

## Relatório Técnico

Nº Grupo: 07

Nome dos integrantes: Bill Hebert Pereira da Rocha Choi,  
César Augusto Araujo Miguel, Dereck Murillo Baksa  
Erick Araujo Ferreira, Gustavo Ziliotto Bello

Turma: 1CCOB

Tema do projeto: Queimadas

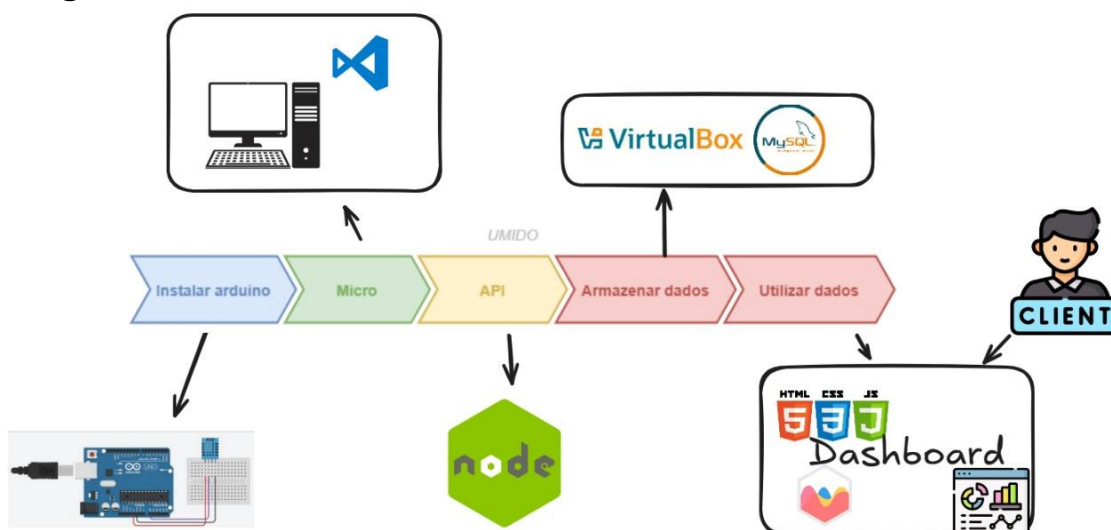
Sensor: DHT11 (Umididade)

### Introdução

Com o passar dos anos se torna mais evidente a vultuosa mudança climática e seus riscos para os grandes produtores, queimadas geram uma perda anual de aproximadamente R\$ 14 bilhões segundo a CNA, e seus números evidenciam um aumento no grau de perigo que esse problema proporciona.

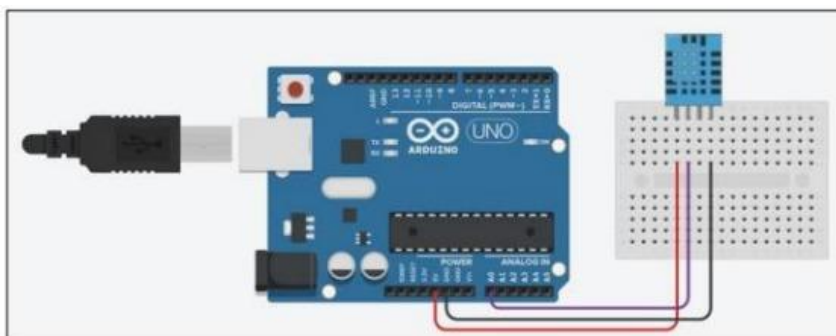
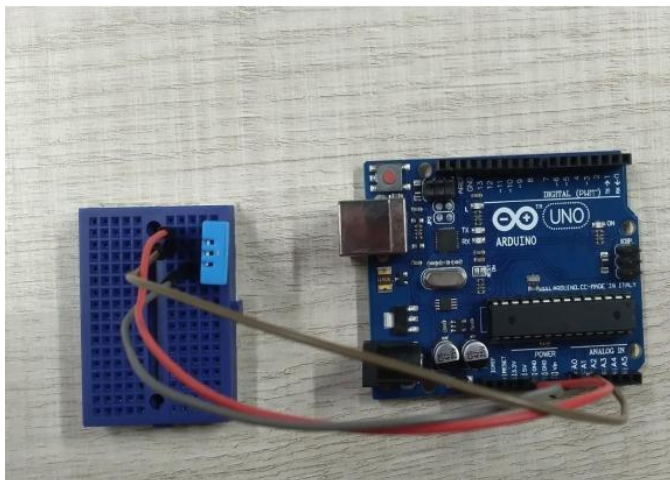
O projeto Umido surgiu como uma solução para o alto risco de queimadas em plantações e áreas agrícolas, onde haverá um monitoramento inteligente dos sensores instalados dentro de um mapa de formato “plano cartesiano” em escala 1 por 1 (um sensor é igual a 1 hectares) sendo assim capaz de monitorar a umidade de vários hectares por um longo terreno.

### Diagrama de solução



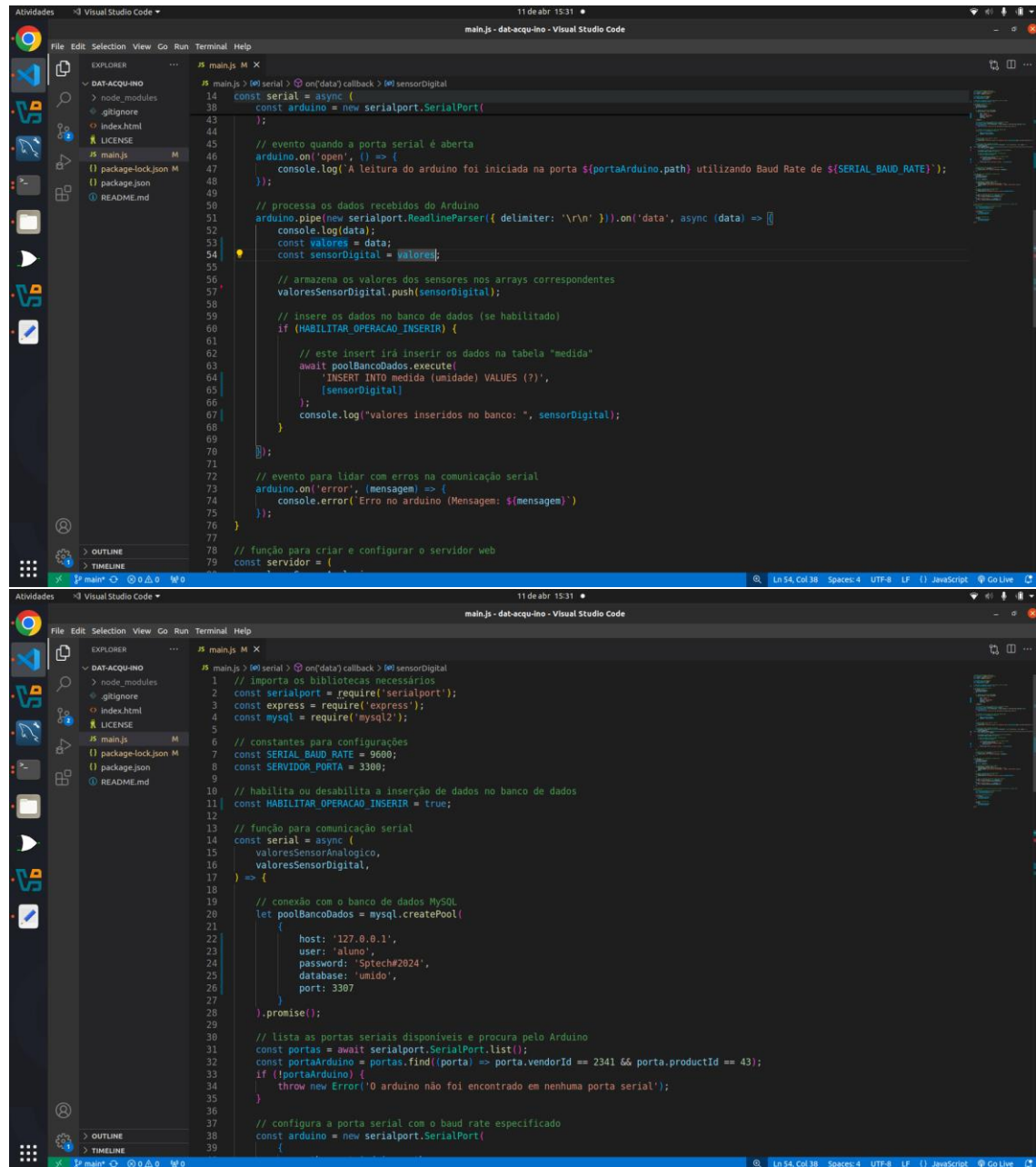
## Arquitetura de Montagem do Sensor

Segue abaixo uma foto da arquitetura do Arduino Uno R3 conectado a uma protoboard via jumpers se comunicando com o sensor DHT11:



## Arquitetura do Sistema

Após a coleta dos dados, nós utilizamos um banco de dados MySQL Sever local para armazená-los por meio de uma API Node JS (dat-acqu-ino), segue abaixo a aplicação em funcionamento:



```
main.js | 11 de abr 15:31 | Visual Studio Code  
main.js - dat-acqu-ino - Visual Studio Code  
14 const serial = async (on('data') callback) => {  
38 const arduino = new serialport.SerialPort(  
43 );  
44  
45 // evento quando a porta serial é aberta  
46 arduino.on('open', () => {  
47 console.log('A leitura do arduino foi iniciada na porta ${portaArduino.path} utilizando Baud Rate de ${SERIAL_BAUD_RATE}');  
48 });  
49  
50 // processa os dados recebidos do Arduino  
51 arduino.pipe(new serialport.ReadlineParser({ delimiter: '\r\n' })).on('data', async (data) => {  
52 console.log(data);  
53 const valores = data;  
54 const sensorDigital = valores;  
55  
56 // armazena os valores dos sensores nos arrays correspondentes  
57 valoresSensorDigital.push(sensorDigital);  
58  
59 // insere os dados no banco de dados (se habilitado)  
60 if (HABILITAR_OPERACAO_INSERIR) {  
61 // este insert irá inserir os dados na tabela "medida"  
62 await poolBancoDados.execute(  
63 'INSERT INTO medida (unidade) VALUES (?)',  
64 [sensorDigital]  
65 );  
66 console.log('valores inseridos no banco: ', sensorDigital);  
67 }  
68 }  
69 }  
70 }  
71  
72 // evento para lidar com erros na comunicação serial  
73 arduino.on('error', (mensagem) => {  
74 console.error('Erro no arduino (Mensagem: ${mensagem})');  
75 }  
76 }  
77 }  
78  
79 // função para criar e configurar o servidor web  
80 const servidor = {  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }  
101 }  
102 }  
103 }  
104 }  
105 }  
106 }  
107 }  
108 }  
109 }  
110 }  
111 }  
112 }  
113 }  
114 }  
115 }  
116 }  
117 }  
118 }  
119 }  
120 }  
121 }  
122 }  
123 }  
124 }  
125 }  
126 }  
127 }  
128 }  
129 }  
130 }  
131 }  
132 }  
133 }  
134 }  
135 }  
136 }  
137 }  
138 }  
139 }  
140 }  
141 }  
142 }  
143 }  
144 }  
145 }  
146 }  
147 }  
148 }  
149 }  
150 }  
151 }  
152 }  
153 }  
154 }  
155 }  
156 }  
157 }  
158 }  
159 }  
160 }  
161 }  
162 }  
163 }  
164 }  
165 }  
166 }  
167 }  
168 }  
169 }  
170 }  
171 }  
172 }  
173 }  
174 }  
175 }  
176 }  
177 }  
178 }  
179 }  
180 }  
181 }  
182 }  
183 }  
184 }  
185 }  
186 }  
187 }  
188 }  
189 }  
190 }  
191 }  
192 }  
193 }  
194 }  
195 }  
196 }  
197 }  
198 }  
199 }  
200 }  
201 }  
202 }  
203 }  
204 }  
205 }  
206 }  
207 }  
208 }  
209 }  
210 }  
211 }  
212 }  
213 }  
214 }  
215 }  
216 }  
217 }  
218 }  
219 }  
220 }  
221 }  
222 }  
223 }  
224 }  
225 }  
226 }  
227 }  
228 }  
229 }  
230 }  
231 }  
232 }  
233 }  
234 }  
235 }  
236 }  
237 }  
238 }  
239 }  
240 }  
241 }  
242 }  
243 }  
244 }  
245 }  
246 }  
247 }  
248 }  
249 }  
250 }  
251 }  
252 }  
253 }  
254 }  
255 }  
256 }  
257 }  
258 }  
259 }  
260 }  
261 }  
262 }  
263 }  
264 }  
265 }  
266 }  
267 }  
268 }  
269 }  
270 }  
271 }  
272 }  
273 }  
274 }  
275 }  
276 }  
277 }  
278 }  
279 }  
280 }  
281 }  
282 }  
283 }  
284 }  
285 }  
286 }  
287 }  
288 }  
289 }  
290 }  
291 }  
292 }  
293 }  
294 }  
295 }  
296 }  
297 }  
298 }  
299 }  
300 }  
301 }  
302 }  
303 }  
304 }  
305 }  
306 }  
307 }  
308 }  
309 }  
310 }  
311 }  
312 }  
313 }  
314 }  
315 }  
316 }  
317 }  
318 }  
319 }  
320 }  
321 }  
322 }  
323 }  
324 }  
325 }  
326 }  
327 }  
328 }  
329 }  
330 }  
331 }  
332 }  
333 }  
334 }  
335 }  
336 }  
337 }  
338 }  
339 }  
340 }  
341 }  
342 }  
343 }  
344 }  
345 }  
346 }  
347 }  
348 }  
349 }  
350 }  
351 }  
352 }  
353 }  
354 }  
355 }  
356 }  
357 }  
358 }  
359 }  
360 }  
361 }  
362 }  
363 }  
364 }  
365 }  
366 }  
367 }  
368 }  
369 }  
370 }  
371 }  
372 }  
373 }  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }
```

The screenshot shows a terminal window with two panes. The left pane displays the output of an API running on an Arduino. It shows a series of messages: 'API executada com sucesso na porta 3300', 'A leitura do arduino foi iniciada na porta /dev/ttyACM0 utilizando Baud Rate de 9600', and then a repeating sequence of 'valores inseridos no banco: 58.00' followed by '58.00'. The right pane shows a MySQL query result for 'select \* from medidas'. The result is a table with columns 'id', 'data', and 'umidade'. It contains 20 rows of data, showing a progression of humidity values from 56.00 to 62.00 over time on 2025-04-11.

id	data	umidade
1	2025-04-11 15:27:45	56.00
2	2025-04-11 15:27:46	56.00
3	2025-04-11 15:27:47	56.00
4	2025-04-11 15:27:48	56.00
5	2025-04-11 15:27:49	56.00
6	2025-04-11 15:27:50	56.00
7	2025-04-11 15:27:51	56.00
8	2025-04-11 15:27:52	56.00
9	2025-04-11 15:27:53	56.00
10	2025-04-11 15:27:54	56.00
11	2025-04-11 15:27:55	56.00
12	2025-04-11 15:27:56	56.00
13	2025-04-11 15:27:57	57.00
14	2025-04-11 15:27:58	57.00
15	2025-04-11 15:27:59	59.00
16	2025-04-11 15:28:00	59.00
17	2025-04-11 15:28:01	60.00
18	2025-04-11 15:28:02	60.00
19	2025-04-11 15:28:03	62.00
20	2025-04-11 15:28:04	62.00

## Código do Projeto

No projeto, nós utilizamos o sensor DHT11 para capturar os dados de umidade, e utilizando as portas seriais coletamos esses dados por meio de uma API node:

```
8   DHT sensorDHT(PINO_SENSOR_DHT11, TIPO_SENSOR); // chama a função da biblioteca
9
10
11 void setup() {           //função de inicialização básica
12   Serial.begin(9600);     // definir a quantidade de dados da usb
13   sensorDHT.begin();     // iniciar sensor
14
15 }
16
17 void loop() {
18   float umidade = sensorDHT.readHumidity()           //variavel do tipo float e onde será armazenado o dado de umidade
19   if (isnan(isnan(umidade))                           // verifica se as variaveis tem dado, caso alguma delas não tenha exibe
20   {
21     Serial.println("ERRO ao ler os dados do sensor"); // mensagem de erro
22   } else {
23     Serial.print("umidade:");           // label umidade
24     Serial.print(umidade);             // exibe o dado de umidade
25     Serial.print(",");                 // separa para a proxima label
26
27     Serial.print("Max_umidade:");      // label max umidade
28     Serial.print(max_u);               // exibe o dado fixo max de umidade para determinar limite
29     Serial.print(",");                 // separa para a proxima label
30     Serial.print("Min_umidade:");      // label min umidade
31     Serial.println(min_u);             // exibe o dado fixo min de umidade para determinar limite
32
33   delay(1000); // espera 1 segundo para finalizar
34 }
```

## Resultados Iniciais

Com bases nos primeiros testes que efetuamos em sala de aula, foi possível realizar a captura desses dados do Arduino utilizando uma API escrita em Node JS e logo em segundas os armazenamos em um banco de dados MySql Server, sendo possível visualizá-los através de queries.