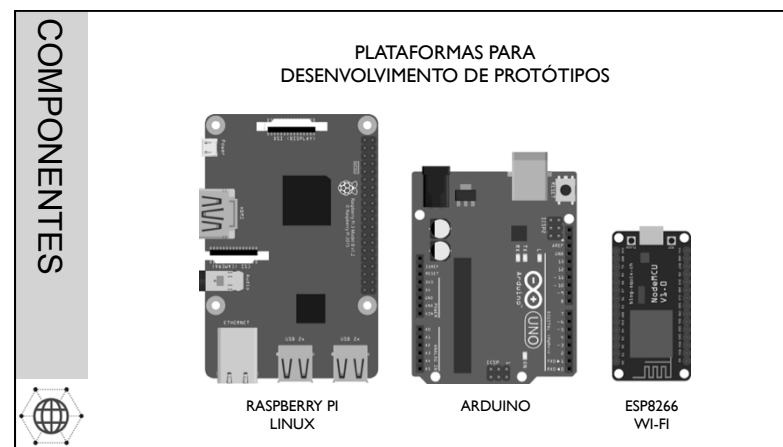
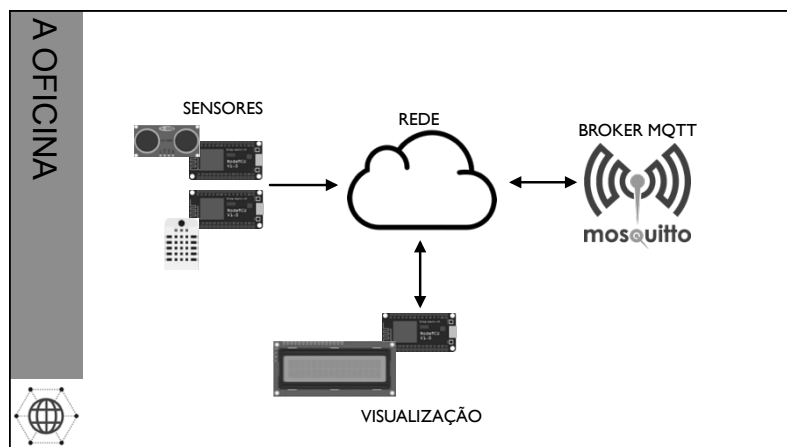
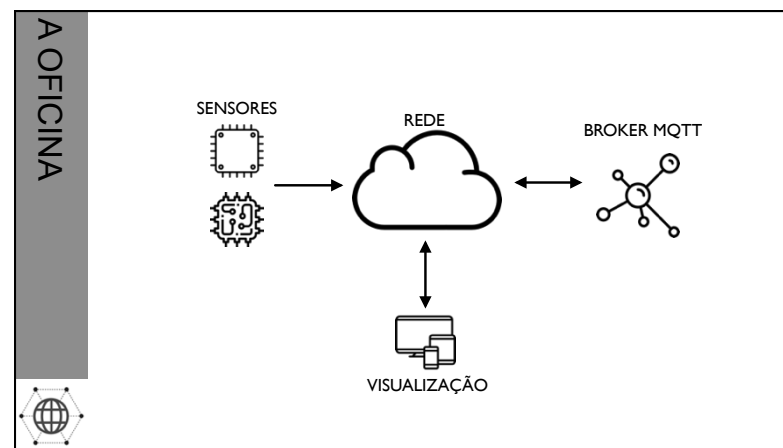
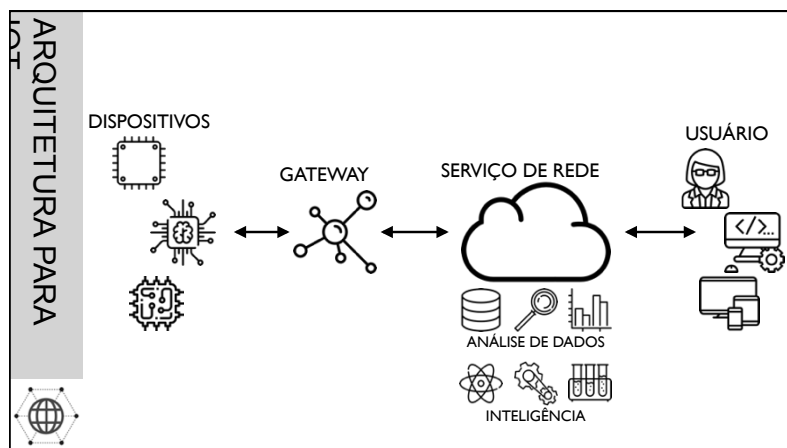


REFLEXÃO  
SOBRE AS  
**POTENCIALIDADES**  
DAS TECNOLOGIAS  
VIVENCIADAS

## PALESTRA



Prof. Eduardo Pellanda:  
“Cidade que sente”



**A OFICINA**

PLATAFORMAS PARA DESENVOLVIMENTO DE PROTÓTIPOS

ESP8266  
Wi-Fi

AMBIENTE ARDUINO

Initializing packages...

**A OFICINA**

NodeMCU V1.0

AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE. LEARN MORE ABOUT THE CONTRIBUTORS OF [ARDUINO.CC](https://www.arduino.cc/en/contribute) ON [arduino.cc/contribute](https://www.arduino.cc/en/contribute)

Initializing packages...

Aplicação

Bibliotecas

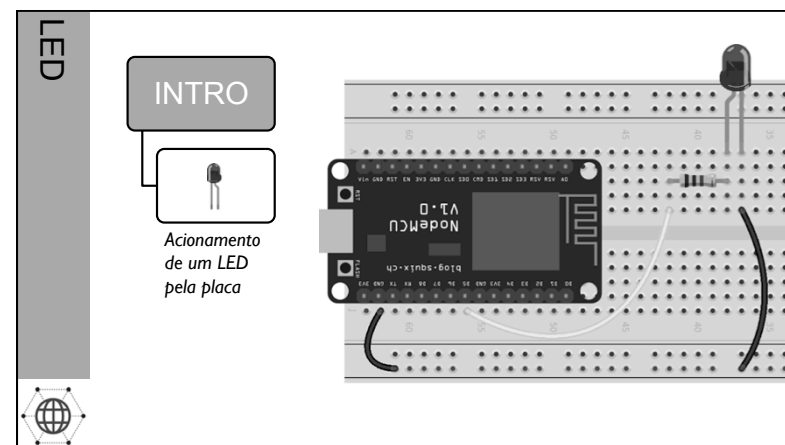
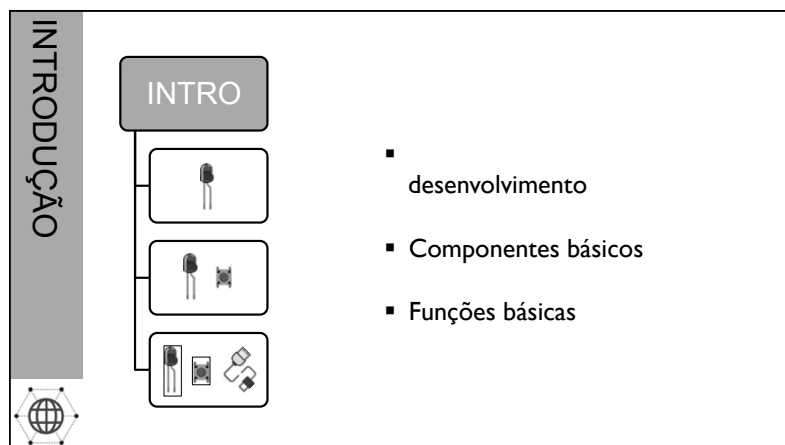
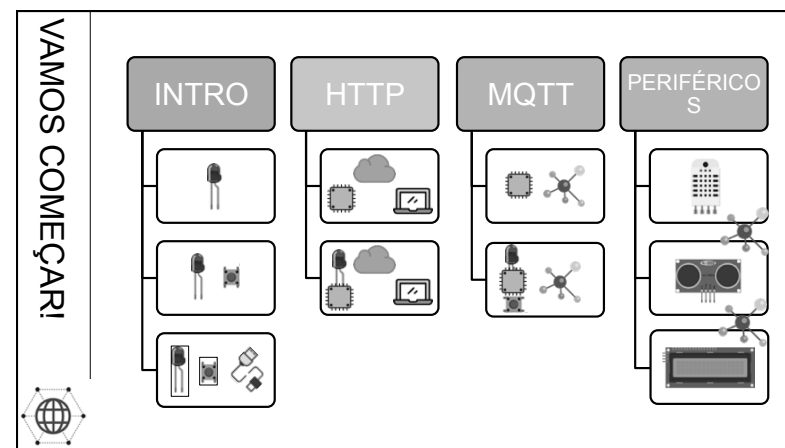
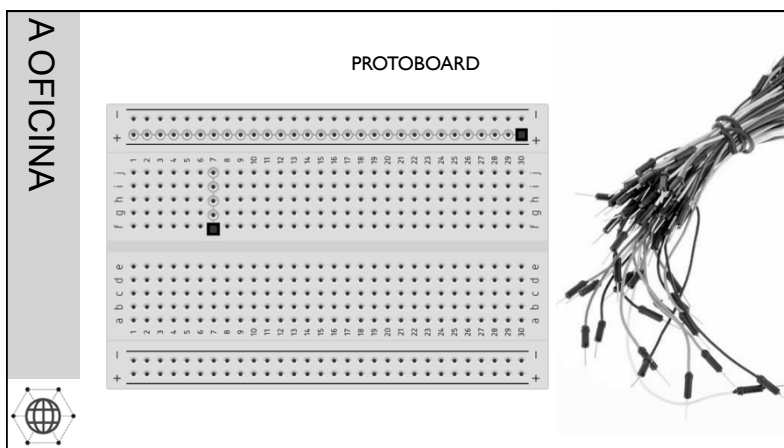
Hardware Abstraction Layer

**A OFICINA**

PLATAFORMAS PARA DESENVOLVIMENTO DE PROTÓTIPOS

Arduino Uno

GP0016  
GP002  
GP004  
GP006  
GP008  
GP010  
GP012  
GP014  
GP016  
GP018  
GP020  
GP022  
GP024  
GP026  
GP028  
GP030  
GP032  
GP034  
GP036  
GP038  
GP040  
GP042  
GP044  
GP046  
GP048  
GP050  
GP052  
GP054  
GP056  
GP058  
GP060  
GP062  
GP064  
GP066  
GP068  
GP070  
GP072  
GP074  
GP076  
GP078  
GP080  
GP082  
GP084  
GP086  
GP088  
GP090  
GP092  
GP094  
GP096  
GP098  
GP100  
GP102  
GP104  
GP106  
GP108  
GP110  
GP112  
GP114  
GP116  
GP118  
GP120  
GP122  
GP124  
GP126  
GP128  
GP130  
GP132  
GP134  
GP136  
GP138  
GP140  
GP142  
GP144  
GP146  
GP148  
GP150  
GP152  
GP154  
GP156  
GP158  
GP160  
GP162  
GP164  
GP166  
GP168  
GP170  
GP172  
GP174  
GP176  
GP178  
GP180  
GP182  
GP184  
GP186  
GP188  
GP190  
GP192  
GP194  
GP196  
GP198  
GP200  
GP202  
GP204  
GP206  
GP208  
GP210  
GP212  
GP214  
GP216  
GP218  
GP220  
GP222  
GP224  
GP226  
GP228  
GP230  
GP232  
GP234  
GP236  
GP238  
GP240  
GP242  
GP244  
GP246  
GP248  
GP250  
GP252  
GP254  
GP256  
GP258  
GP260  
GP262  
GP264  
GP266  
GP268  
GP270  
GP272  
GP274  
GP276  
GP278  
GP280  
GP282  
GP284  
GP286  
GP288  
GP290  
GP292  
GP294  
GP296  
GP298  
GP300  
GP302  
GP304  
GP306  
GP308  
GP310  
GP312  
GP314  
GP316  
GP318  
GP320  
GP322  
GP324  
GP326  
GP328  
GP330  
GP332  
GP334  
GP336  
GP338  
GP340  
GP342  
GP344  
GP346  
GP348  
GP350  
GP352  
GP354  
GP356  
GP358  
GP360  
GP362  
GP364  
GP366  
GP368  
GP370  
GP372  
GP374  
GP376  
GP378  
GP380  
GP382  
GP384  
GP386  
GP388  
GP390  
GP392  
GP394  
GP396  
GP398  
GP400  
GP402  
GP404  
GP406  
GP408  
GP410  
GP412  
GP414  
GP416  
GP418  
GP420  
GP422  
GP424  
GP426  
GP428  
GP430  
GP432  
GP434  
GP436  
GP438  
GP440  
GP442  
GP444  
GP446  
GP448  
GP450  
GP452  
GP454  
GP456  
GP458  
GP460  
GP462  
GP464  
GP466  
GP468  
GP470  
GP472  
GP474  
GP476  
GP478  
GP480  
GP482  
GP484  
GP486  
GP488  
GP490  
GP492  
GP494  
GP496  
GP498  
GP500  
GP502  
GP504  
GP506  
GP508  
GP510  
GP512  
GP514  
GP516  
GP518  
GP520  
GP522  
GP524  
GP526  
GP528  
GP530  
GP532  
GP534  
GP536  
GP538  
GP540  
GP542  
GP544  
GP546  
GP548  
GP550  
GP552  
GP554  
GP556  
GP558  
GP560  
GP562  
GP564  
GP566  
GP568  
GP570  
GP572  
GP574  
GP576  
GP578  
GP580  
GP582  
GP584  
GP586  
GP588  
GP590  
GP592  
GP594  
GP596  
GP598  
GP600  
GP602  
GP604  
GP606  
GP608  
GP610  
GP612  
GP614  
GP616  
GP618  
GP620  
GP622  
GP624  
GP626  
GP628  
GP630  
GP632  
GP634  
GP636  
GP638  
GP640  
GP642  
GP644  
GP646  
GP648  
GP650  
GP652  
GP654  
GP656  
GP658  
GP660  
GP662  
GP664  
GP666  
GP668  
GP670  
GP672  
GP674  
GP676  
GP678  
GP680  
GP682  
GP684  
GP686  
GP688  
GP690  
GP692  
GP694  
GP696  
GP698  
GP700  
GP702  
GP704  
GP706  
GP708  
GP710  
GP712  
GP714  
GP716  
GP718  
GP720  
GP722  
GP724  
GP726  
GP728  
GP730  
GP732  
GP734  
GP736  
GP738  
GP740  
GP742  
GP744  
GP746  
GP748  
GP750  
GP752  
GP754  
GP756  
GP758  
GP760  
GP762  
GP764  
GP766  
GP768  
GP770  
GP772  
GP774  
GP776  
GP778  
GP780  
GP782  
GP784  
GP786  
GP788  
GP790  
GP792  
GP794  
GP796  
GP798  
GP800  
GP802  
GP804  
GP806  
GP808  
GP810  
GP812  
GP814  
GP816  
GP818  
GP820  
GP822  
GP824  
GP826  
GP828  
GP830  
GP832  
GP834  
GP836  
GP838  
GP840  
GP842  
GP844  
GP846  
GP848  
GP850  
GP852  
GP854  
GP856  
GP858  
GP860  
GP862  
GP864  
GP866  
GP868  
GP870  
GP872  
GP874  
GP876  
GP878  
GP880  
GP882  
GP884  
GP886  
GP888  
GP890  
GP892  
GP894  
GP896  
GP898  
GP900  
GP902  
GP904  
GP906  
GP908  
GP910  
GP912  
GP914  
GP916  
GP918  
GP920  
GP922  
GP924  
GP926  
GP928  
GP930  
GP932  
GP934  
GP936  
GP938  
GP940  
GP942  
GP944  
GP946  
GP948  
GP950  
GP952  
GP954  
GP956  
GP958  
GP960  
GP962  
GP964  
GP966  
GP968  
GP970  
GP972  
GP974  
GP976  
GP978  
GP980  
GP982  
GP984  
GP986  
GP988  
GP990  
GP992  
GP994  
GP996  
GP998  
GP1000  
GP1002  
GP1004  
GP1006  
GP1008  
GP1010  
GP1012  
GP1014  
GP1016  
GP1018  
GP1020  
GP1022  
GP1024  
GP1026  
GP1028  
GP1030  
GP1032  
GP1034  
GP1036  
GP1038  
GP1040  
GP1042  
GP1044  
GP1046  
GP1048  
GP1050  
GP1052  
GP1054  
GP1056  
GP1058  
GP1060  
GP1062  
GP1064  
GP1066  
GP1068  
GP1070  
GP1072  
GP1074  
GP1076  
GP1078  
GP1080  
GP1082  
GP1084  
GP1086  
GP1088  
GP1090  
GP1092  
GP1094  
GP1096  
GP1098  
GP1100  
GP1102  
GP1104  
GP1106  
GP1108  
GP1110  
GP1112  
GP1114  
GP1116  
GP1118  
GP1120  
GP1122  
GP1124  
GP1126  
GP1128  
GP1130  
GP1132  
GP1134  
GP1136  
GP1138  
GP1140  
GP1142  
GP1144  
GP1146  
GP1148  
GP1150  
GP1152  
GP1154  
GP1156  
GP1158  
GP1160  
GP1162  
GP1164  
GP1166  
GP1168  
GP1170  
GP1172  
GP1174  
GP1176  
GP1178  
GP1180  
GP1182  
GP1184  
GP1186  
GP1188  
GP1190  
GP1192  
GP1194  
GP1196  
GP1198  
GP1200  
GP1202  
GP1204  
GP1206  
GP1208  
GP1210  
GP1212  
GP1214  
GP1216  
GP1218  
GP1220  
GP1222  
GP1224  
GP1226  
GP1228  
GP1230  
GP1232  
GP1234  
GP1236  
GP1238  
GP1240  
GP1242  
GP1244  
GP1246  
GP1248  
GP1250  
GP1252  
GP1254  
GP1256  
GP1258  
GP1260  
GP1262  
GP1264  
GP1266  
GP1268  
GP1270  
GP1272  
GP1274  
GP1276  
GP1278  
GP1280  
GP1282  
GP1284  
GP1286  
GP1288  
GP1290  
GP1292  
GP1294  
GP1296  
GP1298  
GP1300  
GP1302  
GP1304  
GP1306  
GP1308  
GP1310  
GP1312  
GP1314  
GP1316  
GP1318  
GP1320  
GP1322  
GP1324  
GP1326  
GP1328  
GP1330  
GP1332  
GP1334  
GP1336  
GP1338  
GP1340  
GP1342  
GP1344  
GP1346  
GP1348  
GP1350  
GP1352  
GP1354  
GP1356  
GP1358  
GP1360  
GP1362  
GP1364  
GP1366  
GP1368  
GP1370  
GP1372  
GP1374  
GP1376  
GP1378  
GP1380  
GP1382  
GP1384  
GP1386  
GP1388  
GP1390  
GP1392  
GP1394  
GP1396  
GP1398  
GP1400  
GP1402  
GP1404  
GP1406  
GP1408  
GP1410  
GP1412  
GP1414  
GP1416  
GP1418  
GP1420  
GP1422  
GP1424  
GP1426  
GP1428  
GP1430  
GP1432  
GP1434  
GP1436  
GP1438  
GP1440  
GP1442  
GP1444  
GP1446  
GP1448  
GP1450  
GP1452  
GP1454  
GP1456  
GP1458  
GP1460  
GP1462  
GP1464  
GP1466  
GP1468  
GP1470  
GP1472  
GP1474  
GP1476  
GP1478  
GP1480  
GP1482  
GP1484  
GP1486  
GP1488  
GP1490  
GP1492  
GP1494  
GP1496  
GP1498  
GP1500  
GP1502  
GP1504  
GP1506  
GP1508  
GP1510  
GP1512  
GP1514  
GP1516  
GP1518  
GP1520  
GP1522  
GP1524  
GP1526  
GP1528  
GP1530  
GP1532  
GP1534  
GP1536  
GP1538  
GP1540  
GP1542  
GP1544  
GP1546  
GP1548  
GP1550  
GP1552  
GP1554  
GP1556  
GP1558  
GP1560  
GP1562  
GP1564  
GP1566  
GP1568  
GP1570  
GP1572  
GP1574  
GP1576  
GP1578  
GP1580  
GP1582  
GP1584  
GP1586  
GP1588  
GP1590  
GP1592  
GP1594  
GP1596  
GP1598  
GP1600  
GP1602  
GP1604  
GP1606  
GP1608  
GP1610  
GP1612  
GP1614  
GP1616  
GP1618  
GP1620  
GP1622  
GP1624  
GP1626  
GP1628  
GP1630  
GP1632  
GP1634  
GP1636  
GP1638  
GP1640  
GP1642  
GP1644  
GP1646  
GP1648  
GP1650  
GP1652  
GP1654  
GP1656  
GP1658  
GP1660  
GP1662  
GP1664  
GP1666  
GP1668  
GP1670  
GP1672  
GP1674  
GP1676  
GP1678  
GP1680  
GP1682  
GP1684  
GP1686  
GP1688  
GP1690  
GP1692  
GP1694  
GP1696  
GP1698  
GP1700  
GP1702  
GP1704  
GP1706  
GP1708  
GP1710  
GP1712  
GP1714  
GP1716  
GP1718  
GP1720  
GP1722  
GP1724  
GP1726  
GP1728  
GP1730  
GP1732  
GP1734  
GP1736  
GP1738  
GP1740  
GP1742  
GP1744  
GP1746  
GP1748  
GP1750  
GP1752  
GP1754  
GP1756  
GP1758  
GP1760  
GP1762  
GP1764  
GP1766  
GP1768  
GP1770  
GP1772  
GP1774  
GP1776  
GP1778  
GP1780  
GP1782  
GP1784  
GP1786  
GP1788  
GP1790  
GP1792  
GP1794  
GP1796  
GP1798  
GP1800  
GP1802  
GP1804  
GP1806  
GP1808  
GP1810  
GP1812  
GP1814  
GP1816  
GP1818  
GP1820  
GP1822  
GP1824  
GP1826  
GP1828  
GP1830  
GP1832  
GP1834  
GP1836  
GP1838  
GP1840  
GP1842  
GP1844  
GP1846  
GP1848  
GP1850  
GP1852  
GP1854  
GP1856  
GP1858  
GP1860  
GP1862  
GP1864  
GP1866  
GP1868  
GP1870  
GP1872  
GP1874  
GP1876  
GP1878  
GP1880  
GP1882  
GP1884  
GP1886  
GP1888  
GP1890  
GP1892  
GP1894  
GP1896  
GP1898  
GP1900  
GP1902  
GP1904  
GP1906  
GP1908  
GP1910  
GP1912  
GP1914  
GP1916  
GP1918  
GP1920  
GP1922  
GP1924  
GP1926  
GP1928  
GP1930  
GP1932  
GP1934  
GP1936  
GP1938  
GP1940  
GP1942  
GP1944  
GP1946  
GP1948  
GP1950  
GP1952  
GP1954  
GP1956  
GP1958  
GP1960  
GP1962  
GP1964  
GP1966  
GP1968  
GP1970  
GP1972  
GP1974  
GP1976  
GP1978  
GP1980  
GP1982  
GP1984  
GP1986  
GP1988  
GP1990  
GP1992  
GP1994  
GP1996  
GP1998  
GP2000  
GP2002  
GP2004  
GP2006  
GP2008  
GP2010  
GP2012  
GP2014  
GP2016  
GP2018  
GP2020  
GP2022  
GP2024  
GP2026  
GP2028  
GP2030  
GP2032  
GP2034  
GP2036  
GP2038  
GP2040  
GP2042  
GP2044  
GP2046  
GP2048  
GP2050  
GP2052  
GP2054  
GP2056  
GP2058  
GP2060  
GP2062  
GP2064  
GP2066  
GP2068  
GP2070  
GP2072  
GP2074  
GP2076  
GP2078  
GP2080  
GP2082  
GP2084  
GP2086  
GP2088  
GP2090  
GP2092  
GP2094  
GP2096  
GP2098  
GP2100  
GP2102  
GP2104  
GP2106  
GP2108  
GP2110  
GP2112  
GP2114  
GP2116  
GP2118  
GP2120  
GP2122  
GP2124  
GP2126  
GP2128  
GP2130  
GP2132  
GP2134  
GP2136  
GP2138  
GP2140  
GP2142  
GP2144  
GP2146  
GP2148  
GP2150  
GP2152  
GP2154  
GP2156  
GP2158  
GP2160  
GP2162  
GP2164  
GP2166  
GP2168  
GP2170  
GP2172  
GP2174  
GP2176  
GP2178  
GP2180  
GP2182  
GP2184  
GP2186  
GP2188  
GP2190  
GP2192  
GP2194  
GP2196  
GP2198  
GP2200  
GP2202  
GP2204  
GP2206  
GP2208  
GP2210  
GP2212  
GP2214  
GP2216  
GP2218  
GP2220  
GP2222  
GP2224  
GP2226  
GP2228  
GP2230  
GP2232  
GP2234  
GP2236  
GP2238  
GP2240  
GP2242  
GP2244  
GP2246  
GP2248  
GP2250  
GP2252  
GP2254  
GP2256  
GP2258  
GP2260  
GP2262  
GP2264  
GP2266  
GP2268  
GP2270  
GP2272  
GP2274  
GP2276  
GP2278  
GP2280  
GP2282  
GP2284  
GP2286  
GP2288  
GP2290  
GP2292  
GP2294  
GP2296  
GP2298  
GP2300  
GP2302  
GP2304  
GP2306  
GP2308  
GP2310  
GP2312  
GP2314  
GP2316  
GP2318  
GP2320  
GP2322  
GP2324  
GP2326  
GP2328  
GP2330  
GP2332  
GP2334  
GP2336  
GP2338  
GP2340  
GP2342  
GP2344  
GP2346  
GP2348  
GP2350  
GP2352  
GP2354  
GP2356  
GP2358  
GP2360  
GP2362  
GP2364  
GP2366  
GP2368  
GP2370  
GP2372  
GP2374  
GP2376  
GP2378  
GP2380  
GP2382  
GP2384  
GP2386  
GP2388  
GP2390  
GP2392  
GP2394  
GP2396  
GP2398  
GP2400  
GP2402  
GP2404  
GP2406  
GP2408  
GP2410  
GP2412  
GP2414  
GP2416  
GP2418  
GP2420  
GP2422  
GP2424  
GP2426  
GP2428  
GP2430  
GP2432  
GP2434  
GP2436  
GP2438  
GP2440  
GP2442  
GP2444  
GP2446  
GP2448  
GP2450  
GP2452  
GP2454  
GP2456  
GP2458  
GP2460  
GP2462  
GP2464  
GP2466  
GP2468  
GP2470  
GP2472  
GP2474  
GP2476  
GP2478  
GP2480  
GP2482  
GP2484  
GP2486  
GP2488  
GP2490  
GP2492  
GP2494  
GP2496  
GP2498  
GP2500  
GP2502  
GP2504  
GP2506  
GP2508  
GP2510  
GP2512  
GP2514  
GP2516  
GP2518  
GP2520  
GP2522  
GP2524  
GP2526  
GP2528  
GP2530  
GP2532  
GP2534  
GP2536  
GP2538  
GP2540  
GP2542  
GP2544  
GP2546  
GP2548  
GP2550  
GP2552  
GP2554  
GP2556  
GP2558  
GP2560  
GP2562  
GP2564  
GP2566  
GP2568  
GP2570  
GP2572  
GP2574  
GP2576  
GP2578  
GP2580  
GP2582  
GP2584  
GP2586  
GP2588  
GP2590  
GP2592  
GP2594  
GP2596  
GP2598  
GP2600  
GP2602  
GP2604  
GP2606  
GP2608  
GP2610  
GP2612  
GP2614  
GP2616  
GP2618  
GP2620  
GP2622  
GP2624  
GP2626  
GP2628  
GP2630  
GP2632  
GP2634  
GP2636  
GP2638  
GP2640  
GP2642  
GP2644  
GP2646  
GP2648  
GP2650  
GP2652  
GP2654  
GP2656  
GP2658  
GP2660  
GP2662  
GP2664  
GP2666  
GP2668  
GP2670  
GP2672  
GP2674  
GP2676  
GP2678  
GP2680  
GP2682  
GP2684  
GP2686  
GP2688  
GP2690  
GP2692  
GP2694  
GP2696  
GP2698  
GP2700  
GP2702  
GP2704  
GP2706  
GP2708  
GP2710  
GP2712  
GP2714  
GP2716  
GP2718  
GP2720  
GP2722  
GP2724  
GP2726  
GP2728  
GP2730  
GP2732  
GP2734  
GP2736  
GP2738  
GP2740  
GP2742  
GP2744  
GP2746  
GP2748  
GP2750  
GP2752  
GP2754  
GP2756  
GP2758  
GP2760  
GP2762  
GP2764  
GP2766  
GP2768  
GP2770  
GP2772  
GP2774  
GP2776  
GP2778  
GP2780  
GP2782  
GP2784  
GP2786  
GP2788  
GP2790  
GP2792  
GP2794  
GP2796  
GP2798  
GP2800  
GP2802  
GP2804  
GP2806  
GP2808  
GP2810  
GP2812  
GP2814  
GP2816  
GP2818  
GP2820  
GP2822  
GP2824  
GP2826  
GP2828  
GP2830  
GP2832  
GP2834  
GP2836  
GP2838  
GP2840  
GP2842  
GP2844  
GP2846  
GP2848  
GP2850  
GP2852  
GP2854  
GP2856  
GP2858  
GP2860  
GP2862  
GP2864  
GP2866  
GP2868  
GP2870  
GP2872  
GP2874  
GP2876  
GP2878  
GP2880  
GP2882  
GP2884  
GP2886  
GP2888  
GP2890  
GP2892  
GP2894  
GP2896  
GP2898  
GP2900  
GP2902  
GP2904  
GP2906  
GP2908  
GP2910  
GP2912  
GP2914  
GP2916  
GP2918  
GP2920  
GP2922  
GP2924  
GP2926  
GP2928  
GP2930  
GP2932  
GP2934  
GP2936  
GP2938  
GP2940  
GP2942  
GP2944  
GP2946  
GP2948  
GP2950  
GP2952  
GP2954  
GP2956  
GP2958  
GP2960  
GP2962  
GP2964  
GP2966  
GP2968  
GP2970  
GP2972  
GP2974  
GP2976  
GP2978  
GP2980  
GP2982  
GP2984  
GP2986  
GP2988  
GP2990  
GP2992  
GP2994  
GP2996  
GP2998  
GP3000  
GP3002  
GP3004  
GP3006  
GP3008  
GP3010  
GP3012  
GP3014  
GP3016  
GP3018  
GP3020  
GP3022  
GP3024  
GP3026  
GP3028  
GP3030  
GP3032  
GP3034  
GP3036  
GP3038  
GP3040  
GP3042  
GP3044  
GP3046  
GP3048  
GP3050  
GP3052  
GP3054  
GP3056  
GP3058  
GP3060  
GP3062  
GP3064  
GP3066  
GP3068  
GP3070  
GP3072  
GP3074  
GP3076  
GP3078  
GP3080  
GP3082  
GP3084  
GP3086  
GP3088  
GP3090  
GP3092  
GP3094  
GP3096  
GP3098  
GP3100  
GP3102  
GP3104  
GP3106  
GP3108  
GP3110  
GP3112  
GP3114  
GP3116  
GP3118  
GP3120  
GP3122  
GP3124  
GP3126  
GP3128  
GP3130  
GP3132  
GP3134  
GP3136  
GP3138  
GP3140  
GP3142  
GP3144  
GP3146  
GP3148  
GP3150  
GP3152  
GP3154  
GP3156  
GP3158  
GP3160  
GP3162  
GP3164  
GP3166  
GP3168  
GP3170  
GP3172  
GP3174  
GP3176  
GP3178  
GP3180  
GP3182  
GP3184  
GP3186  
GP3188  
GP3190  
GP3192  
GP3194  
GP3196  
GP3198  
GP3200  
GP3202  
GP3204  
GP3206  
GP3208  
GP3210  
GP3212  
GP3214  
GP3216  
GP3218  
GP3220  
GP3222  
GP3224  
GP3226  
GP3228  
GP3230  
GP3232  
GP3234  
GP3236  
GP3238  
GP3240  
GP3242  
GP3244  
GP3246  
GP3248  
GP3250  
GP3252  
GP3254  
GP3256  
GP3258  
GP3260  
GP3262  
GP3264  
GP3266  
GP3268  
GP3270  
GP3272  
GP3274  
GP3276  
GP3278  
GP3280  
GP3282  
GP3284  
GP3286  
GP3288  
GP3290  
GP3292  
GP3294  
GP3296  
GP3298  
GP3300  
GP3302  
GP3304  
GP3306  
GP3308  
GP3310  
GP3312  
GP3314  
GP3316  
GP3318  
GP3320  
GP3322  
GP3324  
GP3326  
GP3328  
GP3330  
GP3332  
GP3334  
GP3336  
GP3338  
GP3340  
GP3342  
GP3344  
GP3346  
GP3348  
GP3350  
GP3352  
GP3354  
GP3356  
GP3358  
GP3360  
GP3362  
GP3364  
GP3366  
GP3368  
GP3370  
GP3372  
GP3374  
GP3376  
GP3378  
GP3380  
GP3382  
GP3384  
GP3386  
GP3388  
GP3390  
GP3392  
GP3394  
GP3396  
GP3398  
GP3400  
GP3402  
GP3404  
GP3406  
GP3408  
GP3410  
GP3412  
GP3414  
GP3416  
GP3418  
GP3420  
GP3422  
GP3424  
GP3426  
GP3428  
GP3430  
GP3432  
GP3434  
GP3436  
GP3438  
GP3440  
GP3442  
GP3444  
GP3446  
GP3448  
GP3450  
GP3452  
GP3454  
GP3456  
GP3458  
GP3460  
GP3462  
GP3464  
GP3466  
GP3468  
GP3470  
GP3472  
GP3474  
GP3476  
GP3478  
GP3480  
GP3482  
GP3484  
GP3486  
GP3488  
GP3490  
GP3492  
GP3494  
GP3496  
GP3498  
GP3500  
GP3502  
GP3504  
GP3506  
GP3508  
GP3510  
GP3512  
GP3514  
GP3516  
GP3518  
GP3520  
GP3522  
GP3524  
GP3526  
GP3528  
GP3530  
GP3532  
GP3534  
GP3536  
GP3538  
GP3540  
GP3542  
GP3544  
GP3546  
GP3548  
GP3550



## LED

## INTRO



Acionamento  
de um LED  
pela placa



```
#define LED_PIN D5
#define DELAY_TIME 1000

void setup()
{
  pinMode(LED_PIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_PIN, HIGH);
  delay(DELAY_TIME);

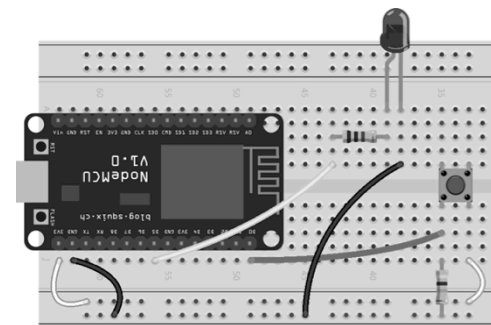
  digitalWrite(LED_PIN, LOW);
  delay(DELAY_TIME);
}
```

## LED E BOTÃO

## INTRO



Botão para  
acionar o LED



## LED E BOTÃO

## INTRO



Botão para  
acionar o LED



```
#define LED_PIN D5
#define BUTTON_PIN D0

void setup()
{
  pinMode(LED_PIN, OUTPUT);
  /* TODO: Configure o pino do botão como uma entrada. */
}

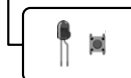
void loop()
{
  bool buttonState = digitalRead(BUTTON_PIN);

  digitalWrite(LED_PIN, buttonState);

  delay(10);
}
```

## LED E BOTÃO

## INTRO



Botão para  
acionar o LED



```
#define LED_PIN D5
#define BUTTON_PIN D0

void setup()
{
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
}

void loop()
{
  bool buttonState = digitalRead(BUTTON_PIN);

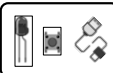
  digitalWrite(LED_PIN, buttonState);

  delay(10);
}
```

## PORTA SERIAL

### INTRO

Comunicação serial com o PC



```

#define LED_PIN D5
#define BUTTON_PIN D0

void setup()
{
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);

  Serial.begin(9600);

  Serial.println("DBServer | DBLab | Procergs");
  Serial.println("Oficina prática de Internet das Coisas");
}

void loop()
{
  bool buttonState = digitalRead(BUTTON_PIN);
  digitalWrite(LED_PIN, buttonState);

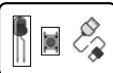
  /* TODO: Adicione mensagens pela porta serial
   * para indicar o estado do botão e LED. */
  delay(100);
}

```

## PORTA SERIAL

### INTRO

Comunicação serial com o PC



```

#define LED_PIN D5
#define BUTTON_PIN D0

void setup()
{
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);

  Serial.begin(9600);

  Serial.println("DBServer | DBLab | Procergs");
  Serial.println("Oficina prática de Internet das Coisas");
}

void loop()
{
  bool buttonState = digitalRead(BUTTON_PIN);
  digitalWrite(LED_PIN, buttonState);

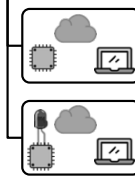
  if(buttonState == HIGH)
    Serial.println("O LED está ligado.");
  else
    Serial.println("O LED está desligado.");

  delay(100);
}

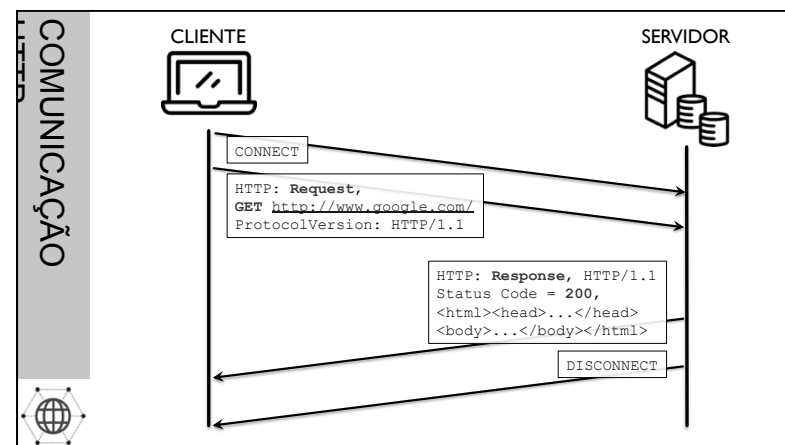
```

## COMUNICAÇÃO


### HTTP



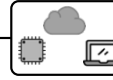
- Serviço HTTP
- Conectividade Wi-Fi
- Controle de um LED pela Internet



**CLIENTE WI-FI**



HTTP



Serviço HTTP  
via Wi-Fi

```

#include <ESP8266WebServer.h>
#include "wifi.h"

ESP8266WebServer server(80);

Wfclass wifi;

void setup(void)
{
  Serial.begin(9600);

  wifi.setup();

  server.on("/", handleRoot);


  server.begin();
}

void loop(void)
{
  server.handleClient();
}

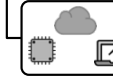
void handleRoot()
{
  server.send(200, "text/plain", "Ola Procegrs! :)");
}

```

**CLIENTE WI-FI**



HTTP



Serviço HTTP  
via Wi-Fi

```

#include <ESP8266WiFi.h>
#include <WiFiClient.h>


class Wfclass
{
public:
  Wfclass();
  void setup();

private:
  /* TODO: Informe aqui o nome e senha da rede
   * Wi-Fi disponível. */
  const char* c_ssid = "";
  const char* c_pass = "";

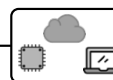
  int status;
};

```

**CLIENTE WI-FI**



HTTP



Serviço HTTP  
via Wi-Fi

```

#include <ESP8266WiFi.h>
#include <WiFiClient.h>


class Wfclass
{
public:
  Wfclass();
  void setup();

private:
  const char* c_ssid = "DBDevices";
  const char* c_pass = "!mP@db@dM";


  int status;
};

```

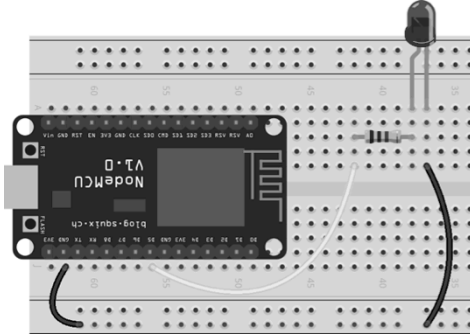
**WEB SERVER E LED**



HTTP



Controle de um  
LED pela web



**WEB SERVER E LED**

HTTP

Controle de um LED pela web

```
#define LED_PIN D5

ESP8266WebServer server(80);
WClass wifi;

void setup()
{
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);
  Serial.begin(9600);

  wifi.setup();

  server.on("/", handleRoot); /* Rota raiz. */
  /* TODO: adicione callbacks para as rotas '/on' e '/off'. */
  server.begin();
}

void handleRoot()
{
  String webString =
    "<!DOCTYPE HTML><html><head>"
    "<meta name='apple-mobile-web-app-capable' content='yes'>"
    "<style>body{background-color:#4285F4;font-size:60px;font-famil
```

**WEB SERVER E LED**

HTTP

Controle de um LED pela web

```
#define LED_PIN D5

ESP8266WebServer server(80);
WClass wifi;

void setup()
{
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);
  Serial.begin(9600);

  wifi.setup();

  server.on("/", handleRoot);
  server.on("/on", handleLedOn);
  server.on("/off", handleLedOff);

  server.begin();
  Serial.println("Servidor HTTP rodando...");
}

void handleRoot()
{
  String webString =
    "<!DOCTYPE HTML><html><head>"
    "<meta name='apple-mobile-web-app-capable' content='yes'>"
```

**WEB SERVER E LED**

HTTP

Controle de um LED pela web

```
void handleLedOn()
{
  digitalWrite(LED_PIN, HIGH);
  handleRoot();
}

void handleLedOff()
{
  digitalWrite(LED_PIN, LOW);
  handleRoot();
}
```

**BOTÃO E LED PELA WEB**

HTTP

Botão controlando LED pela web



**BOTÃO E LED PELA WEB**

HTTP

```
void setup()
{
  Serial.begin(9600);

  pinMode(BUTTON_PIN, INPUT);
  digitalWrite(BUTTON_PIN, LOW);

  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);

  wifi.setup();

  webservice.setHttpHandler("/led", httpHandler);
  webservice.start();
}
```

*Botão controlando LED pela web*

**BOTÃO E LED PELA WEB**

HTTP

```
void loop()
{
  webservice.handleClient();

  int buttonState = digitalRead(BUTTON_PIN);
  if(buttonState != currentButtonState)
  {
    ...

    currentButtonState = buttonState;

    String state;
    if(buttonState == HIGH) state = "on";
    else state = "off";

    webserviceClient.sendRequest(state, host);
  }
}
```

*Botão controlando LED pela web*

**BOTÃO E LED PELA WEB**

HTTP

```
void httpHandler()
{
  String state;

  webservice.httpHandler(state);

  if(state == "on")
    digitalWrite(LED_PIN, HIGH);
  else if(state == "off")
    digitalWrite(LED_PIN, LOW);

  Serial.println("Led está agora " + state);
  Serial.println();
}
```

*Botão controlando LED pela web*

**BOTÃO E LED PELA WEB**

HTTP

```
void ESP8266WebServiceClient::sendRequest(String &state,
                                           const char *host)
{
  wifiClient.connect(host, port);

  Serial.print("Enviando requisição para o host ");
  Serial.println(host);

  // Cria-se URI para requisição.
  String url = "/led?state=" + state;

  Serial.print("GET URL: ");
  Serial.println(url);

  wifiClient.print(String("GET ") + url + " HTTP/1.1\r\n" +
                  "Host: " + host + "\r\n" +
                  "Connection: close\r\n\r\n");

  wifiClient.print(String("GET ") + url + " HTTP/1.1\r\n" +
                  "Host: " + host + "\r\n" +
                  "Connection: close\r\n\r\n");
}
```

*Botão controlando LED pela web*

**BOTÃO E LED PELA WEB**

HTTP

```

#define LED_PIN D5
#define BUTTON_PIN D0

/* TODO: Coloque aqui o endereço IP do host contendo
 * o LED que queremos acionar. */
const char *host = "";

bool currentButtonState = LOW;

...

void setup()
{
  Serial.begin(9600);

  pinMode(BUTTON_PIN, INPUT);

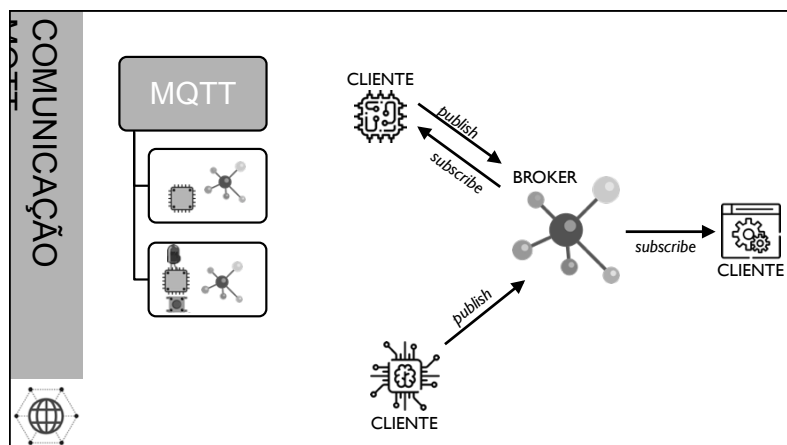
```

Botão controlando LED pela web

**MQTT**

MQTT

- Arquitetura *publish/subscribe*
- Publicação de mensagens em um *broker* público



**MQTT**

MQTT

Cliente MQTT simples

```

WClass wifi;
ESP8266PubSubClient mqttClient;
int value = 0;
long lastTimeMsg = 0;

void setup()
{
  Serial.begin(9600);
  wifi.setup();
  mqttClient.setup();
  mqttClient.connect();
}

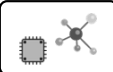
void loop()
{
  if(!mqttClient.connected()) mqttClient.connect();
  mqttClient.loop();

  long now = millis();
  if (now - lastTimeMsg > 2000) {
    lastTimeMsg = now;
    ++value;
    String message("Hello World! ");
    message += String(value, DEC);
    mqttClient.publish(MQTT_OUT_TOPIC, message.c_str());
    Serial.println("Mensagem publicada: " + message);
  }
}

```

MQTT

MQTT

  
Cliente MQTT  
simples

```


ESP8266PubSubClient::ESP8266PubSubClient()
{
    pubSubClient = new PubSubClient(wifiClient);
    byte mac[6];
    WiFi.macAddress(mac);
    deviceID = String(mac[0],HEX)+String(mac[1],HEX)+String(mac[2],HEX)+
                String(mac[3],HEX)+String(mac[4],HEX)+String(mac[5],HEX);
}

void ESP8266PubSubClient::setup()
{
    pubSubClient->setServer(mqtt_server, mqtt_server_port);
}

void ESP8266PubSubClient::setCallback(MQTT_CALLBACK_SIGNATURE)
{
    pubSubClient->setCallback(callback);
}

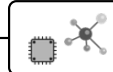
boolean ESP8266PubSubClient::publish(const char* topic,
                                     const char* payload)
{
    return pubSubClient->publish(topic, payload);
}

```



MQTT


MQTT

  
Cliente MQTT  
simples

```

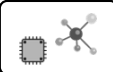
void ESP8266PubSubClient::connect()
{
    // Repete até ter uma conexão efetuada.
    while (!pubSubClient->connected()) {
        Serial.print("Conectando-se ao broker MQTT...");
        // Tenta conexão.
        if (pubSubClient->connect(deviceID.c_str()) {
            Serial.println(" conectado!");
        }
        /* TODO: quando o cliente conseguir se conectar, enviar
        * a mensagem "ESTOU VIVO" para o tópico de saída. */
    } else {
        Serial.print(" falhou, rc= ");
        Serial.print(pubSubClient->state());
        Serial.println(" Tentando novamente em 5s...");
        delay(5000);
    }
}
}

```



MQTT

MQTT


  
Cliente MQTT  
simples

```

void ESP8266PubSubClient::connect()
{
    // Repete até ter uma conexão efetuada.
    while (!pubSubClient->connected()) {
        Serial.print("Conectando-se ao broker MQTT...");
        // Tenta conexão.
        if (pubSubClient->connect(deviceID.c_str()) {
            Serial.println(" conectado!");


            pubSubClient->publish(MQTT_OUT_TOPIC, "ESTOU VIVO!");
        }
        else {
            Serial.print(" falhou, rc= ");
            Serial.print(pubSubClient->state());
            Serial.println(" Tentando novamente em 5s...");
            delay(5000);
        }
    }
}
}

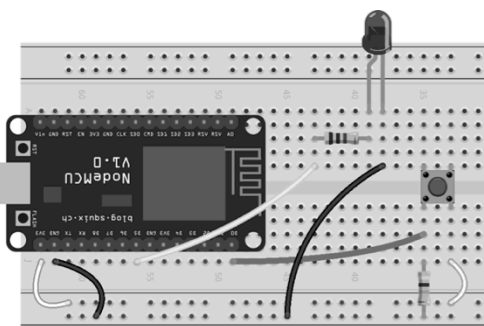
```




MQTT


MQTT

  
Cliente MQTT  
pub/sub

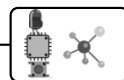




**MQTT**



MQTT



*Cliente MQTT  
pub/sub*

```

#define LED_PIN D5
#define BUTTON_PIN D0

WiFiClass wifi;
ESP8266PubSubClient mqttClient;

int currentButtonState = LOW;

void setup()
{
  Serial.begin(9600);

  pinMode(BUTTON_PIN, INPUT);
  digitalWrite(BUTTON_PIN, LOW);


  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);

  wifi.setup();
  mqttClient.setup();
  mqttClient.connect();

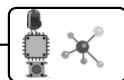
  mqttClient.setCallback(callback);
}

```

**MQTT**



MQTT



*Cliente MQTT  
pub/sub*

```


void callback(char *topic, byte *payload, unsigned int length)
{
  char message[length+1];
  memcpy(message, payload, length);
  message[length]=0;

  Serial.print("Mensagem recebida: ");
  Serial.println(message);

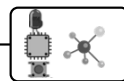
  if(!strcmp(message, "on", 2))
  {
    digitalWrite(LED_PIN, HIGH);
    Serial.println("O LED está ligado!");
  }
  else if(!strcmp(message, "off", 3))
  {
    digitalWrite(LED_PIN, LOW);
    Serial.println("O LED está desligado!");
  }
}

```

**MQTT**



MQTT



*Cliente MQTT  
pub/sub*

```

void loop()
{
  if(!mqttClient.connected()) mqttClient.connect();
  mqttClient.loop();

  int buttonState = digitalRead(BUTTON_PIN);
  if(buttonState != currentButtonState) {
    delay(50);
    buttonState = digitalRead(BUTTON_PIN);
    if(buttonState != currentButtonState)
    {
      currentButtonState = buttonState;

      String message("Meu botão está ");
      if(buttonState == LOW)
        message += String("desligado.");
      else
        message += String("ligado.");


      mqttClient.publish(MQTT_OUT_TOPIC, message.c_str());

      /* TODO: publique o estado do botão no tópico de entrada,
       * com mensagens "on" e "off". */

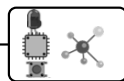
      Serial.println("Mensagem publicada: " + message);
    }
  }
  delay(10);
}

```

**MQTT**



MQTT



*Cliente MQTT  
pub/sub*

```

void loop()
{
  if(!mqttClient.connected()) mqttClient.connect();
  mqttClient.loop();

  int buttonState = digitalRead(BUTTON_PIN);
  if(buttonState != currentButtonState) {
    delay(50);
    buttonState = digitalRead(BUTTON_PIN);
    if(buttonState != currentButtonState)
    {
      currentButtonState = buttonState;

      String message("Meu botão está ");
      if(buttonState == LOW)
        message += String("desligado.");
      else
        message += String("ligado.");

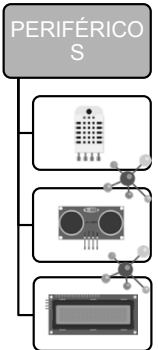
      mqttClient.publish(MQTT_OUT_TOPIC, message.c_str());

      if(buttonState == LOW) mqttClient.publish(MQTT_IN_TOPIC, "off");
      else mqttClient.publish(MQTT_IN_TOPIC, "on");


      Serial.println("Mensagem publicada: " + message);
    }
  }
  delay(10);
}

```

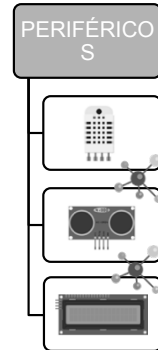
**MQTT**



- Integração com periféricos de entrada e saída
- Comandos e mensagens via MQTT



**MQTT**




**SENSORES:**

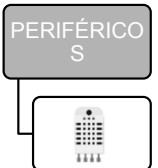
```
#define MQTT_OUT_TOPIC "dmlab/hands-on/mqtt/display"
mqttClient.publish(MQTT_OUT_TOPIC, msg);
```

**DISPLAY:**

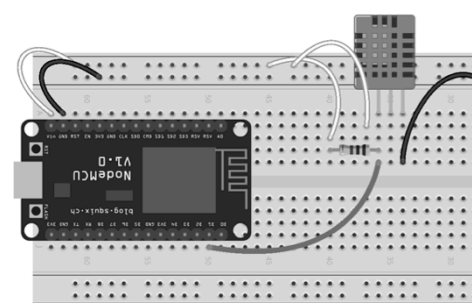

```
#define MQTT_IN_TOPIC "dmlab/hands-on/mqtt/display"
pubSubClient->subscribe(MQTT_IN_TOPIC);
```




**PERIFÉRICOS**



Sensor de temperatura e umidade

**PERIFÉRICOS**



Sensor de temperatura e umidade


```
#include <DHT.h>

#define DHTPIN D1
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup()
{
  Serial.begin(9600);

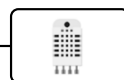
  dht.begin();

  wifi.setup();
  mqttClient.setup();
}
```



**PERIFÉRICOS**

**PERIFÉRICO S**



*Sensor de temperatura e umidade*

```

void loop()
{
  if (!mqttClient.connected()) mqttClient.connect();
  mqttClient.loop();


  float hum = dht.readHumidity();
  float temp = dht.readTemperature();
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print(" % Temp: ");
  Serial.print(temp);
  Serial.println(" Celsius");

  char humidity[MESSAGE_MAX_SIZE] = "";
  char temperature[MESSAGE_MAX_SIZE] = "";

  snprintf (temperature, MESSAGE_MAX_SIZE, "T: %02dC", (int)temp);
  snprintf (humidity, MESSAGE_MAX_SIZE, "H: %02d%%", (int)hum);

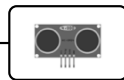
  mqttClient.publish(MQTT_OUT_TOPIC, humidity);
  mqttClient.publish(MQTT_OUT_TOPIC, temperature);
}

```

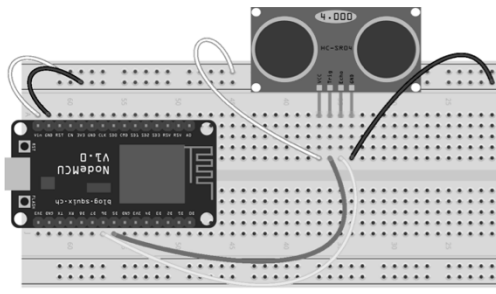



**PERIFÉRICOS**

**PERIFÉRICO S**

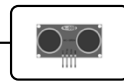


*Sensor de distância*

**PERIFÉRICOS**

**PERIFÉRICO S**



*Sensor de distância*

```

#include <hcsr04.h>

#define TRIG_PIN D5
#define ECHO_PIN D6

HCSR04 hcsr04(TRIG_PIN, ECHO_PIN, 20, 4000);


...

void loop()
{
  if (!mqttClient.connected()) mqttClient.connect();
  mqttClient.loop();

  long now = millis();
  if (now - lastTimeMsg > 500)
  {
    lastTimeMsg = now;


    int distance = hcsr04.distanceInMillimeters();
    char msg[MESSAGE_MAX_SIZE] = "";
    snprintf (msg, MESSAGE_MAX_SIZE, "Dist: %ldmm", distance);
    Serial.println(msg);
    mqttClient.publish(MQTT_OUT_TOPIC, msg);
  }
}

```

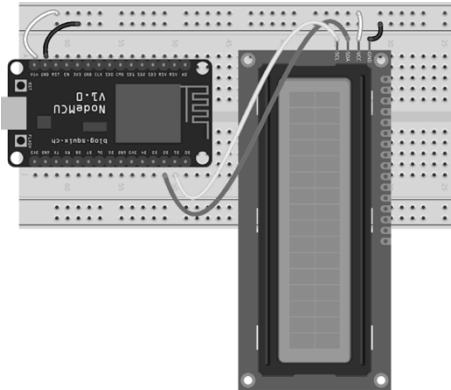



**PERIFÉRICOS**

**PERIFÉRICO S**



*Display de cristal líquido*

**PERIFÉRICOS**

PERIFÉRICO S

Display de cristal líquido

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);


void setup()
{
  Serial.begin(9600);

  wifi.setup();
  mqttClient.setup();
  mqttClient.connect();

  mqttClient.setCallback(callback);

  lcd.begin();
  lcd.backlight();
  lcd.print("DBLAB HANDS-ON");
}

void loop()
{
  if (!mqttClient.connected()) mqttClient.connect();
  mqttClient.loop();
}
```



**PERIFÉRICOS**

PERIFÉRICO S


Display de cristal líquido

```
void callback(char *topic, byte *payload, unsigned int length)
{
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");

  char msg[length+1];
  memcpy(msg, payload, length);
  msg[length]=0;
  Serial.print(msg);

  if(!strcmp(msg, "I", 1)) {
    clearLcd(strlen(msg));
    lcd.setCursor(0, 0);
  }
  else if(!strcmp(msg, "H", 1)) {
    clearLcd(strlen(msg));
    lcd.setCursor(8, 0);
  }
  else if(!strcmp(msg, "D", 1)) {
    clearLcd(strlen(msg));
    lcd.setCursor(0, 1);
  }

  lcd.printstr(msg);
}
```



<https://github.com/dbserver/dblab/tree/master/hands-on/iot>



**Obrigado!**



**dblab@dbserver.com.br**

