

LÚCIO MÁRIO ALMEIDA LOPES

**ELETRONTEST - DESENVOLVIMENTO DE KIT EDUCACIONAL,
POR TABULEIRO DE INSERÇÃO DE COMPONENTES COM
VERIFICAÇÃO E VALIDAÇÃO DA ATIVIDADE NO SISTEMA DE
GERENCIAMENTO DE APRENDIZAGEM**

Betim - MG

Dezembro/2022

LÚCIO MÁRIO ALMEIDA LOPES

**ELETRONTEST - DESENVOLVIMENTO DE KIT EDUCACIONAL, POR
TABULEIRO DE INSERÇÃO DE COMPONENTES COM VERIFICAÇÃO E
VALIDAÇÃO DA ATIVIDADE NO SISTEMA DE GERENCIAMENTO DE
APRENDIZAGEM**

Trabalho de conclusão de curso apresentado à Coordenadoria do Curso de Engenharia de Controle e Automação do Campus Betim do Instituto Federal de Minas Gerais para obtenção do diploma de Bacharel em Engenharia de Controle e Automação.

Orientador: Helbert Ribeiro de Sá

Coorientador: Virgil Del Duca Almeida

Dezembro/2022

FICHA CATALOGRÁFICA

L864e Lopes, Lúcio Mário Almeida
Eletrontest - desenvolvimento de kit educacional, por tabuleiro de inserção de componentes com verificação e validação da atividade no sistema de gerenciamento de aprendizagem / Lúcio Mário Almeida Lopes. – 2022.
77 f. : il.

Trabalho de conclusão de curso (Bacharelado em Engenharia de Controle e Automação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Câmpus Betim, 2022.

Orientação: prof. Me. Helbert Ribeiro de Sá
Coorientação: prof. Me. Vilgil Del Duca Almeida

1. Ensino à distância. 2. SCORM. 3. Objetos de aprendizagem interativos. 4. ESP32. I. Lopes, Lúcio Mário Almeida. II. Título.

CDU: 681.3:37.09

LÚCIO MÁRIO ALMEIDA LOPES

**ELETRONTEST - DESENVOLVIMENTO DE KIT EDUCACIONAL, POR
TABULEIRO DE INSERÇÃO DE COMPONENTES COM VERIFICAÇÃO E
VALIDAÇÃO DA ATIVIDADE NO SISTEMA DE GERENCIAMENTO DE
APRENDIZAGEM**

Este trabalho foi julgado adequado para obtenção do título de Bacharel em Engenharia de Controle e Automação, pelo Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Betim - MG, 08 de dezembro de 2022:



Helbert Ribeiro de Sá, Msc.
Orientador
Instituto Federal de Minas Gerais



Virgil Del Duca Almeida, Msc.
Coorientador
Instituto Federal

Documento assinado digitalmente
gov.br REGINALDO VAGNER FERREIRA
Data: 14/12/2022 13:07:40-0300
Verifique em <https://verificador.iti.br>

Reginaldo Vagner Ferreira, Dr.
Instituto Federal de Minas Gerais

Documento assinado digitalmente
gov.br VICTOR ALVES SILVA E MELO
Data: 14/12/2022 16:52:07-0300
Verifique em <https://verificador.iti.br>

Victor Alves Silva e Melo, Msc
Instituto Federal de Minas Gerais

*Este trabalho é dedicado à minha família e
a todas as famílias, sem as quais perderíamos
nossos referenciais de humanidade.*

AGRADECIMENTOS

Os agradecimentos principais são direcionados aos professores do Instituto Federal de Minas Gerais, Hebert Ribeiro de Sá, Virgil Del Duca Almeida, Leandro Freitas e Reginaldo Ferreira pelo imenso apoio e orientação na confecção deste trabalho, e na pessoa dos quais agradeço a todo o corpo docente do Campus Betim IFMG por sua sempre pronta atenção.

Agradecimentos também são necessários à toda a equipe administrativa do Campus Betim, principalmente aos setores da Biblioteca e Secretaria que sempre prontamente atendem as solicitações dos alunos.

Agradecimentos especiais são direcionados à minha esposa que sempre me motiva, apoia e revisa meus agradecimentos. Também extensivo à toda a minha família que compreensivamente se privou do meu convívio durante o período deste curso.

Agradecimento contínuo e reiterado à Deus que nos proporciona a vida, a capacidade para buscar o conhecimento e principalmente coloca em nosso caminho as pessoas importantes para nosso crescimento.

*“Se clamares por conhecimento, e por inteligência
alçares a tua voz, se como a prata a buscares e como
a tesouros escondidos a procurares, então entenderás
o temor do Senhor, e acharás o conhecimento de Deus.”
(Bíblia Sagrada, Provérbios 2, 3-5)*

RESUMO

Este trabalho consiste no projeto e execução de um kit Educacional por tabuleiro de inserção de componentes com verificação e validação da atividade prática para auxílio ao ensino à distância de disciplinas laboratoriais técnicas da área de eletricidade e eletrônica. O objetivo deste kit é fornecer ao docente a capacidade de avaliar experiências práticas guiadas podendo ser aplicada em um laboratório local ou à distância quando o discente comunica-se pela internet. O equipamento elaborado possui um custo baixo de produção e pode ser facilmente confeccionado possibilitando ao docente a condução de aulas com mais rendimento e preparação. O uso do Microcontrolador ESP32 da Espressif® na confecção do kit possibilitou a utilização dos recursos de comunicação sem fio e leitura analógica que o mesmo possui.

Palavras-chave: Ensino a distância. ESP32. Scorm.

ABSTRACT

This work consists of the design and execution of an Educational kit. The kit consist in a table to inserting components to do the lessons. The kit also does verification and validation of the practical activity in the LMS. It improve the ability to the distance learning of laboratory subjects in the area of electricity and electronics. The objective of this kit is to provide the teacher with the ability to evaluate guided practical experiences that can be applied in a local laboratory or in the distance when the student communicates over the Internet. The implementation of this kit has a low production cost and can be easily made. And it allows the teacher to conduct classes with more income and preparation. The use of Esspressif's ESP32 Microcontroller® in the making of the kit made it possible by using the wirelles communication and the analog reading resources that it has.

Keywords: Distance learning. ESP32. Scorm.

LISTA DE ILUSTRAÇÕES

Figura 1 – Foto mostrando aulas radio-difundidas em Chicago 1937	17
Figura 2 – Montagem em Proto-board	18
Figura 3 – Montagem em KIT DIY	18
Figura 4 – Kit Didático De Lorenzo	19
Figura 5 – Tabuleiro do Kit De Lorenzo	19
Figura 6 – Módulo de Componentes do Kit De Lorenzo	20
Figura 7 – Comparação do Uso e Não uso do Kit.	21
Figura 8 – Equipamento Usando o Princípio da Cama de Pregos	22
Figura 9 – Esquema de Pinos do DevKitV1 da Doit	24
Figura 10 – Dimensões do DEVKIT-V1	25
Figura 11 – Arquivo em HTML	25
Figura 12 – Visão no Navegador	26
Figura 13 – Estrutura do JSON	31
Figura 14 – Maleta para Acondicionar o Kit	33
Figura 15 – Maleta em MDF para Acondicionar o Kit	33
Figura 16 – Acoplamento por Amplificadores Operacionais	34
Figura 17 – Placa Principal do PROGRAMA	35
Figura 18 – Camadas da Placa CPU	36
Figura 19 – Vista Superior da Placa CPU	36
Figura 20 – Vista Inferior da Placa CPU	37
Figura 21 – Perspectiva da Placa CPU	37
Figura 22 – Vista Superior da Placa de Tabuleiro	38
Figura 23 – Vista Inferior da Placa de Tabuleiro	38
Figura 24 – Visão das Camadas da Placa de Tabuleiro	38
Figura 25 – Vista Superior da Placa de Resistores	39
Figura 26 – Vista Inferior da Placa de Resistores	39
Figura 27 – Tabuleiro de Inserção e Componentes	40
Figura 28 – Placa Principal Montada	40
Figura 29 – Conexões entre as Partes do KIT	41
Figura 30 – Kit Montado na Maleta	41
Figura 31 – Ordem de Serviço - CPU	42
Figura 32 – Formato do JSON liberado pelo Programa	46
Figura 33 – Funcionamento do Kit na Bancada	53
Figura 34 – Telas Iniciais	53
Figura 35 – Telas de Configuração	54
Figura 36 – Telas de Medição de Resistores	54
Figura 37 – Telas de Montagem e Medição de Tensão	54
Figura 38 – Telas Finais da Prática	55
Figura 39 – Validação da Atividade no Sistema de Aprendizagem	55
Figura 40 – Relatório do LMS	56
Figura 41 – Características ESP32	63
Figura 42 – Baixando a Interface do Arduino	65
Figura 43 – Executando o arquivo	65

Figura 44 – Instalação Padrão do Arduino	66
Figura 45 – Configurando o repositório de Placas	66
Figura 46 – Visualizando as Placas do ESP32	67
Figura 47 – Buscando as Placas do ESP32	68
Figura 48 – Baixando o driver CP2102	68
Figura 49 – Executando a Instalação do Driver	69
Figura 50 – Painel de Projetos no ISEAZY	74
Figura 51 – Menu do Projeto no ISEAZY	75
Figura 52 – Menu de Distribuição no ISEAZY	76
Figura 53 – Menu de Publicação no ISEAZY	76
Figura 54 – Tela de Download no ISEAZY	77

LISTA DE TABELAS

Tabela 1 – Tabela Comparativa dos Kits Didáticos	20
Tabela 2 – Tabela Comparativa das Caixas de Acondicionamento	33
Tabela 3 – Tabela de Registros de Despacho da Mercadoria	42
Tabela 4 – Tabela de Custos do Projeto	43
Tabela 5 – Tabela de Comparativa de Custos	43
Tabela 6 – Notas de Lançamento do ESP32	64

LISTA DE CÓDIGOS

Código 2.1 – Exemplo de SCORM	29
Código 2.2 – Declaração de Organizações	30
Código 2.3 – Especificação do ID	30
Código 2.4 – Declaração de Recursos	31
Código 2.5 – Especificação do Recurso	31
Código 5.1 – looping.ino - Looping principal do programa	45
Código 5.2 – looping.ino - Looping principal do programa	45
Código 5.3 – looping.ino - Looping principal do programa	45
Código 6.1 – Cama de Pregos	47
Código 6.2 – Código de Encriptação	51
Código C.1 – TCC.ino - Arquivo descritivo	70
Código C.2 – Bibliotecas.ino - Bibliotecas do programa	70
Código C.3 – Objetos.ino - Declaração de Objetos do programa	70
Código C.4 – Pinos.ino - Configuração dos Pinos usados no programa	70
Código C.5 – Setup.ino - Setup do programa	71
Código C.6 – looping.ino - Looping principal do programa	71

LISTA DE ABREVIATURAS E SIGLAS

HTML " <i>Hyper Text Markup Language</i> " - Língua de Marcação de Hipertexto	23
SCORM " <i>Sharable Content Object Reference Model</i> " - Conteúdo Compartilhável do Modelo de Referência a Objeto	23
WYSIWYG " <i>What You See Is What You Get</i> " - O que você vê é o que você tem	26
API " <i>Application Programming Interface</i> " - Interface de programação de aplicações	27
eLearning Educação pela Internet	26
LMS " <i>Learning Management System</i> " - Sistema de Gestão de Aprendizagem	23
DIY " <i>Do It Yourself</i> " - Faça Você Mesmo	18
JSON " <i>Javascript Object Notation</i> " - Notação de Objeto em Javascript	31
SCO " <i>Sharable Content Object</i> " - Objeto de Conteúdo Compartilhado	28
XML " <i>eXtensive Markup Language</i> " - Língua extensiva de Marcação de Objeto	29
SAR " <i>Successive Approximation Register</i> " - Registro de aproximação sucessiva	46

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Kits Educacionais Utilizados em Aulas Práticas Atualmente.	17
1.1.1	Proto-Board	18
1.1.2	Kits de Montagem em PCI	18
1.1.3	Kits por Tabuleiro de Inserção de Componentes	19
1.1.4	Comparativo entre os Tipos de Kits Didáticos	20
1.2	Necessidade de Criação de Equipamento.	21
2	REVISÃO BIBLIOGRÁFICA	22
2.1	Cama de Pregos	22
2.2	ESP32 e DEVKIT-V1	23
2.3	Ferramentas para a programação do ESP32	24
2.4	HTML	25
2.5	SCORM	26
2.6	Visão Geral do SCORM	26
2.6.1	Sistema de Gestão de Aprendizagem	27
2.6.2	SCORM API	27
2.6.3	Sistema de Gerenciamento de Conteúdo de Aprendizagem.	27
2.6.4	Objeto de Conteúdo Sharable (SCO)	28
2.6.5	Recursos	28
2.6.6	Modelos de Dados	28
2.6.7	Nomeando Convenções e Documentos De Controle	28
2.6.8	Pacote de conteúdo	28
2.6.9	Manifesto	28
2.6.10	Versões de SCORM	29
2.6.11	Elementos e Variações Obrigatórias no SCORM	29
2.6.12	Entendendo o Manifesto SCORM	29
2.6.13	Elementos MANIFEST_ID	30
2.6.14	Organizações	30
2.6.15	Recursos	31
2.7	JAVASCRIPT® e JSON®	31
3	DESCRIÇÃO GERAL DO TRABALHO	32
3.1	Princípio de Funcionamento	32
3.2	Hardware do Equipamento	32
3.2.1	Caixa do Equipamento	32
4	HARDWARE DO KIT	34
4.1	Placa Principal	34
4.2	Diagrama Esquemático da Placa Principal.	34
4.2.1	Custos do Projeto	43
5	PROGRAMAÇÃO DA PLACA PRINCIPAL DO KIT	44

5.1	O Código	44
5.1.1	TCC.INO	44
5.1.2	Bibliotecas.INO	44
5.1.3	Objetos.INO	44
5.1.4	Pinos.INO	44
5.1.5	Setup.INO	44
5.1.6	Looping.INO	45
5.1.7	Resposta da Entrada Analógica do ESP32	46
6	CÓDIGOS JAVASCRIPT PARA VALIDAÇÃO DA ATIVIDADE	47
6.1	Código JavaScript da Cama de Pregos	47
6.2	Código de Encriptação do Resultado	51
7	RESULTADOS OBTIDOS	53
7.1	Telas do Arquivo Scorm Criado	53
8	CONCLUSÃO	57
8.1	Possíveis Trabalhos em Continuação Deste	57
8.2	Possibilidade de Utilização do Kit em Trabalhos de Extensão	58
	REFERÊNCIAS	59
	APÊNDICES	61
	APÊNDICE A – CARACTERÍSTICAS DO ESP32	62
A.1	ESP32 – Características	62
A.1.1	Processamento	62
A.1.2	Entradas e Saídas	62
A.1.3	Segurança	63
A.1.4	Diagrama de Blocos	63
A.1.5	Documentação do ESP32	64
	APÊNDICE B – INSTALANDO E CONFIGURANDO A INTERFACE DO ARDUINO	65
B.1	Interface do Arduino:	65
B.2	Adicionando as Placas do ESP32 na IDE do Arduino	66
B.3	Instalando o driver de comunicação	67
	APÊNDICE C – CÓDIGO DO KIT	70
	APÊNDICE D – FERRAMENTA ONLINE PARA FAZER SCORM - ISEAZY®	74

1 INTRODUÇÃO

Segundo Pasini (PASINI; CARVALHO; ALMEIDA, 2020), após a pandemia possivelmente haverá aumento da hibridismo entre educação presencial e a distancia. Este hibridismo, mistura entre os dois tipos, deve se à possibilidade factível de novas doenças coletivas no futuro, ressalta Pasini nas considerações de seu artigo.

No artigo de Pedro Prata (PRATA, 2021) no Jornal Estadão - Estado de São Paulo é verificada a autenticidade de uma foto que mostra crianças tendo aulas difundidas via rádio na cidade de Chicago em 1937. A reportagem cita ainda que a foto tornou-se comum nas redes sociais devido ao momento atual no qual vivemos uma pandemia. A foto está mostrada na Figura 1. A foto demonstra que a tecnologia disponível à época foi utilizada para vencer as restrições de isolamento impostas pelo momento ¹.



Figura 1 – Foto mostrando aulas radio-difundidas em Chicago 1937

Fonte: Blog Política do Estadão(ver nota.)

Entretanto a educação à distância apresenta desvantagens para algumas disciplinas impossibilitando a sua aplicação. Vlasenko e Bozhok (2014) ² citam em seu artigo algumas desvantagens da educação à distância e dentre estas destaca-se aqui, o acesso à tecnologia necessária e a ausência da percepção do docente quanto ao aprendizado do discente de forma imediata.

1.1 KITS EDUCACIONAIS UTILIZADOS EM AULAS PRÁTICAS ATUALMENTE.

Buscando na internet por "Ferramentas Educacionais de Eletricidade", é possível encontrar vários kits para ensino de eletrônica e eletricidade. Mas basicamente são kits de três tipos. Os kits de experimento, identificados neste trabalho como kits por tabuleiro de inserção, os kits de montagem em placas prontas e as placas para protótipo, comumente conhecidos como protoboards.

¹ <<https://politica.estadao.com.br/blogs/estadao-verifica/foto-viral-mostra-criancas-tendo-aulas-pelo-radio-por-causa-de-epidemia-de-polio>>

² <<http://dspace.nuft.edu.ua/jspui/handle/123456789/20684>>

1.1.1 PROTO-BOARD

Os Proto-Boards, ou Placas Protótipos são matrizes de contatos que são utilizadas para fazer as montagens conforme a vontade do executor da mesma. Podemos ver na Figura 2 um exemplo de proto-board extraído do site da Minipa ®(MINIPA DO BRASIL, 2022). Trata-se do Proto-board fabricado pela Minipa do Brasil®.

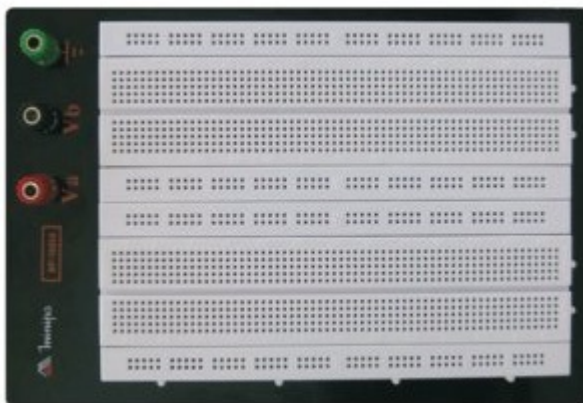


Figura 2 – Montagem em Proto-board

Fonte: Sítio eletrônico <<https://minipa.com.br/component/djcatalog2/items?search=protoboard>>

1.1.2 KITS DE MONTAGEM EM PCI

Montagens em placas de circuito impresso prontas são os chamados kits do tipo **DIY**³. Este recurso de aprendizagem de circuitos elétricos e eletrônicos consiste em um conjunto de componentes e uma placa pronta para o discente executar a montagem e fixar os conhecimentos aos quais esta se refere. Um exemplo de Kit **DIY** está mostrado na Figura 3, trata-se da montagem de um circuito VU/Bargraph comercializado pela Casa dos Resistores ⁴ (CASA DO RESISTOR, 2022). Com este kit o discente monta uma prática de um VU de LEDs que é um circuito que mede os níveis de som e mostra através de LEDs estes níveis e o docente avalia a correção de sua montagem e seu funcionamento.

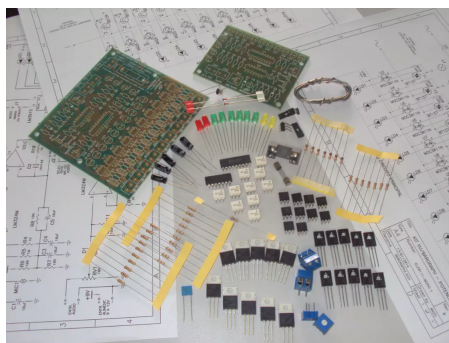


Figura 3 – Montagem em KIT DIY

Fonte: Sítio eletrônico <<https://www.casadoresistor.com.br/kit-vu-bargraph-led-elz0110-p4641>>

³ "Do It Yourself" - Faça Você Mesmo (DIY)

⁴ <<https://www.casadoresistor.com.br/kit-vu-bargraph-led-elz0110-p4641>>

1.1.3 KITS POR TABULEIRO DE INSERÇÃO DE COMPONENTES

Este nome "*Tabuleiro de Inserção de Componentes*" não é comumente utilizado e está identificado assim para melhor identificação do tipo de recurso didático. Um exemplo deste tipo de kit são os Kits de aprendizagem da empresa De Lorenzo®⁵. A Figura 4 mostra a fotografia retirada do sitio da De Lorenzo®(DELORENZO DO BRASIL, 2022) que mostra o Kit DL 2150 que é um exemplo de equipamento para esta finalidade.



Figura 4 – Kit Didático De Lorenzo

Fonte: Sítio eletrônico <<https://delorenzo.com.br/atuacao>>

Ainda sobre o Kit didático De Lorenzo temos a Figura 5 que mostra o tabuleiro do equipamento no qual as montagens podem ser executadas e a Figura 6 que mostra os módulos de componentes usados para as montagens.

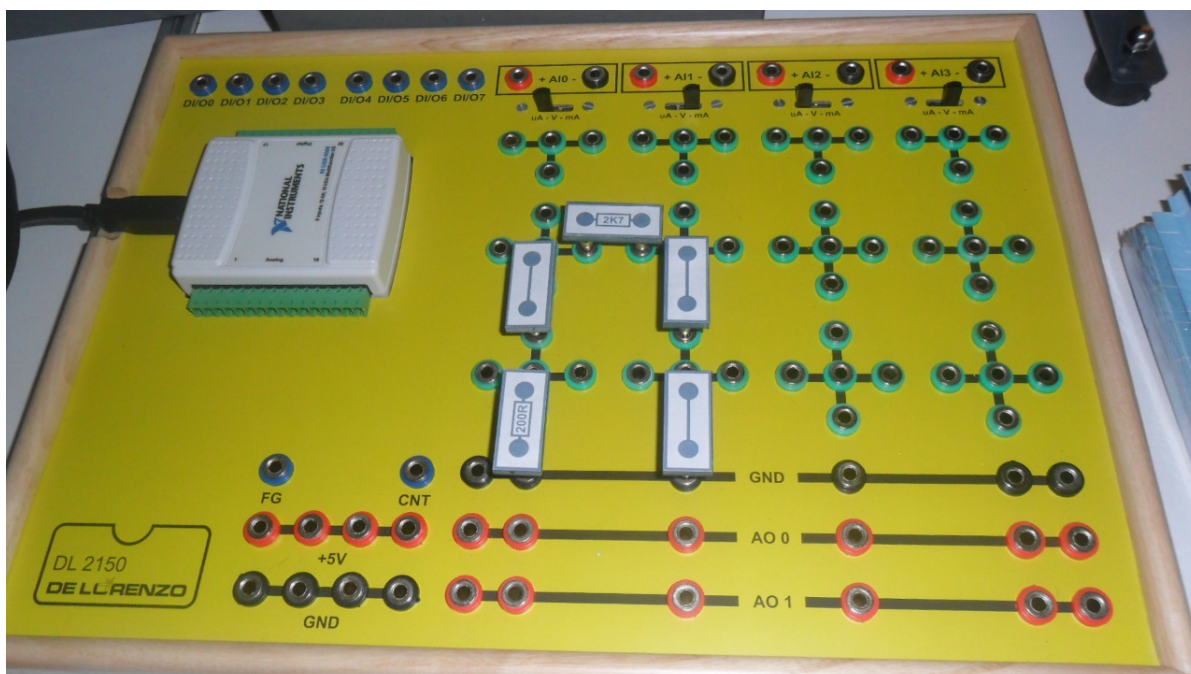


Figura 5 – Tabuleiro do Kit De Lorenzo

Fonte: Sítio eletrônico <<https://delorenzo.com.br/atuacao>>

⁵ <<https://delorenzo.com.br/atuacao>>

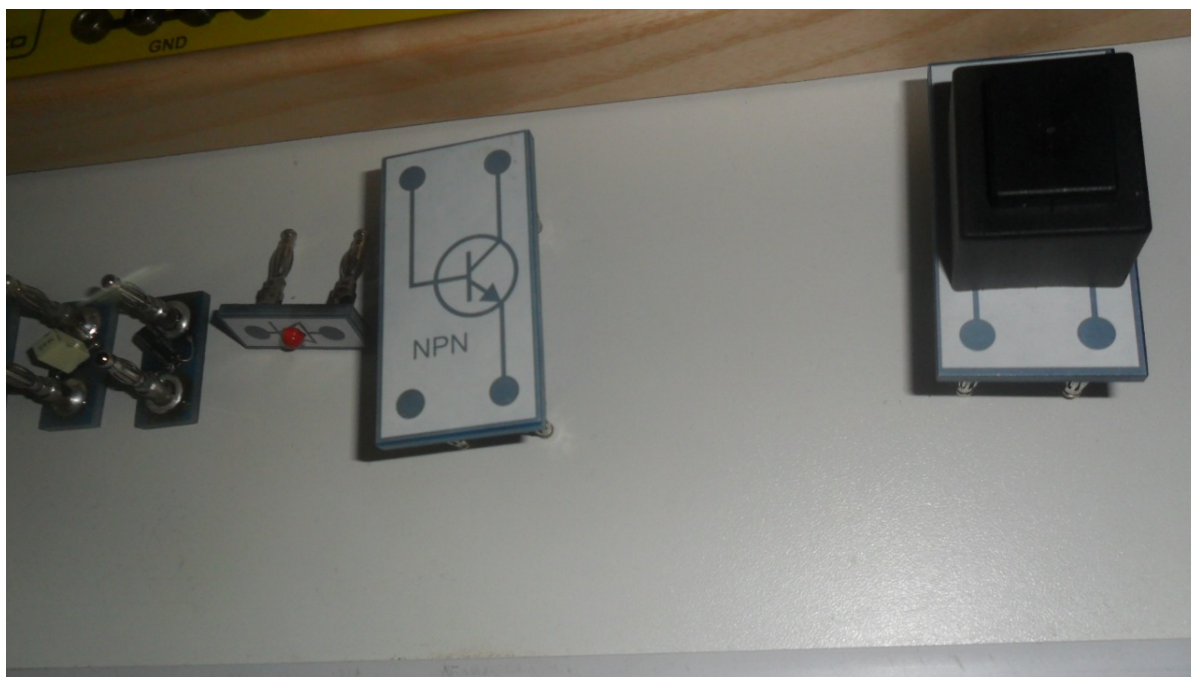


Figura 6 – Módulo de Componentes do Kit De Lorenzo

Fonte: Sítio eletrônico <<https://delorenzo.com.br>>

1.1.4 COMPARATIVO ENTRE OS TIPOS DE KITS DIDÁTICOS

A Tabela 1 mostra uma comparação entre os tipos de Kits Didáticos abordados nas seções 1.1.2, 1.1.3 e 1.1.1.

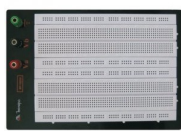
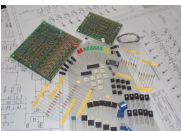

NOME DO KIT	Kits Didáticos		
	Protoboard	Montagem PCI	Tabuleiro de Inserção
IMAGEM			
Vantagens	Capacidade de executar infinitos tipos de montagem com componentes de vários tipos. Preço baixo em relação aos demais kits.	Minimiza a chance de montagem equivocada pelo aluno.	Possibilita melhor compreensão do que está sendo executado na montagem.
Desvantagens	Dificuldade de interpretação e correção do que está montado. Impossibilidade de prever como será a montagem impossibilitando o uso de automação.	Um kit diferente para cada tipo de prática que for necessária e limitação do conhecimento que está sendo aplicado, elevando o custo final do curso completo.	Limitação quanto à quantidade de componentes usados e necessidade de padronização do tamanho dos componentes. Preço elevado em relação aos demais kits.

Tabela 1 – Tabela Comparativa dos Kits Didáticos

1.2 NECESSIDADE DE CRIAÇÃO DE EQUIPAMENTO.

O problema de realimentação citado por Vlasenko e Bozhok (Vlasenko; BOZHOK, 2014) pode ser mitigado com a criação de Kit educacional para fazer a realimentação dos conteúdos ministrados. Observe na Figura 7 a comparação entre o ensino sem a kit educacional e com o uso do mesmo.

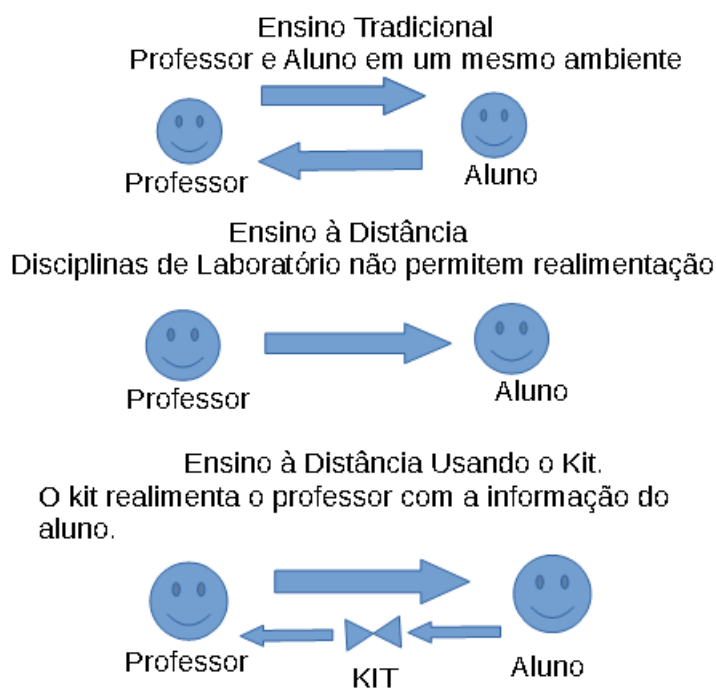


Figura 7 – Comparação do Uso e Não uso do Kit.

Fonte: Autor.

Outro ponto interessante a ser ressaltado sobre o kit é que este deve ter baixo custo de confecção para possibilitar acesso a uma quantidade maior de usuários. Para isto, deve ser usado componentes de baixo custo para sua construção.

O uso deste trabalho em sala de aula presencialmente também traz vantagens ao docente. Isto se deve ao fato de possibilitar a programação dos conteúdos a serem ministrados de forma a automatizar a correção dos trabalhos executados pelo discente. Isto proporciona a capacidade de fazer um aprendizado no qual o discente e o docente percebem qual conteúdo ainda não está devidamente absorvido possibilitando a utilização de outros meios para fazer o reforço dos mesmos. Em outras palavras, o equipamento dá ao docente a capacidade de automatizar a correção dos trabalhos e a própria condução da aula.

Assim sendo, o trabalho será útil tanto para a ministração de aulas práticas à distância quanto para a utilização dentro de sala de aula fazendo a automação dos recursos da aula para a ministração do conteúdo.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo estão os conceitos necessários para o entendimento dos conhecimentos aplicados na elaboração deste trabalho.

2.1 CAMA DE PREGOS

Este trabalho é um Kit educacional por Tabuleiro de Inserção de Componentes com o diferencial de fazer a transmissão do que está sendo executado no Kit através do princípio da "Cama de Pregos". Luppi Junior (LUPPI, 2017) cita a cama de pregos como um arranjo no qual colocamos vários pontos de teste em um placa para verificar seu funcionamento. O autor mostra a alta velocidade de execução do teste e a capacidade de medição de capacitância e indutância como vantagem deste método em relação à outros.

A Figura 8 mostra um equipamento destinado à este tipo de teste de placas de circuito impresso.

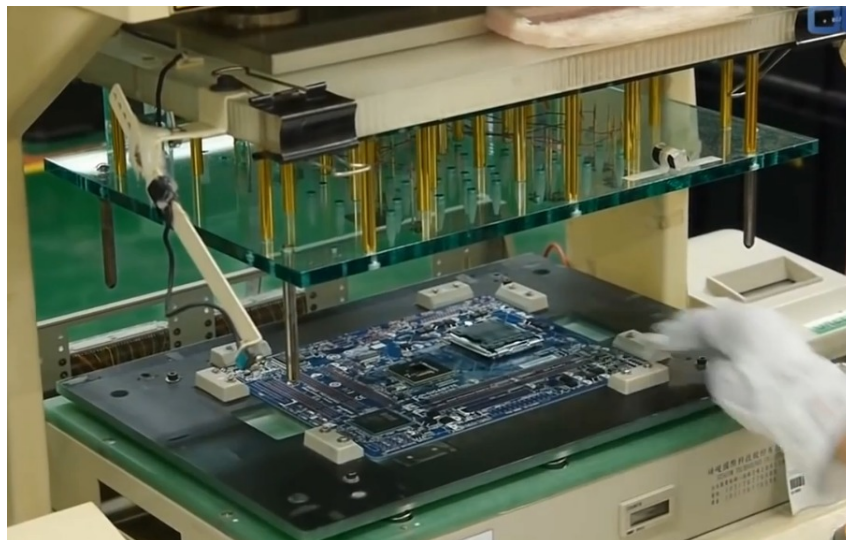


Figura 8 – Equipamento Usando o Princípio da Cama de Pregos

Fonte: Sítio eletrônico <<https://engenhariahibrida.com.br>>

O Sítio Eletrônico da SUNSOAR®¹ (SHENZHEN SUNSOAR CIRCUIT TECHNOLOGY, 2022) descreve o teste usando o princípio da cama de pregos como um teste de placas de circuito eletrônico que faz medições através de pontos específicos do dispositivo em teste. Nos pontos do dispositivo em teste são lidas grandezas elétricas de tensão e corrente. Os dispositivos testados com este tipo de testador geralmente apresentam pequenas marcas nos locais que as pontas de teste acoplaram-se para medição. O Sítio eletrônico afirma ainda que outros testes de qualidade são executados no dispositivo depois deste passar pelo teste da cama de pregos.

Usando o princípio da cama de pregos este trabalho se propõe a introduzir a capacidade do docente ter a informação do que foi absorvido pelo discente em disciplinas práticas de eletricidade e eletrônica através das medidas dos sinais elétricos em cada ponto da montagem executado pelo discente.

¹ <<http://pt.oem-pcb.com/news/what-is-bed-of-nails-tester-for-pcb-10745631.html>>

Isto equivale a dizer que estamos fechando a malha da educação à distância fornecendo uma re-alimentação do que é absorvido pelo discente para o docente atacando um dos problemas citados por (Vlasenko; BOZHOK, 2014). Desta forma, a implementação proposta neste trabalho é a transformação de uma malha em aberto composta pelo docente, conteúdo e discente em uma malha fechada com o equipamento fazendo o trabalho de realimentação. Com isto, o docente passa a ter a correta noção do que o discente absorveu dos conteúdos ministrados.

2.2 ESP32 E DEVKIT-V1

Para conectar-se à rede o equipamento utiliza as capacidades do chip ESP32 em um módulo cujo nome é DEVKITV1². O DEVKITV1 possui a capacidade de fazer 6(seis) leituras analógicas, quando é usada a rede Sem Fio. Feitas estas leituras o chip comunica-se com a rede sem fio usando o protocolo IEEE 802.11 possibilitando comunicação via rede local ou até mesmo via Internet.

As leituras coletadas pelo ESP32 são incorporadas em um código HTML³ (HTML, COM COMMUNITY, 2021) contido na prática enviada pelo docente ao discente. Esse código HTML é enviado através de um SCORM⁴ para o servidor LMS⁵. O HTML e o SCORM rodam diretamente no dispositivo do discente e o servidor LMS é a plataforma de ensino à distância que o docente utiliza para ministrar as aulas. A Figura 9 mostra as entradas e saídas existentes no DevKitV1 da DOIT®. Nesta Figura retirada do manual do fabricante (ZERYNTH CO., LTD., 2019), podemos ver que no módulo 16 entradas analógicas estão disponíveis para uso. Dentre os pinos restantes usaremos um para fazer o controle da entrada para registro na rede sem fio e temos a opção de fazer sinalização com outros. As entradas analógicas serão conectadas aos pinos que levam à placa de aplicação que definirá o tipo de prática a ser implementada pelo docente.

O ESP32 é O componente da placa que faz o processamento e onde estão armazenados os dados do programa. Trata-se de um sistema construído em um único chip SOC. Segundo Chang e Martin (GRANT; CHANG, 2003), o uso de sistemas em um único chip "System on Chip"(SOC) teve seu início em meados dos anos 90, quando o processo de fabricação de componentes de eletrônicos atingiu a capacidade de miniaturização de 250 a 350 nanômetros. Isto possibilitou a criação de componentes com vários equipamentos dentro de uma mesma pastilha.

Desta forma, os sistemas antes compostos por processadores, memórias e periféricos de entrada e saída ligados por um barramento e montados em uma placa de circuito impresso, são agora integrados em um mesmo chip. A tendência de SOC possibilitou a criação de equipamentos cada vez menores com muitos recursos. Dentre estes recursos, temos a rede sem fio que é a um excelente aliado para quem deseja criar produtos versáteis e com integração às tendências IOT e indústria 4.0 . Inicialmente os módulos SOC não possuíam rede sem fio integrado. Para utilização das redes sem fio era necessário acoplar aos módulos um periférico destinado a essa finalidade.

Conforme o manual do Fabricante(ESPRESSIF SYSTEMS CO., LTD., 2022), a família ESP veio alterar essa realidade trazendo dentro de seu chip o microprocessador LX106 Xtensa® da Tensilica que possui dois núcleos de 32 bits e suporte a rede sem fio. O ESP32 veio substituir o ESP8266 que mudou o mercado, que antes era dominado pelos módulos Arduino. O ESP32 é um chip poderoso e possuindo muitos recursos para uso em vários projetos, podendo facilmente ser usado em aplicações comerciais e industriais, obviamente respeitados os limites do equipamento.

² Placa de aplicação do componente ESP32

³ "Hyper Text Markup Language" - Língua de Marcação de Hipertexto (HTML)

⁴ "Sharable Content Object Reference Model" - Conteúdo Compartilhável do Modelo de Referência a Objeto (SCORM)

⁵ "Learning Management System" - Sistema de Gestão de Aprendizagem (LMS)



Figura 10 – Dimensões do DEVKIT-V1

Fonte: Autor.

2.4 HTML

HTML é uma sigla que significa linguagem de marcação de hipertexto. Trata-se da linguagem das páginas WEB e entendida pelos navegadores WEB instalados nas estações dos usuários (Internet Explorer, Google Chrome, Mozilla Firefox e Aple Safari). A linguagem html é baseada em "tags" e isto torna seu uso simples para os navegadores. Isto porque basta abrir uma tag "center" e depois encerrá-la para centralizar um texto. Uma tag é sempre colocada entre os sinais maior que e menor que, para abri-la basta fazer a declaração e para fechar basta coloca uma barra antes da tag.

Observe o código enviado pelo servidor para compor uma página muito simples contendo apenas uma frase centralizada mostrada na Figura 11:

```

    Abrir  [ícone]  pagina.html  Salvar  [menu]  x
    ~/Área de Trabalho

    <html>
    <title>PAGINA</title>
    <body>
    <center> Minha primeira pagina </center>
    </body>
    </html>

    Largura da tabulação: 8  Lin 6, Col 8  INS
  
```

Figura 11 – Arquivo em HTML

Fonte: Autor

Observe agora a visão deste arquivo através do navegador na Figura 12, foi usado o navegador Mozilla Firefox em sistema operacional Ubuntu.

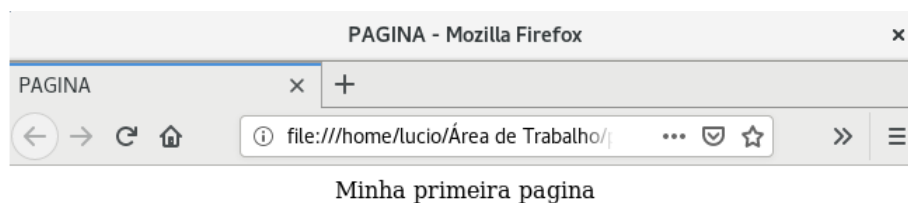


Figura 12 – Visão no Navegador

Fonte: Autor

Pode-se observar que é possível controlar atributos das páginas com o html, no caso usamos a tag "title" para informar ao navegador o título da página que foi mostrado na barra de títulos da aba aberta.

Para o navegador é muito simples interpretar todas as tags html e criar o texto na tela do usuário. Esta interpretação não é muito fácil para um operador humano, isto porque o arquivo html não possui uma característica amigável de interpretação pois destina-se a ser interpretado por máquinas.

Para criar páginas html pode se criar manualmente decorando todas as tags html e modificando seus atributos conforme as necessidades. Isto é trabalhoso e o único ganho envolvido é o conhecimento de tudo que está sendo informado ao navegador. Outra forma é recorrer a softwares que tornem esse trabalho mais amigável. Softwares do tipo "WYSIWYG"⁶ que facilitam muito a vida de quem deseja compor páginas em html provendo uma maneira mais simples de compor o que será mostrado e gerando o código necessário para adicionar às páginas.

O SCORM possui uma sintaxe muito próxima à do HTML. Isto é facilmente percebido pelo fato de ao abrir uma determinada tag < tag > tudo que vier em seguida é afetado por esta até o fechamento da mesma que ocorre com < / tag >. Além disto é possível acessar o kit utilizando uma página HTML bastando para isto que a mesma faça uso do endereço do kit e do ponto da placa que deseja fazer a leitura.

Existem muitos softwares dedicados a fazer arquivos SCORM para uso em sites de aprendizado à distância. Dentre estes podemos citar iSpring®, o CourseLab®, o LAMS® e o LCDS® da Microsoft®. No apêndice D está demonstrado o básico para usar a ferramenta online de geração de SCORM ISEAZY.com®.

2.5 SCORM

Segundo o sítio eletrônico da Rustici⁷(RUSTICI SOFTWARE SCORM, 2022), o SCORM é um conjunto de padrões técnicos para produtos de eLearning⁸. Ele fornece o método de comunicação e os modelos de dados que permitem que o conteúdo do eLearning e o LMS trabalhem juntos. Ele diz aos programadores como escrever código para que o que estes funcionem adequadamente com outro software de eLearning. SCORM é o padrão de eLearning mais amplamente usado disponível.

2.6 VISÃO GERAL DO SCORM

Segundo a documentação técnica do SCORM encontrada no sítio eletrônico de seus fabricantes(RUSTICI SOFTWARE SCORM, 2022), o SCORM é uma especificação que pretende padronizar a forma como os materiais de aprendizagem on-line são fornecidos para permitir que partes diferentes contribuam com recursos de aprendizagem e que esse recurso se integre a qualquer outro elemento

⁶ "What You See Is What You Get" - O que você vê é o que você tem (WYSIWYG)

⁷ <<https://scorm.com/>>

⁸ Educação pela Internet (eLearning)

SCORM de forma consistente. Pretende-se também permitir que os recursos individuais de aprendizagem sejam reutilizados mais facilmente. O aprendizado online é gerenciado por um Sistema de Gestão de Aprendizagem chamado **LMS**. Esta é essencialmente uma interface para um banco de dados de discentes e materiais de ensino. Geralmente é um site que permite aos discentes e docentes escolher e gerenciar seus cursos. Esses cursos consistem em material de treinamento que é revisado online e o **LMS** mantém um registro do status de cada discente em cada curso.

Um aspecto fundamental da especificação **SCORM** é que ele fornece uma especificação da forma como o progresso e a atividade são comunicados entre o **LMS** e os elementos de aprendizagem individuais. Ele define o que um sistema conformante **SCORM** envolve e define a sintaxe exata para a passagem de informações entre o **LMS**, o discente e os elementos de aprendizagem individuais. Obviamente, essa comunicação inclui o nome do discente e se este realizou o suficiente no curso para ser considerado aprovado. A especificação é mantida pela Advanced Distributed Learning e os diversos elementos e documentação são fornecidos em seu site www.adlnet.org. O **SCORM** consolida e incorpora elementos-chave de uma série de outras especificações.

Há um equivalente offline referido como AICC, mas na prática é largamente substituído pelo **SCORM** on-line que se desenvolveu a partir dele. Existem vários elementos para um sistema de treinamento on-line sob a especificação **SCORM**.

2.6.1 SISTEMA DE GESTÃO DE APRENDIZAGEM

O Sistema de Gestão da Aprendizagem, como explicado, é uma interface para um banco de dados de discentes e materiais do curso. Neste sistema são gerenciadas as lições, são gerenciados os discentes e gerenciadas os conteúdos das lições. Este sistema também permite que os usuários se registrem e criem um login e, em seguida, permite que eles escolham cursos para executar e monitorar seu progresso através das porções individuais de aprendizagem, permitindo que eles retomem de onde pararam e mantendo uma nota de qualquer pontuação.

Os administradores usam o **LMS** para fazer upload de cursos de design de materiais e atribuí-los a planos de aprendizagem. Alguns serão obrigatórios, alguns serão apenas para informações, enquanto outros incluirão um teste e o discente será obrigado a passar no curso. O **LMS** acompanha todos esses aspectos e permite que diferentes porções de treinamento sejam montadas em cursos ou planos de aprendizagem.

2.6.2 SCORM API

De acordo com a especificação **SCORM**, o **LMS** fornece um invólucro HTML no qual o conteúdo de aprendizagem é exibido. Neste invólucro ele fornece uma API⁹ que é essencialmente um conjunto de funções de script com uma especificação particular que fornece o canal para comunicação. Como esta API sempre tem uma especificação sob o **SCORM**, significa que os elementos de aprendizagem podem vir de uma variedade de fontes e ainda podem se comunicar corretamente.

2.6.3 SISTEMA DE GERENCIAMENTO DE CONTEÚDO DE APRENDIZAGEM.

Além disso, alguns sistemas possuem um Sistema de Gerenciamento de Conteúdo de Aprendizagem ou LCMS. Este é um sistema que se integra com o **LMS** e permite que os usuários gerenciem, carreguem, renomeiem os vários elementos do conteúdo fornecidos através do **LMS**. O conteúdo pode ser tão simples quanto um único vídeo para visualizar ou uma série de lições e testes fornecidos como uma única porção.

⁹ "Application Programming Interface" - Interface de programação de aplicações (API)

2.6.4 OBJETO DE CONTEÚDO SHARABLE (SCO)

A menor parte individual do treinamento de um sistema de treinamento **SCORM** é chamada de Objeto de Conteúdo Sharable ou **SCO**¹⁰. Esta é muitas vezes uma única lição e/ou teste, mas também pode ser uma série de "lições". O **SCO** consiste no material on-line pelo qual o discente navega, nos recursos que o material se baseia e em uma lista estruturada do que constitui esse **SCO**. Este último é chamado de manifesto. Os recursos incluídos no **SCO** podem ser referidos por links, permitindo assim que os mesmos recursos sejam reutilizados em diferentes materiais de aprendizagem sem serem duplicados.

2.6.5 RECURSOS

No **SCORM**, o recurso de termo refere-se aos elementos constituintes do **SCO** e, portanto, pode se referir a um arquivo que é a lição completa que contém todos os sons e imagens em si e não apenas os arquivos dos sons e arquivos de imagens separadamente.

2.6.6 MODELOS DE DADOS

O **SCORM** também especifica os nomes dados a vários elementos de informação que podem ser trocados entre uma aula e o curso ou **LMS**. Estes geralmente incluem o nome do usuário, se o discente completou o curso, qual a pontuação que obteve em qualquer teste e se passou ou não. Às vezes, a concessão de um passe é feita pelo **SCO** e às vezes pelo **LMS**. Também especifica a sintaxe que tanto o **SCO** quanto o **LMS** devem usar para se comunicar para garantir que ambos se entendam. Esses elementos de informação são chamados de modelos de dados no **SCORM**, e podem ser entendidos com as variáveis usadas em um programa.

2.6.7 NOMEANDO CONVENÇÕES E DOCUMENTOS DE CONTROLE

O **SCORM** ainda especifica como os vários elementos do pacote **SCORM** devem ser nomeados – quais caracteres são ou não são permitidos e assim por diante. As especificações dos elementos de aprendizagem tem como ponto de partida um conjunto de documentos adicionais chamados de "documentos de controle". Estes documentos de controle devem acompanhar qualquer elemento de aprendizagem.

2.6.8 PACOTE DE CONTEÚDO

O **SCORM** exige que o **SCO**, ou elemento de aprendizagem, seja fornecido em um arquivo zip embalado com toda a sua especificação. Este é o arquivo importado para o **LMS** e deve conter o elemento de aprendizagem ou **SCO** utiliza e não estão especificados em outros documentos de controle. Este arquivo deve conter também uma lista dos conteúdos com títulos individuais e identificadores que o **LMS** usará, esta lista é chamada de manifesto.

2.6.9 MANIFESTO

Esta é uma lista ou catálogo de todos os elementos fornecidos em um Pacote de Conteúdo **SCORM**. Além de listar os documentos de controle necessários, ele dá a tudo um título e um identificador (ID). É esse manifesto que o **LMS** lê para entender o que foi importado e geralmente lista o curso de acordo com um dos títulos do manifesto.

¹⁰ "Sharable Content Object" - Objeto de Conteúdo Compartilhado (**SCO**)

2.6.10 VERSÕES DE SCORM

Existem agora duas versões do **SCORM**, pois o padrão foi revisado em 2004 e uma nova linha de base de especificação lançada. **SCORM** v1.2 é a versão da especificação original que é mais utilizada (embora tenha sido atualizada desde então). O mais novo **SCORM** 2004 é menos amplamente suportado, mas se tornando cada vez mais popular por causa da funcionalidade estendida. Os modelos de dados e parte da sintaxe utilizada por eles variam entre v1.2 de **SCORM** e o v2004 mais recente.

Os dois formatos usam nomes de funções e modelos de dados ligeiramente diferentes e exigem um conjunto diferente de documentos **XML**¹¹ para acompanhar seus arquivos.

2.6.11 ELEMENTOS E VARIAÇÕES OBRIGATÓRIAS NO SCORM

Embora o **SCORM** seja uma especificação abrangente e detalhada, não é obrigatório apoiar todas as partes da especificação. Alguns **LMS** suportarão todos os modelos de dados, enquanto outros suportarão apenas alguns. Alguns disponibilizarão muitas informações para sua publicação, enquanto outros só permitem a comunicação do **SCO** para o **LMS**.

Alguns têm maneiras incomuns de apresentar a **API** e alguns têm requisitos particulares quando o **SCO** termina para permitir que o usuário navegue até a próxima seção.

2.6.12 ENTENDENDO O MANIFESTO SCORM

O manifesto **SCORM** documenta o conteúdo do pacote **SCORM** tanto quanto um manifesto de envio. Contém detalhes dos recursos utilizados no material e da forma como são organizados. Alguns **LMS** usam elementos do manifesto como identificador padrão ao importar a embalagem.

O código 2.1 mostra um exemplo de arquivo **SCORM**.

```
1 <?xml versão="1.0" codificação="utf-8"?>
2
3 <manifest
4
5   identificador="MANIFEST_ID"
6
7   version="1.100000"
8
9   xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
10
11  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
12
13  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
14
15  xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2 imscp_rootv1p1p2.xsd
16
17  http://www.imglobal.org/xsd/imsmd\_rootv1p2p1 imsmd_rootv1p2p1.xsd
18
19  http://www.adlnet.org/xsd/adlcp\_rootv1p2 adlcp_rootv1p2.xsd" >
20
21 <organizações padrão=" ORG_ID">
22
23 <sorganizador="ORG_ID">
24
25 <ORG_TITLE></título>
26
```

¹¹ "eXtensive Markup Language" - Linguagem extensiva de Marcação de Objeto (**XML**)

```

27 <item identificador="ITEM_ID" identifierref="RESOURCE_ID">
28
29 <título>ITEM_TITLE</título>
30
31 </item>
32
33 </organização>
34
35 </organizações>
36
37 < recursos>
38
39 <resource identifier="RESOURCE_ID" tipo="webcontent" adlcp:scormtype="sco" href="ScormCheck.html">
40
41 <file href="ScormCheck.html" >
42
43 <file href="ScormCheck.swf" >
44
45 </recurso>
46
47 </recursos>
48
49 </manifest>

```

2.6.13 ELEMENTOS MANIFEST_ID

Para identificar o pacote específico, ele tem um identificador próprio que é definido pelo programador, este é o MANIFEST_ID no exemplo. Também inclui referência a uma série de documentos (chamados de documentos de controle) que incluem partes fundamentais da especificação **SCORM**, como a forma como os IDs e títulos devem ser nomeados (referido como convenção de nomeação). Esses documentos de controle são diferentes para **SCORM 1.2** e **SCORM 2004**.

2.6.14 ORGANIZAÇÕES

Dentro do manifesto há uma subseção chamada organizações. Esta subseção é uma lista de tratamentos que os recursos do pacote receberam e contém uma ou mais subseções detalhando cada estrutura de organização individualmente. Cada manifesto deve conter, no mínimo, uma lista de organização.

Cada organização tem um identificador – isso é ORG_ID no exemplo é o elemento frequentemente usado pelo **LMS** como um nome padrão para a lição quando é importado, mostrado no código **2.2**.

```

23 <sorganizador="ORG_ID">
24
25 <ORG_TITLE></título>

```

O manifesto exige que a seção de organizações especifique se o ID da organização individual que deve ser usado como padrão.

```

23 <sorganizador="ORG_ID">

```

Dentro da organização há uma lista de itens que são usados na ordem em que aparecem no arquivo. Cada item tem um identificador (ITEM_ID no exemplo) e um título (ITEM_TITLE) e também se refere a um recurso (por meio de um ID) que é basicamente a sua identificação de conteúdo (RESOURCE_ID).

2.6.15 RECURSOS

A subseção final do manifesto é a seção de recursos. Isso lista os recursos utilizados em cada item da aula. Os recursos estão vinculados ao item relevante pela referência do identificador que, em nosso exemplo, é RESOURCE_ID. Este ID é então seguido pelo tipo de conteúdo que ele contém e o nome de arquivo (e localização, se necessário) do recurso específico que deve ser usado para lançar essa seção da lição.

```
39 <resource identifier="RESOURCE_ID" tipo="webcontent" adlcp:scormtype="sco" href="ScormCheck.html">
```

Após a identificação do recurso, segue-se uma lista de todos os arquivos utilizados neste, incluindo a repetição do arquivo principal especificado na primeira linha, neste caso "ScormCheck.html".

```
41 <file href="ScormCheck.html" >
42
43 <file href="ScormCheck.swf" >
```

2.7 JAVASCRIPT® E JSON®

Javascript®, de acordo com o sitio da Mozilla ([MOZILLA FOUNDATION, 2022](#)) é uma linguagem de programação que executa aplicações através de um navegador. Desta forma os programas podem ficar integrados em um HTML ou SCORM e serem executados pelo navegador ou até em ambientes preparados para executá-los.

JSON®¹² é um padrão de notação de objetos em javascript®. De acordo com o sitio da JSON ([JSON ORGANIZATION, 2022](#)), trata-se de um padrão leve e intercambiável que adiciona facilidade para escrita e compreensão para o utilizador e agilidade para a troca de informação através de navegadores.

Na Figura 13, do sitio "Visual Dicas" ([FERREIRA, 2020](#)), podemos ver que o JSON é formado por chaves (*Keys*) e valores (*values*).

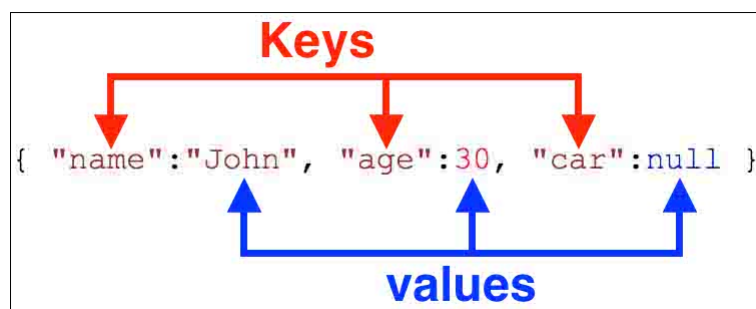


Figura 13 – Estrutura do JSON

Fonte: Ferreira, Angelo Luis

Desta forma, enviamos pela rede sem fio um arquivo JSON® contendo os nomes dos pontos do Kit e as suas leituras respectivas, assim sendo o nome dos pontos é a chave (*"key"*) e as respectivas leituras são os valores (*"values"*). E através de programas em javascript® podemos fazer a manipulação necessária dos dados lidos e interpretá-los na aplicação SCORM ou HTML. Um ponto importante a ser ressaltado no uso do formato JSON® é a interoperabilidade, uma vez que este pode ser lido em diferentes ambientes ou navegadores. No subseção 5.1.6 é demonstrado o arquivo JSON liberado pelo programa da placa principal para facilitar o uso em qualquer tipo de software que se deseje construir para a leitura do módulo.

¹² "Javascript Object Notation" - Notação de Objeto em Javascript (JSON)

3 DESCRIÇÃO GERAL DO TRABALHO

A intenção deste capítulo é descrever o funcionamento do trabalho de forma geral. As minúcias de cada parte do kit educacional serão tratadas em capítulos dedicados a cada item do mesmo.

Conforme citado no capítulo 1, o trabalho faz a automação da aula dos conteúdos de eletricidade e eletrônica e pode comunicar ao docente o que foi feito pelo discente de forma pré programada.

3.1 PRINCÍPIO DE FUNCIONAMENTO

Este trabalho funciona analogamente a uma cama de pregos citada no capítulo 2, fazendo a leitura da montagem executada pelo discente e enviando o resultado dos testes ao docente. O envio de resultados é feito através de uma conexão sem fio que pode comunicar se através da Internet ou de uma rede interna, com o equipamento no qual a aula está sendo conduzida.

Um software construído em [HTML](#) ou [SCORM](#) instalado em um dispositivo do discente que pode ser um dispositivo portátil ou um computador de mesa. Este software libera uma senha para ser inserida no [LMS](#) onde teremos um código em [SCORM](#) fazendo a validação desta senha e registro no [LMS](#).

3.2 HARDWARE DO EQUIPAMENTO

O hardware do trabalho é composto por uma placa na qual está inserido o Módulo do Esp32 e conexões para suas entradas analógicas, uma fonte de alimentação e uma placa Tabuleiro para as montagens executadas conforme a necessidade da aula prática a ser ministrada. E este hardware deve ser acondicionado uma caixa para agregar proteção e mobilidade.

3.2.1 CAIXA DO EQUIPAMENTO

Para acomodar o circuito e proporcionar mais facilidade de transporte e robustez foi escolhido colocá-lo em uma maleta da marca Bauker®. A maleta escolhida possui as dimensões de 380mm x 256mm x 120mm, possibilitando acomodar o tabuleiro e os circuitos que são necessários para o seu funcionamento. A maleta está mostrada na Figura 14

A maleta é feita de alumínio e possui uma forração em EVA capaz de proteger os circuitos eletrônicos do Kit.

Outra opção com custo mais baixo é a utilização de uma caixa feita de MDF conforme mostrado na Figura 15.

Para a escolha da versão deve-se atentar para as vantagens e desvantagens de um acondicionamento em relação ao outro. As principais características de um e de outro estão na Tabela comparativa 2.

Pela Tabela comparativa 2 podemos ver que a melhor utilização para a caixa em MDF é para usos internos em um ambiente acadêmico com a presença do orientador da prática e a maleta de alumínio proporciona um acondicionamento com maior robustez para ambientes externos. A maleta em alumínio possui um custo cerca de R\$ 100,00 maior que a caixa em MDF.



Figura 14 – Maleta para Acondicionar o Kit

Fonte: <<https://produto.mercadolivre.com.br/MLB-1856758550>>



Figura 15 – Maleta em MDF para Acondicionar o Kit

Fonte: <<https://produto.mercadolivre.com.br/MLB-1671661066>>



Caixas para Acondicionamento		
Tipo de Acondicionamento	Caixa MDF	Maleta Alumínio
IMAGEM		
Vantagens	Custo mais atrativo. Livre e interferência no sinal da rede sem fio	Robustez da maleta dá maior proteção mecânica aos equipamentos.
Desvantagens	Fragilidade da caixa para trabalhos externos.	Causa um pouco de interferência reduzindo o alcance do módulo. Custo mais elevado.

Tabela 2 – Tabela Comparativa das Caixas de Acondicionamento

4 HARDWARE DO KIT

O hardware do kit é composto por uma placa principal, uma placa tabuleiro e placas de componentes para inserção no tabuleiro. A placa principal é responsável por receber os sinais elétricos originados no tabuleiro e encaminhar através da rede sem fio para o programa que está em um dispositivo que processa o [SCORM](#) ou o [HTML](#) para interpretar o que está montado neste tabuleiro. O dispositivo que executa o programa que indica ao discente como fazer a montagem no tabuleiro pode ser um desktop, tablet, notebook ou um smartphone.

4.1 PLACA PRINCIPAL

A placa principal foi desenvolvida para ajustar os sinais do meio externo, no caso o tabuleiro, para os níveis de sinal que são trabalhados no ESP32. Isto acontece porque o ESP32 trabalha com níveis de tensão de 0V a 3,3V e necessita de proteção e adaptação para os níveis de tensão trabalhados na aula prática que podem variar de 0V as 9V. Desta forma, foram adicionados amplificadores operacionais para garantir que os níveis que serão entregues ao ESP32 estejam dentro da faixa entre 0V e 3.3V.

4.2 DIAGRAMA ESQUEMÁTICO DA PLACA PRINCIPAL.

Inicialmente foi executado o programa usando o NODEMCU32S. O motivo para isso foi que o NODEMCU32 da AI-THINKER estava facilmente acessível e seu manual estava amplamente divulgado na internet ([SHENZHEN AI-THINKER TECHNOLOGY CO., LTD., 2019](#)). Como o chip principal é o ESP32, todos os programas feitos para o NODEMCU32S deverão funcionar no DEVKITV1 da DOIT após algumas adaptações. Testado o programa no NODEMCU32S, foi feita a adaptação do código para o DEVKITV1 da DOIT Co. LTDA. E a partir disto iniciado o processo de confecção da placa principal.

Na Figura 16 é mostrado o acoplamento feito pelos amplificadores operacionais. Observe se que o amplificador operacional está na configuração "Buffer" que não altera o sinal de entrada e simplesmente fornece corrente para o circuito impedindo interferência de ruídos vindos de flutuações externas. Outro detalhe é o resistor variável que faz um divisor de tensão visando reduzir proporcionalmente o sinal até o nível da entrada do ESP32. Para calibração deste resistor variável é necessário colocar um nível 9V na entrada e medir 3,3V no ponto de teste. Desta forma, enquanto o sinal externo varia de 0 a 9V, a entrada do ESP recebe proporcionalmente de 0 a 3,3V. Nesta placa para aumentar a sensibilidade do ajuste foi escolhido um resistor variável de 12 voltas. A alimentação do amplificador operacional é feita com 12V, mas o acoplamento ajusta o nível de 9V para 3,3V.

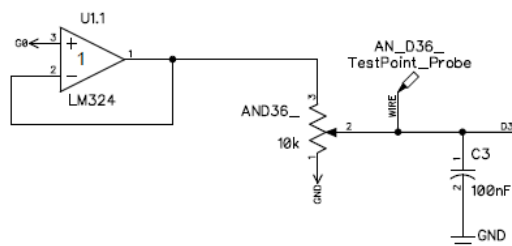


Figura 16 – Acoplamento por Amplificadores Operacionais

Fonte: Autor

O diagrama esquemático completo da placa principal está mostrado na Figura 17.

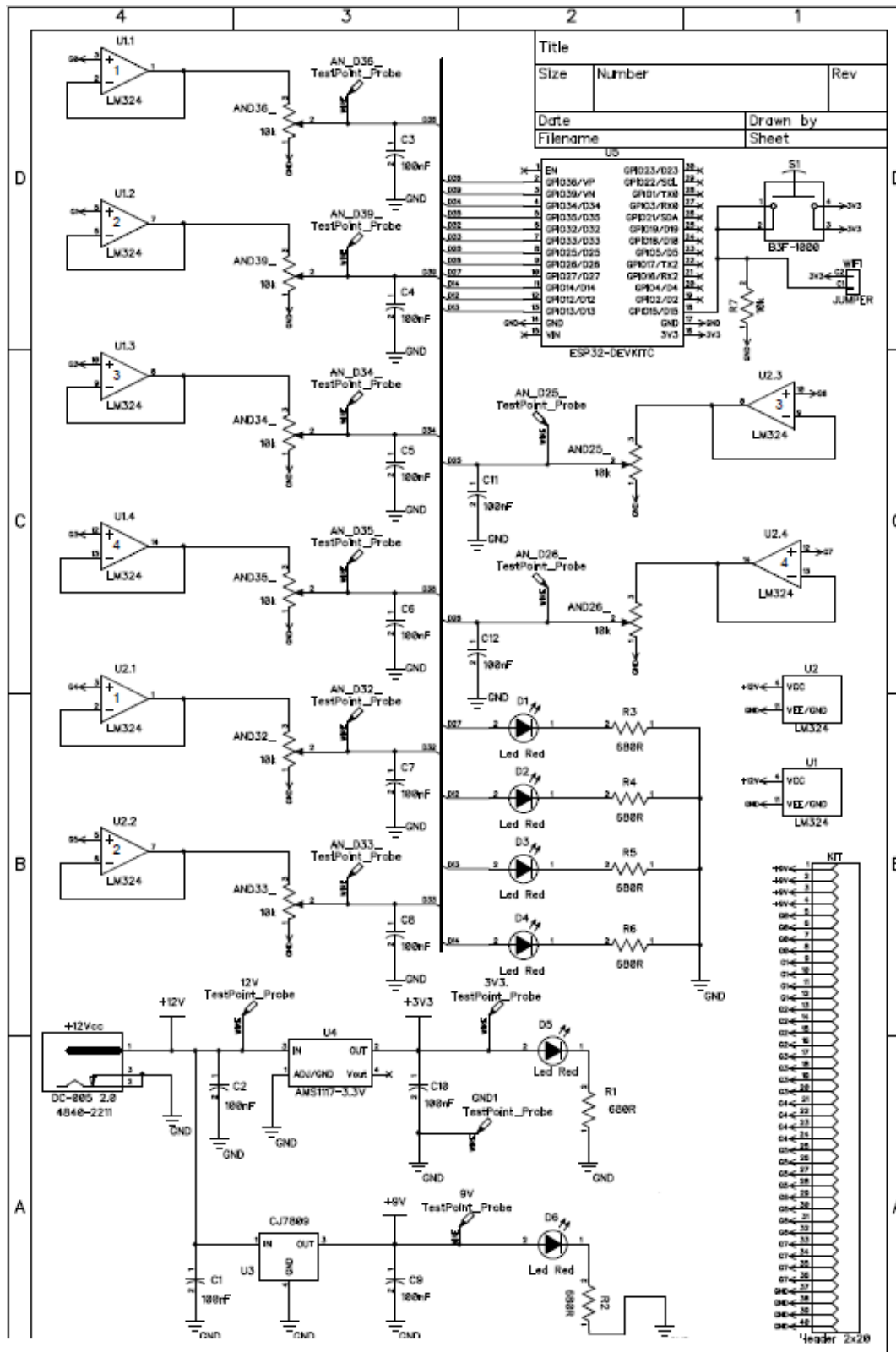


Figura 17 – Placa Principal do PROGRAMA

Fonte: Autor

No diagrama completo podemos ver também os dois reguladores de tensão que recebem alimentação

de 12V e fornecem 9V para alimentar o Tabuleiro e 3,3V para alimentar o ESP32. Também podemos ver a conexão com o Tabuleiro que é feita através do Conector de 40 pinos do tipo "Header 2x20 2mm de passo". Também foram adicionados 6(seis) LEDs que são dois para sinalizar a saída dos reguladores de tensão e quatro para sinalizar estados específicos da placa caso necessário.

A visão das camadas da placa CPU do protótipo está mostrada na Figura 18 e a vista superior da placa CPU do protótipo está mostrada na Figura 19.

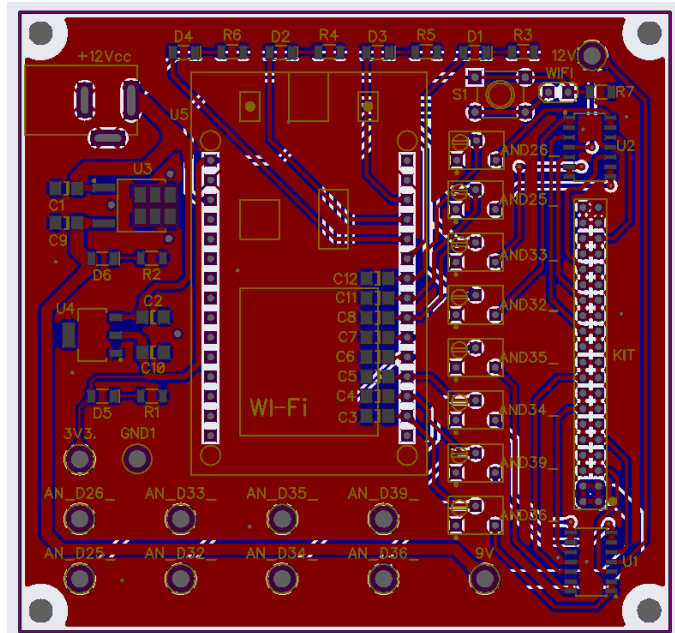


Figura 18 – Camadas da Placa CPU

Fonte: Autor

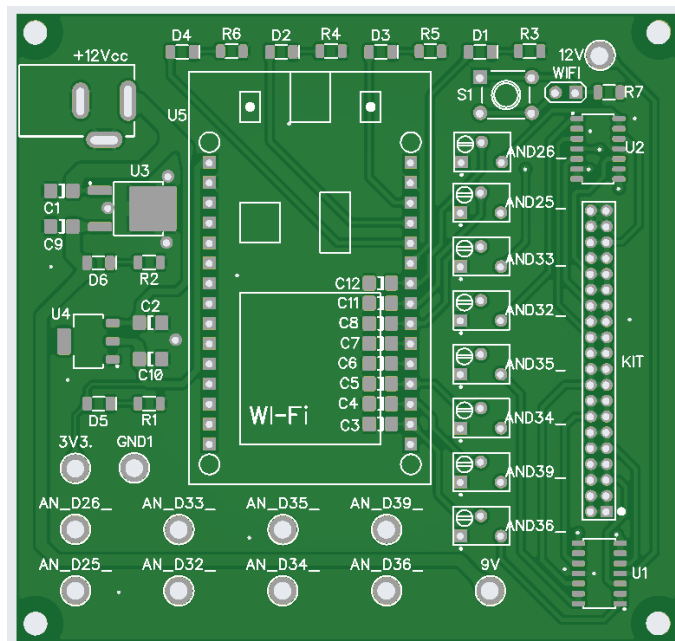


Figura 19 – Vista Superior da Placa CPU

Fonte: Autor

A placa do tabuleiro foi criada para receber os bornes apenas parafusando-os na mesma, dispensando o uso de solda e evitando possíveis pontos de corrosão e alteração de medidas devido à diferenças de potencial entre a solda desgastada e a placa. As vistas superior, inferior e das camadas da placa tabuleiro estão mostradas nas Figuras 22, 23 e 24 respectivamente.

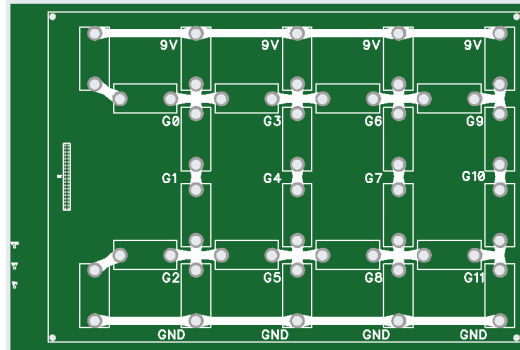


Figura 22 – Vista Superior da Placa de Tabuleiro

Fonte: Autor

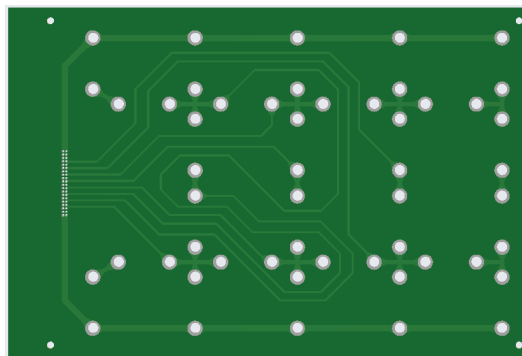


Figura 23 – Vista Inferior da Placa de Tabuleiro

Fonte: Autor

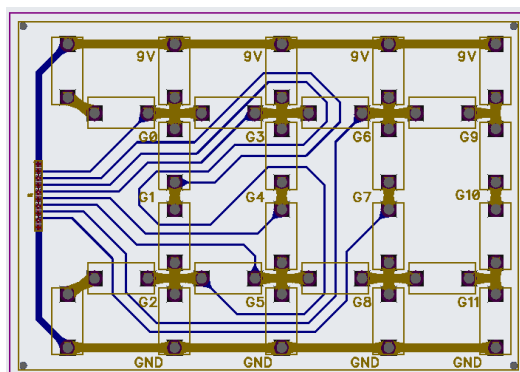


Figura 24 – Visão das Camadas da Placa de Tabuleiro

Fonte: Autor

Para a criação dos módulos de componentes utilizáveis no trabalho foi criado uma placa utilizando o método de "panelização" que consiste em fazer a placa com o tamanho exato e recortes necessários para sua utilização, segundo o sítio eletrônico (PRODUZA INTELIGÊNCIA EM MANUFATURA, 2022). É possível ver que foi também feito um ponto de conexão além dos necessário para os bornes e para o componente, este ponto será utilizado para medições que o discente necessita fazer nas aulas práticas. A vista superior do placa dos resistores está mostrada na Figura 25 e vista inferior da placa dos resistores está mostrada na Figura 26.

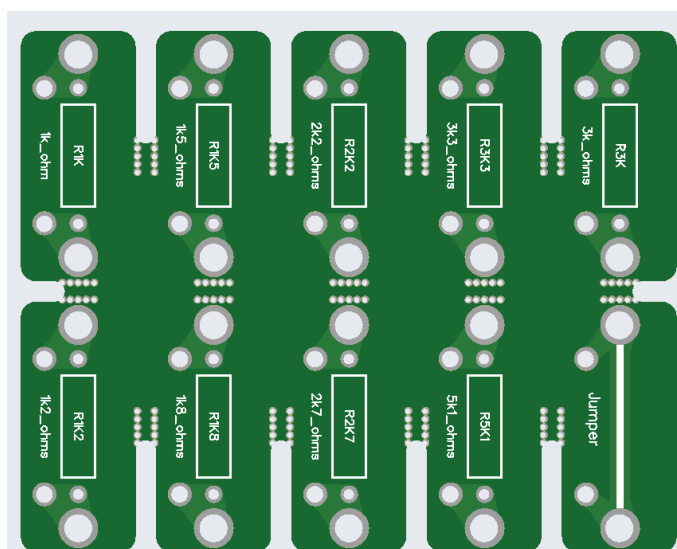


Figura 25 – Vista Superior da Placa de Resistores

Fonte: Autor

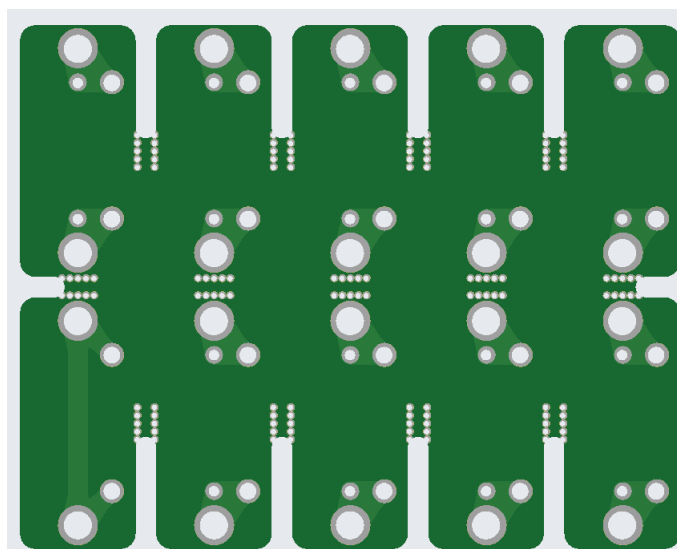
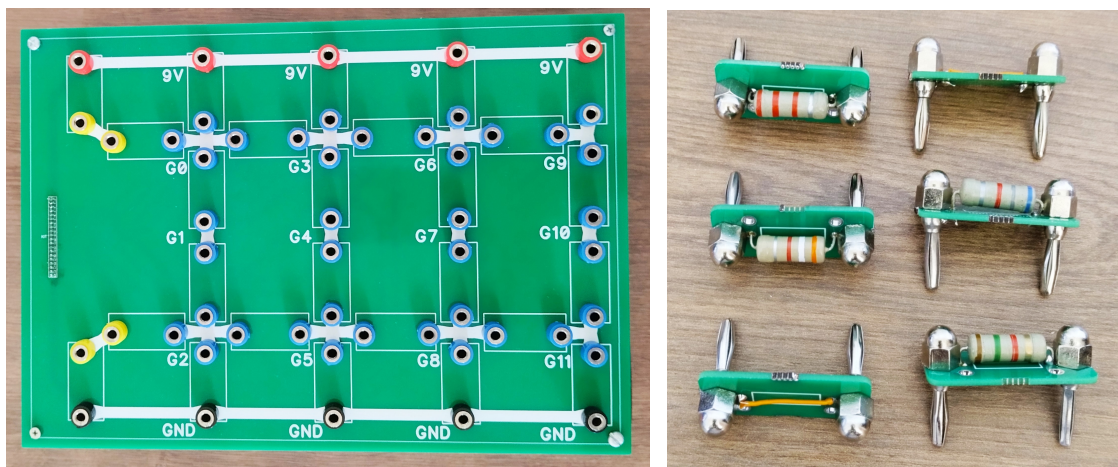


Figura 26 – Vista Inferior da Placa de Resistores

Fonte: Autor

A foto da placa do tabuleiro confeccionada e já com os bornes instalados está mostrada na Figura 27 ao lado dos Componentes montados para Inserção. A Figura 28 mostra fotos da Placa Principal montada com os componentes vista de ambos os lados.

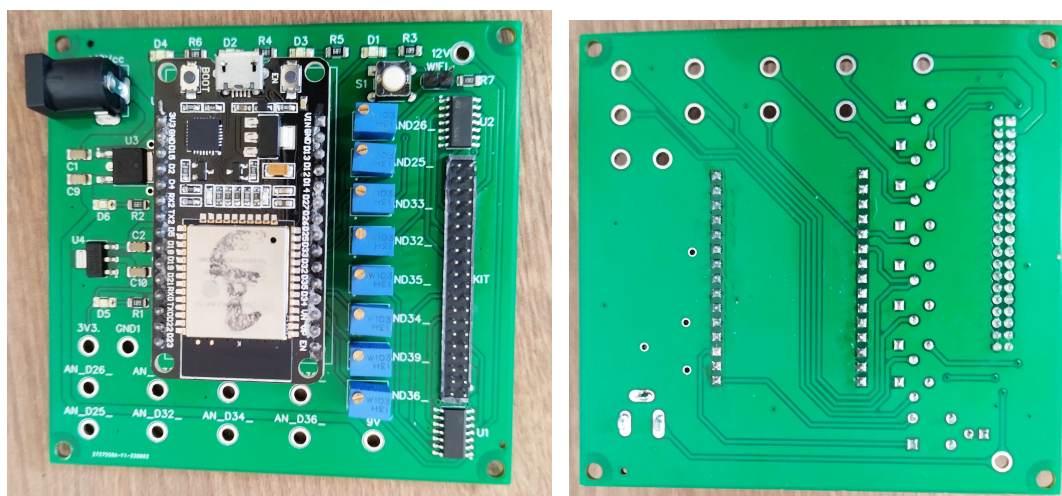


(a) Tabuleiro de Inserção

(b) Componentes Montados

Figura 27 – Tabuleiro de Inserção e Componentes

Fonte: Autor



(a) Lado dos Componentes

(b) Lado da Trilhas

Figura 28 – Placa Principal Montada

Fonte: Autor

Na Figura 29 vemos a foto das conexões entre as partes do Kit e na Figura 30 vemos a foto do KIT Montado na maleta pronto para utilização



Figura 29 – Conexões entre as Partes do KIT

Fonte: Autor



Figura 30 – Kit Montado na Maleta

Fonte: Autor

A confecção da placa protótipo deste trabalho foi feita a empresa JLCPCB Factories®¹. Esta empresa produz placas na China e faz envio para todas as partes do mundo. Para exemplificar, a Figura 31 mostra as etapas de confecção da placa.

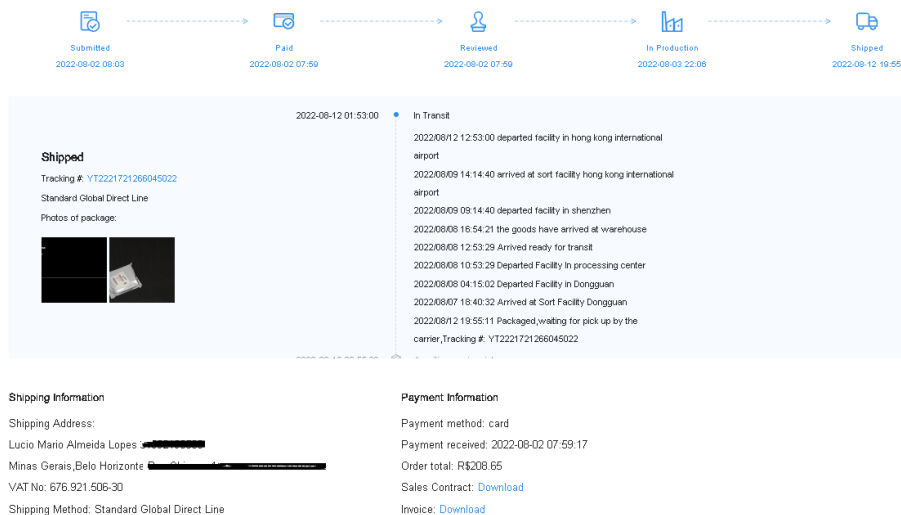


Figura 31 – Ordem de Serviço - CPU

Fonte: Autor

Pode-se verificar que desde o recebimento do Pagamento da Ordem, ocorre a verificação dos arquivos enviados e posteriormente vai para a produção. Durante a etapa de verificação foi necessário fazer reenvio de arquivos em algumas placas que foram solicitadas. A ordem foi submetida no dia 02/08/2022, entrou em produção em 03/08/2022 e despachada em 12/08/2022. A Tabela 3 mostra nos "logs" de rastreamento que 12/08/2022 foi a data de saída da China e a entrega final foi 19/08/2022. Todo o processo de envio dos arquivos até a entrega final consumiu 17 dias.

Data	Hora	Cidade	Evento
07/08/2022	18:40	Dongguan	recebido pela transportadora
08/08/2022	04:15	Dongguan	entrada no centro de processamento.
08/08/2022	10:53	Dongguan	liberado para sair do centro de processamento.
08/08/2022	12:53	Shenzhen	liberado para sair em trânsito
08/08/2022	16:54	Shenzhen	mercadorias chegaram ao armazém
09/08/2022	09:14	Shenzhen	partiu do aeroporto de Shenzhen
09/08/2022	14:14	Hong Kong	chegou ao aeroporto internacional de Hong Kong
12/08/2022	12:53	Hong Kong	partiu no aeroporto internacional de Hong Kong
13/08/2022	17:00	Hong Kong	chegou ao aeroporto para destino internacional
15/08/2022	20:32	CURITIBA	objeto recebido pelos correios do brasil
16/08/2022	04:23	CURITIBA	fiscalização aduaneira finalizada
16/08/2022	04:55	CURITIBA	objeto em trânsito - por favor aguarde
18/08/2022	13:56	BELO HORIZONTE	objeto em trânsito - por favor aguarde
19/08/2022	09:42	BELO HORIZONTE	objeto saiu para entrega ao destinatário
19/08/2022	15:20	BELO HORIZONTE	objeto entregue ao destinatário

Tabela 3 – Tabela de Registros de Despacho da Mercadoria

Fonte: Autor

¹ <<https://jlcpcb.com/>>

4.2.1 CUSTOS DO PROJETO

A Tabela 4 mostra os custos envolvidos na produção do kit. Embora estes custos estejam ajustados para uma unidade (um kit), alguns componentes e as placas foram necessários pedidos maiores que uma única unidade. Os custos também podem diminuir caso a produção ocorra em quantidades maiores.

Item	Quantidade	Descrição	Valor
Placa Principal	1	Confecção da placa	R\$ 2,08
Componentes Placa Principal	1	Componente e Montagem na placa.	R\$ 100,67
Placa Módulos Componentes	1	Confecção da placa	R\$ 2,10
Placa Tabuleiro	1	Confecção da placa	R\$ 29,87
Bornes	52	Para o tabuleiro	R\$ 104,00
Barra de Pinos	1	Para o tabuleiro	R\$ 3,70
Pinos	20	Para os módulos de componentes	R\$ 40,00
Porca Calota	20	Porca	R\$ 16,00
Cabo	1	Conexão entre placas	R\$ 16,54
Esp32	1	Microcontrolador	R\$ 29,00
Maleta	1	Maleta	R\$ 145,59
Total			R\$ 489,55

Tabela 4 – Tabela de Custos do Projeto

Fonte: Autor

Os custos listados na Tabela 4 foram os executados neste trabalho, mas podem variar dependendo da escolha dos fornecedores dos materiais. Para ilustrar o quanto pode acontecer nestas variações, na Tabela 5 está mostrada a variação de custos entre três fornecedores para cada item. Estes dados foram retirados da Internet usando o sítio de busca Google® na opção "Shopping". Pode-se notar a diferença de preços de R\$ 207,97. Não foram analisados nesta comparação os custos de confecção das placas de circuito eletrônico.

Item	Sítio 1	Sítio 2	Sítio 3
52 Bornes	R\$ 141,44	R\$ 131,04	R\$ 104,00
1 Barra de Pinos	R\$ 5,80	R \$ 4,58	R\$ 3,70
20 Pinos	R\$ 80,00	R\$ 66,00	R\$ 40,00
20 Porca Calota	R\$ 45,00	R\$ 22,00	R\$ 16,00
1 Cabo	R\$ 55,00	R\$ 27,50	R\$ 16,54
1 Esp32	R\$ 80,66	R\$ 38,00	R\$ 29,00
1 Maleta	R\$ 154,90	R\$ 153,91	R\$ 145,59
Totais	R\$ 562,80	R\$ 443,03	R\$ 354,83

Tabela 5 – Tabela de Comparativa de Custos

Fonte: Autor

5 PROGRAMAÇÃO DA PLACA PRINCIPAL DO KIT

5.1 O CÓDIGO

O código completo do módulo está no apêndice C deste trabalho, e as partes mais importantes serão explicitadas nesta seção. O código foi dividido em 6(seis) arquivos para facilitar alterações e revisões futuras. Os arquivos são:

- TCC.ino - Arquivo com a descrição geral do programa.
- Bibliotecas.ino - Contêm as bibliotecas do programa.
- Objetos.ino - Contêm as declarações de variáveis globais do programa.
- Pinos.ino - Contêm as definições dos pinos utilizados.
- Setup.ino - Contêm o setup do programa.
- looping.ino - Contêm o looping principal do programa.

5.1.1 TCC.INO

TCC.INO é o arquivo no qual está a descrição do programa contendo o nome do Autor e do Instituto e uma breve explanação do que o programa faz.

5.1.2 BIBLIOTECAS.INO

Neste arquivo consta a declaração das bibliotecas necessárias para o funcionamento do programa. As bibliotecas são:

- WIFI.h, necessária para utilizar a rede sem fio.
- WIFILIENT.h necessária para aceitar clientes do serviço Web.
- WIFIAP.h necessária para usar o modo Ponto de Acesso.

5.1.3 OBJETOS.INO

No arquivo Objetos.INO estão as declarações dos objetos SSID, identificação da rede e do servidor com sua respectiva porta.

5.1.4 PINOS.INO

No arquivo Pinos.INO está a definição de cada pino para a interface do Arduino fazer a compilação do programa.

5.1.5 SETUP.INO

No arquivo Setup.INO está o setup do programa, que faz a configuração inicial do programa para o seu correto funcionamento. Neste arquivo estão as configurações dos modos dos pinos, da saída serial e do servidor da rede sem fio.

5.1.6 LOOPING.INO

Este arquivo é o Looping principal do programa. Neste está configurado o servidor web com as chamadas para a leitura de cada pino.

No Looping principal vale destacar os laços de leitura das entradas analógicas. Este laço demonstrado a seguir mostra que ao receber a requisição "GET" de uma entrada, o programa entrega a leitura do pino correspondente.

```

33     if (currentLine.endsWith("GET /GO")) {
34         int entrada = analogRead(GPI036);           // GET /GO Ler o Valor de GPI036
35         client.print(entrada);
36     }

```

Outro detalhe a ressaltar é a entrega de um arquivo json® quando o programa recebe um requisição "GET" "tudo", conforme mostrado nas linhas do programa a seguir.

```

65     if (currentLine.endsWith("GET /tudo")) {
66         String json = fazjson();                     // gera o json
67         client.print("Content-Type");
68         client.print(",");
69         client.print("application/json");
70         client.print(json);
71     }

```

No trecho mostrado, ao receber o "GET tudo" o programa responde com o cabeçalho informando que o conteúdo é um JSON e em seguida envia o JSON com a estrutura criada pela função "fazjson". A função "fazjson" está mostrada a seguir.

```

80 String fazjson(){
81     int entg0 = analogRead(GPI036);
82     int entg1 = analogRead(GPI039);
83     int entg2 = analogRead(GPI034);
84     int entg3 = analogRead(GPI035);
85     int entg4 = analogRead(GPI032);
86     int entg5 = analogRead(GPI033);
87     bool entg6 = digitalRead(GPI026);
88     bool entg7 = digitalRead(GPI025);
89     String resultado = "{";
90     resultado = resultado + "\"G0\":\""+entg0+"\", ";
91     resultado = resultado + "\"G1\":\""+entg1+"\", ";
92     resultado = resultado + "\"G2\":\""+entg2+"\", ";
93     resultado = resultado + "\"G3\":\""+entg3+"\", ";
94     resultado = resultado + "\"G4\":\""+entg4+"\", ";
95     resultado = resultado + "\"G5\":\""+entg5+"\", ";
96     resultado = resultado + "\"G6\":\""+entg6+"\", ";
97     resultado = resultado + "\"G7\":\""+entg7+"\", ";
98     resultado = resultado + "}";
99     return resultado;
100 }

```

A função "fazjson" cria uma string com a estrutura mostrada na Figura 32.

```
{  
  "G0":<valor>,  
  "G1":<valor>,  
  "G2":<valor>,  
  "G3":<valor>,  
  "G4":<valor>,  
  "G5":<valor>,  
  "G6":<valor>,  
  "G7":<valor>  
}
```

Figura 32 – Formato do JSON liberado pelo Programa

5.1.7 RESPOSTA DA ENTRADA ANALÓGICA DO ESP32

De acordo com o manual do fabricante ([ESPRESSIF SYSTEMS CO., LTD., 2022](#)), o ESP32 possui conversores ADC SAR¹ que funcionam fazendo sucessivas comparações com um conversor DAC interno e para isto este conversor DAC recebe uma tensão de referência para calibrar sua saída. O problema que ocorre no ESP32 é a diferença entre as tensões de referência internas do chip, causando erros da ordem de 6% nas regiões mais próximas dos extremos(alto e baixo) da curva de resposta do conversor. Para corrigir pode se usar vários métodos, o mais comum destes é encontrar uma tabela de conversão dos valores medidos para os valores que deveriam ser medidos, a outra forma é gerar uma função polinomial do erro e ajustar através deste polinômio a saída.

O método que usado neste trabalho foi contornar os valores altos e baixos da curva de conversão e manter as leituras na região mediana, que apresenta uma resposta mais linear e fazer ajuste do circuito de entrada para trabalhar nesta região.

De acordo com Reis ([REIS, 2017](#)), os conversores ADC são baseados no critério Nyquist-Shannon que limita a banda de passagem da amostragem do sinal analógico. A taxa máxima de amostragem de acordo com o manual é de 200k/s no modo RTC e 2M/s no modo Digital. Para ministrar aulas básicas de eletricidade e eletrônica estes valores são suficientes, porém para fazer leitura sinais com maior frequência deve se buscar um conversor com uma banda de passagem maior.

¹ "Successive Approximation Register" - Registro de aproximação sucessiva (SAR)

6 CÓDIGOS JAVASCRIPT PARA VALIDAÇÃO DA ATIVIDADE

6.1 CÓDIGO JAVASCRIPT DA CAMA DE PREGOS

O kit funciona como uma cama de pregos fazendo a verificação de cada ponto através dos níveis de tensão obtidos pelas leituras analógicas. Depois de recebidas pelo dispositivo que recebe as leituras, um javascript é executado para fazer a interpretação do que está sendo obtido. O código de verificação em javascript foi criado para verificar a montagem de dois resistores em serie e está mostrado no código a seguir:

```

1 // 1)Teste se tem tensão no KIT (verificar se tem mais de 96%(3925) de 9V
2
3 if (G3 > 3925)
4     {
5         E = "E/ Ok, o kit está alimentado com tensão acima de 96% da fonte 9 V; "
6     }
7 else
8     {
9         E = ""
10    }
11
12 //
13
14 // 2)Teste se o kit está desligado com tensão menor de 25% no KIT (verificar se tem menor de 25%(1023)
15 // de 9V
16 if (G3 < 1023)
17     {
18         E0 = "E0/ O kit está desligado verifique a alimentação; "
19         Erro0 = 0
20     }
21 else
22     {
23         E0 = ""
24         Erro0 = 1
25     }
26
27 //
28
29 // 3)Teste se a Tensão de alimentação do Kit está desregulada menor que 90% e maior que 25%;
30 // (verificar se tem maior de 10%(410) de 9V e menor que 96% (3925)
31
32 if ((G3 <= 3925) && (G3 >= 1023))
33     {
34         E1 = "E1/ Tensão de alimentação do Kit desregulada, menor que 96% e maior que 10%; "
35         Erro1 = 0
36     }
37 else

```

```
38     {
39         E1 = ""
40         Erro1 = 1
41     }
42
43 //
44
45 // 4)Teste se o jumper J1 está aberto, sua tensão tem que estar próxima de G5, com tolerância de 5% a
46 //     mais;
47 //     (verificar se G2,G1 e G0 está menor/igual do que 5% de GND (0) e menor/igual que G3 (410)
48 if ((G0 <= 410) && (G1 <= 410) && (G2 <= 410))
49     {
50         E2 = "E2/ Possivelmente, o jumper1 interligado aos terminais 9V e G0 não esteja bem conectado
51         refaça de novo a conexão; "
52         Erro2 = 0
53     }
54 else
55     {
56         E2 = ""
57         Erro2 = 1
58     }
59 //
60
61 // 5)Teste se o jumper J2 está aberto, a tensão de G2 tem que estar próxima de G3, 9V, com tolerância
62 //     de 5% a menos;
63 //     (verificar se G2,G1 e G0 está maior/igual do que 5% de 9 V (3925)).
64 if ((G0 >= 3925) && (G1 >= 3925) && (G2 >= 3925))
65     {
66         E3 = "E3/ Possivelmente, o jumper2 interligado aos terminais G2 e GND não esteja bem
67         conectado refaça de novo a conexão; "
68         Erro3 = 0
69     }
70 else
71     {
72         E3 = ""
73         Erro3 = 1
74     }
75 //
76
77 // 6)Teste se a resistência R1 está em curto, a tensão de G0 e G1 tem que estar próxima de 9V, com
78 //     tolerância de 5% a menos;
79 //     (verificar se G1 e G0 está maior/igual do que 5% de 9 V (3925)
80 if ((G1 >= 3925) && (G0 >= 3925))
81     {
82         E4 = "E4/ Parece que o resistor R1 está em curto verifique a ligação; "
83         Erro4 = 0
84     }
85 else
```

```
86     {
87         E4 = ""
88         Erro4 = 1
89     }
90
91 //
92
93 // 7)Teste se a resistência R2 está em curto, a tensão de G1 tem que estar próxima de 0V, com tolerância de 5% a menos;
94 // ((verificar se G1 e G2 está menor/igual do que 5% de GND (205)
95
96 if ((G1 <= 205) && (G2 <= 205))
97     {
98         E5 = " E5/ Parece que o resistor R2 está em curto verifique a ligação; "
99         Erro5 = 0
100    }
101 else
102     {
103         E5 = ""
104         Erro5 = 1
105     }
106
107
108 //
109
110 // 8)Teste se a resistência R1 está em aberto, a tensão de G1 e G2 tem que estar próxima de GND, com tolerância de 5% a menos
111 // e G0 se está maior/igual do que 5% de 9V (3925) ;
112
113 if ((G1 <= 205) && (G2 <= 205) && (G0 >= 3925))
114     {
115         E6 = "E6/ Parece que o resistor R1 está em aberto verifique a ligação; "
116         Erro6 = 0
117     }
118 else
119     {
120         E6 = ""
121         Erro6 = 1
122     }
123
124 //
125
126 // 9)Teste se a resistência R2 está em aberto, a tensão de G0 e G1 tem que estar próxima de 9V, com tolerância de 5% a menos,
127 // e G2 se está menor/igual do que 5% de GND (205) ;
128
129 if ((G1 >= 3925) && (G0 >= 3925) && (G2 <= 205))
130     {
131         E7 = "E7/ Parece que o resistor R2 está em aberto verifique a ligação; "
132         Erro7 = 0
133     }
134 else
135     {
```

```
136     E7 = ""
137     Erro7 = 1
138   }
139
140 //
141
142 // 10) Teste se os resistores R1 e R2 estão invertidos. Possivelmente, existe um erro na conexão e
143 // montagem entre os
144 // resistores R1 e o R2 tente invertê-los; " Resistores trocados, logo VR1 será VR2 e vice versa
145 VR1 = (G0-G1)*((R1Medido)/(R1Medido + R2Medido))
146 VR2 = (G1-G2)*((R2Medido)/(R1Medido + R2Medido))
147 VR1Max = VR1*1.15
148 VR1Min = VR1*0.85
149 VR2Max = VR2*1.15
150 VR2Min = VR2*0.85
151
152 if ((VR1Max > VR2) && (VR2 >= VR1Min) && (VR2Max > VR1) && (VR1 >= VR2Min))
153   {
154     E8 = " E8/ Possivelmente, existe um erro na conexão e montagem entre os resistores R1 e o R2
155     tente invertê-los; "
156     Erro8 = 0
157   }
158 else
159   {
160     E8 = ""
161     Erro8 = 1
162   }
163 //
164
165 // 11) Teste se algum dos resistores R1 e R2 apresenta valor diferente do que foi sorteado/medido
166 // anteriormente
167 //ou pode ter sido trocado. Verifica se G1 está mais/menos 10%
168 G1Est = (R2Medido/(R1Medido+R2Medido))*4095
169 G1Max = G1*1.10
170 G1Min = G1*0,90
171
172 if ((G1Max >= G1Est) && (G1Est >= G1Min))
173   {
174     E9 = ""
175     Erro9 = 1
176   }
177 else
178   {
179     E9 = "E9/ Algum dos resistores apresenta valor diferente do que foi medido anteriormente ou
180     pode ter sido trocado; "
181     Erro9 = 0
182   }
183 //
184 // 12) A montagem está correta e a atividade poderá prosseguir."
```

```
185 if ((Erro0) &&(Erro1) && (Erro2) && (Erro3) && (Erro4) && (Erro5) && (Erro6) && (Erro7) && (Erro8) &&
    (Erro9))
186     {
187         E10 = "E10/ A montagem está correta e a atividade poderá prosseguir."
188         Button_Proseguir.Show()
189         Yellow1.Show()
190     }
191 else
192     {
193         E10 = ""
194         Button_Proseguir.Hide()
195         Yellow1.Hide()
196     }
```

É possível ver que o javascript da cama de pregos faz 10(dez) testes que podem gerar erros conforme a sua resposta. Se nenhum erro for obtido o programa pode avançar indicado que a montagem foi concluída com sucesso.

6.2 CÓDIGO DE ENCRIPTAÇÃO DO RESULTADO

O programa de encriptação cria dois números que são as senhas para uso no [LMS](#). Para gerar estas senhas foi usado um sistema simples de encriptação substituindo o valor do carácter ASCII do usuário de autenticação do discente por números e efetuando uma multiplicação por um número previamente estabelecido gerando os dois campos mostrados. O código usado para encriptação está mostrado no [6.2](#). O [LMS](#) possui o mesmo algoritmo para efetuar a operação inversa e fazer a validação dos códigos.

```
1 Result = "ELE002" + Integrante[0].toUpperCase() + ";000"
2
3 K = Result.length
4 K = K - 6
5 Dezena = K/10
6 Dezena = Math.floor(Dezena)
7 Unidade = K - (Dezena*10)
8 Fator = Dezena+Unidade
9 i= 0
10
11 while (i < K)
12 {
13     ASCII1 = Result.charCodeAt(i)
14     ASCII1 = ASCII1*Dezena
15     i++
16
17     ASCII2 = Result.charCodeAt(i)
18     ASCII2 = ASCII2*Unidade
19     i++
20
21     ASCII3 = Result.charCodeAt(i)
22     ASCII3 = ASCII3*Fator
23     i++
24
25     ASCII4 = Result.charCodeAt(i)
26     ASCII4 = ASCII4+Dezena
27     i++
28
29     ASCII5 = Result.charCodeAt(i)
30     ASCII5 = ASCII5+Unidade
```

```
31 i++
32
33 ASCII6 = Result.charCodeAt(i)
34 ASCII6 = ASCII6+Fator
35 i++
36
37 ASCII = (ASCII1 + ASCII2 + ASCII3 + ASCII4 + ASCII5 + ASCII6 - Dezena)*Unidade
38 ASCIIITOTAL = ASCII + ASCII1 + ASCII2 + ASCII3 + ASCII4 + ASCII5 + ASCII6 + Fator
39 }
40 Codigo0 = "Box1= " + ASCII + " e Box2= " + ASCIIITOTAL
41 if (Integrante[0] == "") Codigo0 = " "
```

7 RESULTADOS OBTIDOS

Pode-se observar o uso do Kit fazendo a prática de resistores em serie está mostrado na Figura 33.

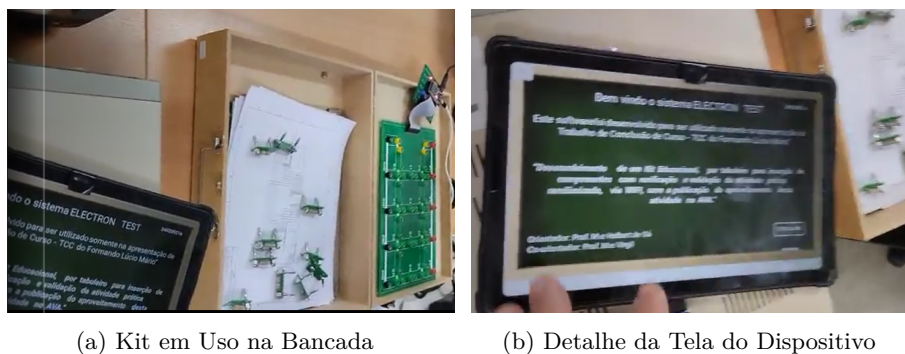


Figura 33 – Funcionamento do Kit na Bancada

Fonte: Autor

7.1 TELAS DO ARQUIVO SCORM CRIADO

As telas do arquivo criadas estão mostradas nesta subseção. Na Figura 34 vemos a tela de boas vindas e a de identificação dos alunos.

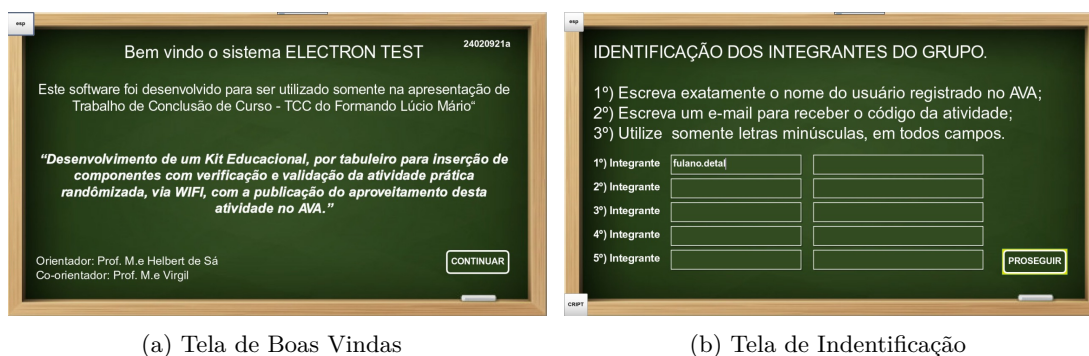
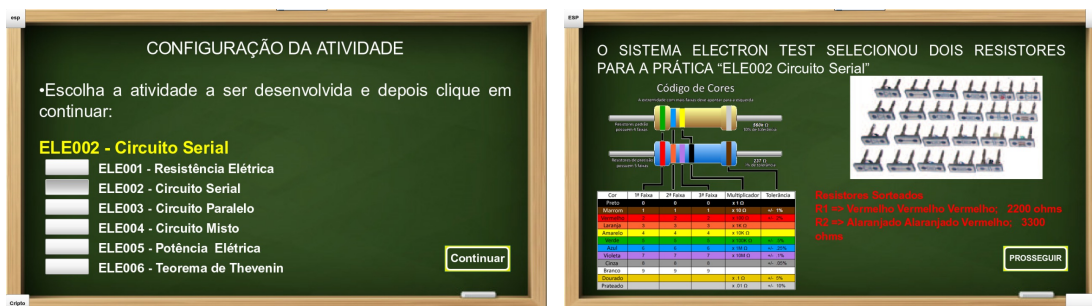


Figura 34 – Telas Iniciais

Fonte: Autor

Na Figura 35 vemos as telas de configuração da atividade e a tela inicial da atividade de circuito serial.



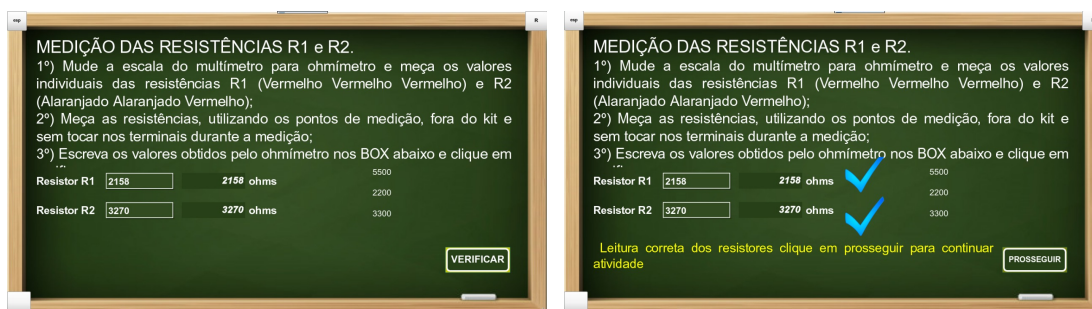
(a) Tela de Configuração da Atividade

(b) Tela da Inicial de Circuito Serial

Figura 35 – Telas de Configuração

Fonte: Autor

Na Figura 36 podemos ver as telas de instrução de medição dos resistores. Estes resistores são selecionados aleatoriamente dentro dos resistores que temos montados.



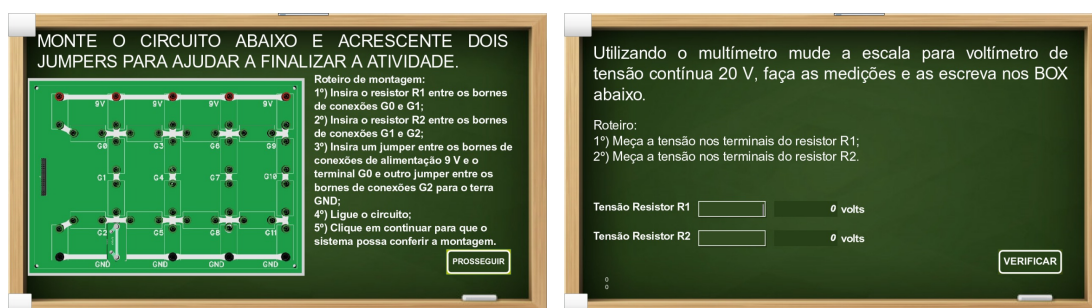
(a) Tela de Instrução de Medição de Resistores

(b) Tela do Resultado da Medição

Figura 36 – Telas de Medição de Resistores

Fonte: Autor

Na Figura 37 vemos a instrução de montagem e medição de tensão.



(a) Tela de Instrução de Montagem

(b) Tela de Instrução de Medição de Tensão

Figura 37 – Telas de Montagem e Medição de Tensão

Fonte: Autor

Na Figura 38 vemos a tela para medição da tensão e a tela do relatório da prática.

Este exemplo pode ser feito em SCORM ou HTML bastando para isto que o se configure as variáveis de acordo com o JSON obtido na leitura do Kit. Foi executado em SCORM neste trabalho apenas para demonstrar a aplicação direta em sistemas de gerenciamento de aprendizagem.



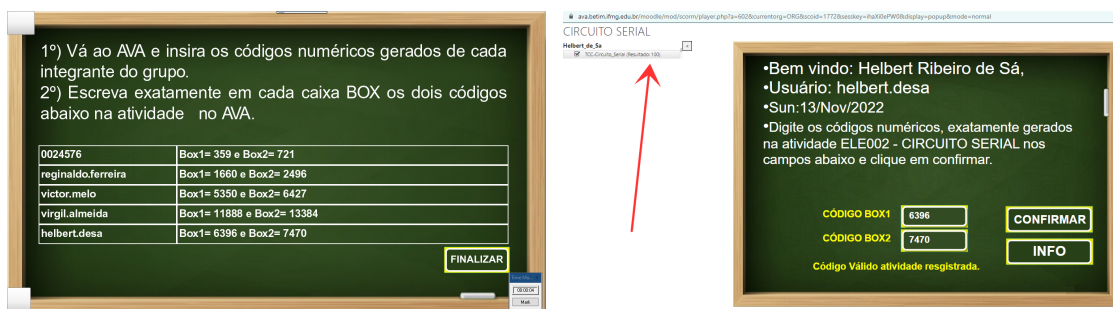
(a) Tela de Inserir Resultado de Tensão

(b) Tela do Relatório

Figura 38 – Telas Finais da Prática

Fonte: Autor

Após a tela do relatório, é gerado uma tela contendo dois números para inserir na atividade do Sistema de Gerenciamento de Aprendizagem LMS conforme mostrado na Figura 39.



(a) Tela com Códigos de Validação

(b) Tela de Inserção do Código de Validação

Figura 39 – Validação da Atividade no Sistema de Aprendizagem

Fonte: Autor

Esses números quando recebidos e validados instruem o LMS a marcar a atividade como concluída, como pode ser visto no relatório mostrado na Figura 40.

The screenshot shows a Moodle LMS report for 'CIRCUITO SERIAL'. The browser address bar indicates the URL: `ava.betim.ifmg.edu.br/moodle/mod/scorm/report.php?id=75681&mode=basic`. The page title is 'CIRCUITO SERIAL' and the user is logged in as 'Helbert Ribeiro de Sá'. The report is for 'Todos os participantes' (All participants) and shows '5 tentativas para usuários 8, fora de resultados 8' (5 attempts for 8 users, out of 8 results). The table below lists the participants and their performance.

	Nome / Sobrenome	Identificação de usuário	Endereço de email	Tentativa	Iniciado em	Último acesso em	Resultado	TCC- Circuito_Serial
<input type="checkbox"/>	Lucio Mario Almeida Lopes	0024756	[Redacted]	1	quarta, 16 Nov 2022, 15:01	quarta, 16 Nov 2022, 15:01	100	<input checked="" type="checkbox"/> 100
<input type="checkbox"/>	Helbert Ribeiro de Sá	helbert.desa	[Redacted]	1	domingo, 13 Nov 2022, 04:21	domingo, 13 Nov 2022, 04:26	100	<input checked="" type="checkbox"/> 100
<input type="checkbox"/>	Virgil Del Duca Almeida	virgil.almeida	[Redacted]	1	quinta, 17 Nov 2022, 14:45	quinta, 17 Nov 2022, 14:47	100	<input checked="" type="checkbox"/> 100
<input type="checkbox"/>	Reginaldo Vagner Ferreira	reginaldo.ferreira	[Redacted]	1	quinta, 17 Nov 2022, 13:08	quinta, 17 Nov 2022, 13:09	100	<input checked="" type="checkbox"/> 100
<input type="checkbox"/>	Victor Alves Silva e Melo	victor.melo	[Redacted]	1	sexta, 18 Nov 2022, 10:38	sexta, 18 Nov 2022, 10:39	100	<input checked="" type="checkbox"/> 100

Figura 40 – Relatório do LMS

Fonte: Autor

8 CONCLUSÃO

Com a aplicação deste trabalho nas aulas laboratoriais das disciplinas de eletricidade e eletrônica tem-se o ganho de obter a automação da correção da aula em tempo real. Ou seja, o discente recebe a informação imediatamente de qual o problema está ocorrendo com seu experimento.

Observou-se a possibilidade de usar o equipamento em uma sala presencial e também em ensino à distância. Pois os conteúdos ministrados ficam armazenados em um servidor que pode ser acessado pelo módulo por uma estrutura de rede local ou através da Internet.

Foi observado que a conversão analógica do ESP32 possui limitações que, para um trabalho com necessidade de uma maior precisão dos dados obtidos, torna necessário a utilização de um componente externo com melhor resposta. A discussão sobre a resposta ADC do ESP32 foi mostrada na seção 5.1.7 deste trabalho.

8.1 POSSÍVEIS TRABALHOS EM CONTINUAÇÃO DESTE

Este trabalho poderia ser dividido em dois ou três trabalhos menores, sendo uma parte hardware, outra software e ainda as integrações entre estes. Por ser um trabalho extenso, foi feita uma lista de itens com trabalhos poderiam ser sequencia deste.

- O desenvolvimento de mais práticas e módulos para aplicação destas. A prática demonstrada é o início de um curso de eletricidade básica e devem ser desenvolvidas outras práticas para fazer o funcionamento completo deste equipamento. Usando o módulo mostrado é possível fazer prática de resistores em série e em paralelo e um circuito misto puramente resistivo. Também é possível fazer práticas com semicondutores tipo diodos retificadores, Zeners ou LEDs.
- A confecção de módulos de testes de transistores e componentes mais elaborados pode ser feita e atestada pelo professor utilizando o circuito, apenas alterando a parte de conexão de circuitos em teste e criando o SCORM necessário para fazer a leitura e transmissão dos dados para o servidor do sistema de gerenciamento de aprendizagem correspondente.
- Melhorias no software, como a utilização de configuração de rede externamente para facilitar o seu uso em ambientes diversos.
- Melhoria no software seria a aplicação de mais informações no arquivo JSON, possibilitando a transmissão de validação e outros itens de segurança.
- Desenvolvimento de uma placa de leitura utilizando módulos ADC externos que propiciariam melhor resolução e mais canais de entrada, deixando o ESP32 apenas com o trabalho de comunicação de rede e com este periférico.
- Criação de placa de emissão de som para orientação a portadores de necessidades visuais especiais sobre a utilização do kit em conjunto com a utilização de programas específicos para a leitura das telas. Visando maior acessibilidade ao projeto.

Estas e a mostrada na subseção 8.2 foram as principais sequencias identificadas como possíveis para este trabalho.

8.2 POSSIBILIDADE DE UTILIZAÇÃO DO KIT EM TRABALHOS DE EXTENSÃO

Outro resultado obtido é a possibilidade de implementação de trabalhos para a comunidade com a utilização do kit, os chamados Trabalhos de Extensão. Foi feita uma subseção dedicada à este assunto por ser uma necessidade do Campus com possibilidade de ser atendida por este trabalho.

A resolução da Câmara de Educação Superior do Ministério da Educação ([SUPERIOR, CÂMARA DE EDUCAÇÃO, 2018](#))¹ exige que 10% da carga horaria dos cursos de graduação seja composta por atividades de extensão em seu artigo 4º e o artigo 7º da mesma resolução define que as atividades de extensão são as interações diretas da Instituição com a comunidade.

Desta forma, o Instituto pode promover um programa de incentivo tecnológico através de visitas de graduandos orientados e acompanhados por docentes à comunidade e instituições de ensino médio ou fundamental. Neste tipo de interação o Kit educacional deste trabalho pode ser utilizado para demonstrar práticas simples que podem despertar o interesse científico na comunidade fomentando a educação tecnológica. Esse projeto de extensão tem potencial de promover melhoria das condições sociais da comunidade através de promover o despertar de interesse pela Engenharia.

¹ Resolução 7 de 18 de dezembro de 2018 - CES/CNE/MEC

REFERÊNCIAS

- CASA DO RESISTOR. *Kit VU Bargraph*. 2022. Disponível em: <<https://www.casadoresistor.com.br/kit-vu-bargraph-potencia-dc-elz0111a-p4638>>. Acesso em: 15 nov. 2022. Citado na página 18.
- DELORENZO DO BRASIL. *Kit didático*. 2022. Disponível em: <<https://delorenzo.com.br/atuacao>>. Acesso em: 12 nov. 2022. Citado na página 19.
- ESPRESSIF SYSTEMS CO., LTD. *ESP32 Technical Reference Manual*. [S.l.], 2022. Version V4.7. Citado 2 vezes nas páginas 23 e 46.
- FERREIRA, A. L. *Como ler e obter dados de um arquivo JSON*. 2020. Disponível em: <<https://www.visual dicas.com.br/programacao/js/83-como-ler-e-obter-dados-de-um-arquivo-json-com-javascript>>. Acesso em: 13 out. 2022. Citado na página 31.
- GRANT, M.; CHANG, H. *Winning the Soc Revolution: Experiences in Real Design*. 1ª. ed. New York, NY, USA: Springer Science & Business Media, 2003. Citado na página 23.
- HTML,COM COMMUNITY. *HTML For Beginners The Easy Way: Start Learning HTML CSS Today*. 2021. Disponível em: <<https://html.com>>. Acesso em: 10 jul. 2022. Citado na página 23.
- JSON ORGANIZATION. *JSON*. 2022. Disponível em: <<https://www.json.org/json-en.html>>. Acesso em: 13 out. 2022. Citado na página 31.
- LUPPI, A. C. J. *Aplicação do modelo de atores em um programa de teste e diagnóstico de circuitos eletrônicos*. Monografia (TCC) — Engenharia Elétrica Universidade Federal de Santa Catarina, Florianópolis-SC, 2017. Citado na página 22.
- MINIPA DO BRASIL. *proto-board*. 2022. Disponível em: <<https://minipa.com.br/diversos/protoboards?search=&order=i.ordering&dir=asc&cm=0#tlb>>. Acesso em: 04 set. 2022. Citado na página 18.
- MOZILLA FOUNDATION. *JavaScript MDN*. 2022. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 13 out. 2022. Citado na página 31.
- PASINI, C. G. D.; CARVALHO, E. d.; ALMEIDA, L. H. C. A educação híbrida em tempos de pandemia: algumas considerações. *Observatório Socioeconômico da COVID-19 (OSE)*, v. 9, 2020. Citado na página 17.
- PRATA, P. Foto viral mostra crianças tendo aulas pelo rádio durante epidemia de polio em 1937. *Estado de São Paulo*, São Paulo-SP, 2021. Disponível em: <<https://politica.estadao.com.br/blogs/estadao-verifica/foto-viral-mostra-criancas-tendo-aulas-pelo-radio-por-cao-de-epidemia-de-polio-em-1937/>>. Acesso em: 20 de agosto de 2021. Citado na página 17.
- PRODUZA INTELIGÊNCIA EM MANUFATURA. *COMO UMA PANELIZAÇÃO BEM PLANEJADA PODE AFETAR POSITIVAMENTE A PRODUÇÃO DE UMA PCB*. 2022. Disponível em: <<https://produza.ind.br/tecnologia/pcb/>>. Acesso em: 02 dez. 2022. Citado na página 39.
- REIS, V. de L. Conversor configurável analógico para informação. 2017. Citado na página 46.
- RUSTICI SOFTWARE SCORM. *Scorm Technical Documentation*. 2022. Disponível em: <https://rusticisoftware.com/technical-documentation/?_ga=2.171317988.597200290.1653331761-1099360493.1653331761>. Acesso em: 01 out. 2022. Citado na página 26.
- SHENZHEN AI-THINKER TECHNOLOGY CO., LTD. *Nodemcu-32s Datasheet*. [S.l.], 2019. Version V1. Citado na página 34.
- SHENZHEN SUNSOAR CIRCUIT TECHNOLOGY. *O que é a cama de pregos Tester para Pcb*. 2022. Disponível em: <<http://pt.oem-pcb.com/news/what-is-bed-of-nails-tester-for-pcb-10745631.html>>. Acesso em: 26 nov. 2022. Citado na página 22.

SUPERIOR, CÂMARA DE EDUCAÇÃO. Resolução nº 7, de 18 de dezembro de 2018. *Estabelece as*, 2018. Disponível em: <https://normativasconselhos.mec.gov.br/normativa/view/CNE_RES_CNECESN72018.pdf>. Acesso em: 10 nov. 2022. Citado na página 58.

Vlasenko, L.; BOZHOK, N. Advantages and disadvantages of distance learning. *Modern trends and resources for teachers of foreign languages*, v. 1, n. 1, p. 20–25, 2014. Citado 3 vezes nas páginas 17, 21 e 23.

ZERYNTH CO., LTD. *DOIT DevKit-V1 Datasheet*. [S.l.], 2019. Version V1. Citado na página 23.

Apêndices

APÊNDICE A – CARACTERÍSTICAS DO ESP32

A.1 ESP32 – CARACTERÍSTICAS

A.1.1 PROCESSAMENTO

O esp32 é um módulo compacto e poderoso que possui o LX6 da Xtensa® que é um microprocessador dual-core de 32-bit com até 600 MIPS (200 MIPS para o ESP32-S0WD, 400 MIPS para o ESP32-D2WD). A memória deste módulo conta com 448KB de ROM, 520KB de SRAM e 16KB SRAM in RTC e suporte a múltiplos chips flash/SRAM(QSPI).

O oscilador interno é de 8MHz com calibração. Possui também um oscilador RC interno com calibração. Cristais osciladores externos de 2 a 60MHz e um oscilador de 40MHz para o Wifi e o BT. Possui também dois grupos de timers de 64bits e um watchdog em cada grupo de timer. Conta também com um RTC e um watchdog para o RTC.

A.1.2 ENTRADAS E SAÍDAS

O módulo possui 34 entradas/saídas (GPIO) que podem ser configuradas conforme abaixo:

- 12-bit SAR ADC – até 18 entradas capazes de ler sinais analógicos com resolução de 12bits.
- 8-bit DAC - até 2 saídas capazes de enviar sinais analógicos com resolução de 8 bits.
- Touch sensors – até 10 sensores capacitivos facilitando a confecção de teclas de acionamento.
- SPI – até 4 interfaces serial para comunicação com dispositivos periféricos.
- I2S – até 2 interfaces com protocolo I2S usado para comunicação de audio digital.
- I2C – até 2 interfaces com protocolo I2C, que é um barramento serial desenvolvido pela Philips® que funciona como multi-mestre e mestre-escravo.
- UART – até 3 interfaces universais de recepção e transmissão assíncrona.
- Host (SD/eMMC/SDIO) – uma entrada para leitor de cartão SD/eMMC/SDIO.
- Escravo(SDIO/SPI) – pode funcionar com uma entrada no modo escravo SDIO/SPI.
- Ethernet MAC interface com DMA dedicado e suporte a IEEE 1588.
- CAN 2.0 – protocolo de rede desenvolvido pela Bosch®, usado em veículos e na indústria através do OpenCAN.
- IR (TX/RX) – protocolo para transmissão e recepção de sinais através do infra-vermelho.
- PWM – controlador de modulação por largura de pulso, usada para controle de transferência de potência com tensão fixa.
- Sensor Hall – sensor que varia a tensão conforme a variação de campo magnético.

A.1.3 SEGURANÇA

Com o ESP32 é possível implementar algumas características de segurança. As características de segurança do ESP32 são:

- Secure boot.
- Possibilidade de encriptação do Flash.
- 1Mbit OTP, com até 768bits por clientes.
- Criptografia acelerada por hardware com algoritmos AES, Hash(SHA-2), RSA e ECC.
- Gerador de números aleatórios (RNG).

A.1.4 DIAGRAMA DE BLOCOS

O diagrama de blocos mostrado na Figura 41 resume as características do esp32 :

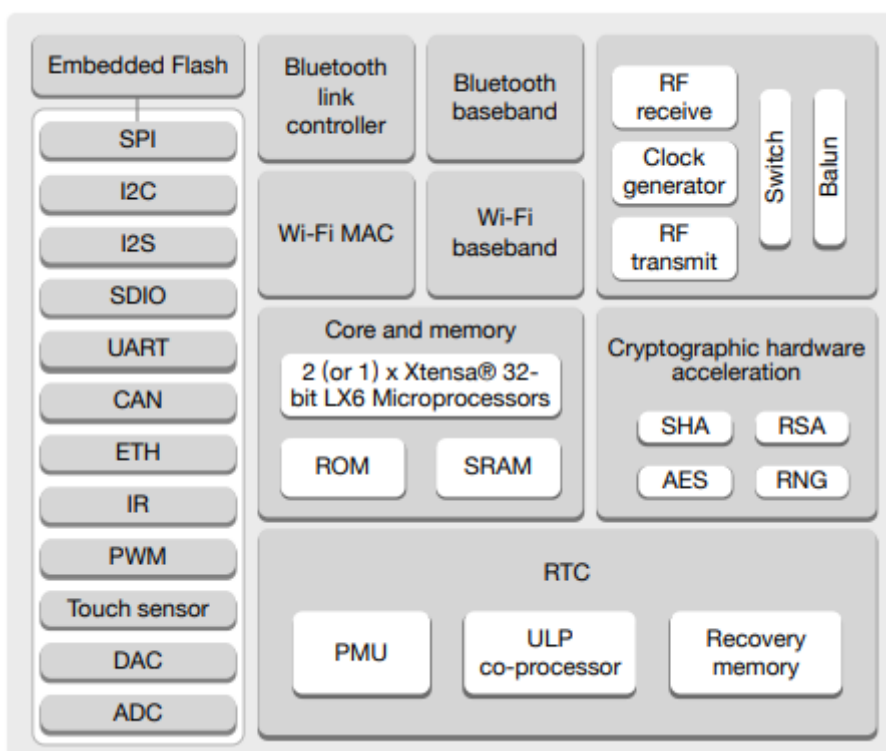


Figura 41 – Características ESP32

Fonte: Espressif®

A.1.5 DOCUMENTAÇÃO DO ESP32

As notas de lançamento da documentação do Esp32 estão mostradas na tabela [1] :

Tabela 6 – Notas de Lançamento do ESP32

Data	Versão	Lançamento
06/2017	1.0	Lançamento do Esp32
08/2017	1.1	ESP32-PICO-D4.
08/2017	1.2	Retirado o ESP8689. Atualização do SPQ e MOQ para ESP32-PICO-D4
09/2017	1.3	Atualizados: ESP32-D0WD e ESP32-D2WD. ESP-WROOM-02D, ESP-WROOM-32D e ESP32-WROOM-32U
11/2017	1.4	ESP32-PICO-KIT, ESP-WROOM-02D e ESP-WROOM-02U. Atualizados os SPQ e MOQ.
03/2018	1.6	Renomeados: ESP-WROOM-32 and ESP-WROOM-32D
06/2018	1.7	Datasheets: ESP32-DevKitC, ESP32-WROVER(PSRAM) e ESP32-WROVER-I(PSRAM). ESP32-SOLO-1, ESP32-LyraT, ESP32-LyraTD-MS, ESP32-Sense Kit, ESP32-Sense Kit, e ESP-Prog.
06/2018	1.8	Datasheet do ESP32-SOLO-1 e lançamento dos ESP32-WROVER-B and ESP32-WROVER-IB
06/2018	1.8	Datasheets atualizados ESP32-PICO-D4, ESP32-LyraT, ESP32-LyraTD-MS, ESP32-Sense Kit, ESP-Prog, ESP32-WROVER-B, and ESP32-WROVER-IB
07/2018	1.9	ESP32-MeshKit-Sense and ESP32-MeshKit-Light.
08/2018	2.0	Atualizadas as informações de dimensões dos módulos e número dos produtos e a capa dos documentos.

Fonte: Espressif®.

Nesta tabela podemos perceber que o ESP32 é um módulo bem recente e sua documentação está em constante atualização pela fabricante Espressif®. O ESP32 surgiu como uma atualização do ESP8266 que possui apenas um core e menos memória.

APÊNDICE B – INSTALANDO E CONFIGURANDO A INTERFACE DO ARDUINO

B.1 INTERFACE DO ARDUINO:

A instalação da interface do Arduino pode ser feita no ubuntu/debian abrindo um terminal e digitando “sudo apt-get update | sudo apt-get install arduino”. No Windows pode ser feita a instalação seguindo os passos:

1. Baixe a versão mais atual do programa compatível com sua versão do sistema windows no sitio do Arduino ¹.

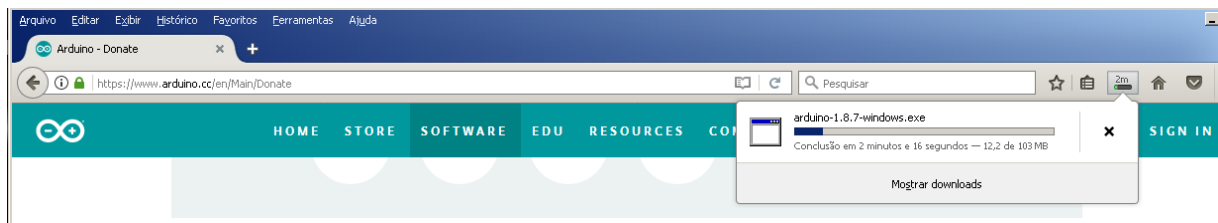


Figura 42 – Baixando a Interface do Arduino

Fonte: Autor

2. Execute o arquivo de instalação baixado e aceite os termos dando um click em “I Agree”:

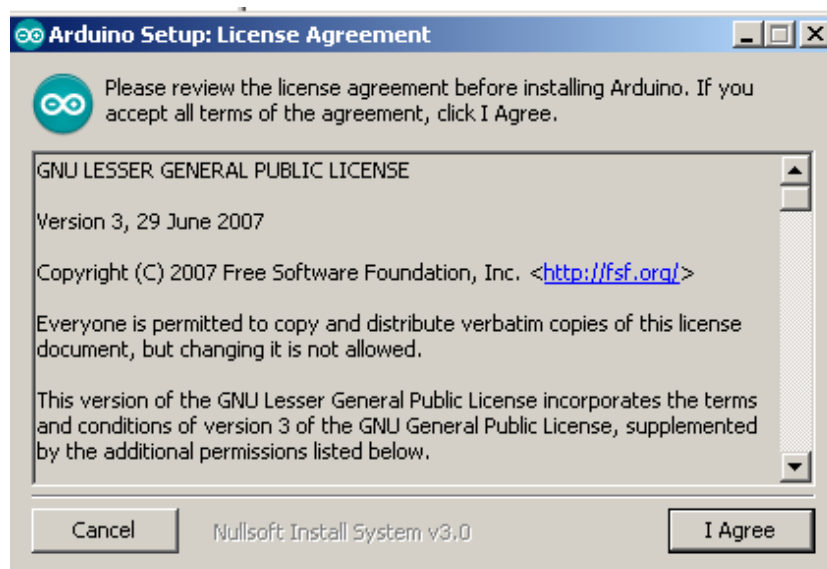


Figura 43 – Executando o arquivo

Fonte: Autor

¹ <<https://www.arduino.cc/en/Main/>>

3. Click em “Next” e aguarde até o fim da instalação e click em close:

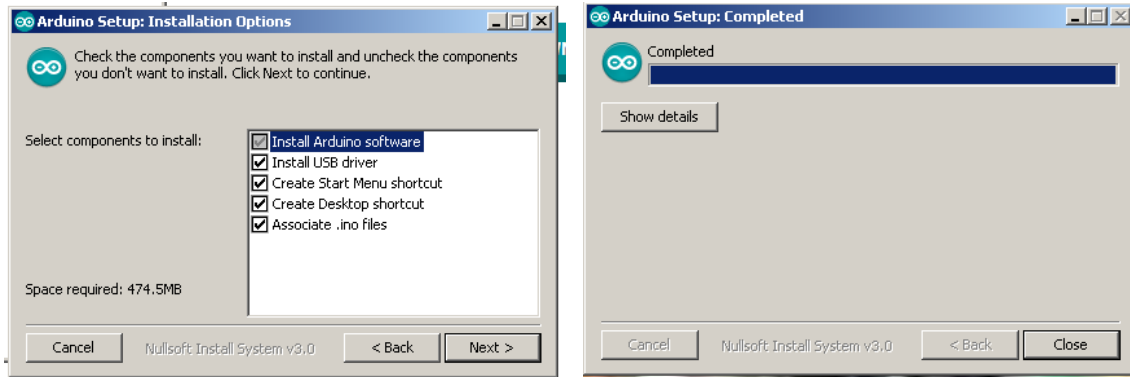


Figura 44 – Instalação Padrão do Arduino

Fonte: Autor

4. A interface do Arduino já está instalada na máquina.

Neste ponto já podemos utilizar a interface do Arduino para construir programas. Mas para a utilização da Interface do Arduino com o ESP32 vamos necessitar de instalar as placas do ESP32 na interface. A instalação da placa do ESP32 na interface será mostrada na subseção B.2

B.2 ADICIONANDO AS PLACAS DO ESP32 NA IDE DO ARDUINO

Agora é necessário instalar o ESP32 na IDE do Arduino. Para isto, no linux ou no windows, os seguintes passos são necessários:

1. Abra a IDE do Arduino e click em “Arquivo” e seguindo ao menu Preferências. No menu Preferências preencha o campo URLs adicionais para Gerenciadores de Placa com no sitio da Espressif² E feche a IDE do Arduino.

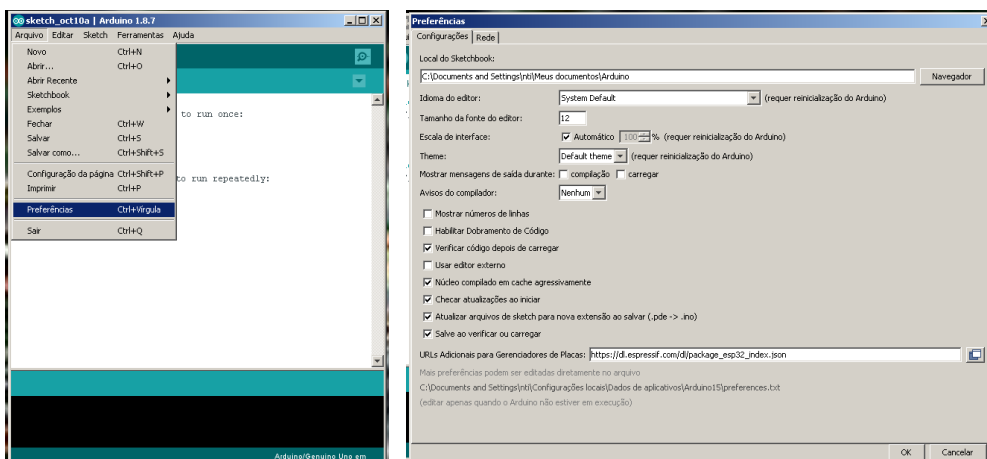


Figura 45 – Configurando o repositório de Placas

Fonte: Autor

² <https://dl.espressif.com/dl/package_esp32_index.json>

- Abra novamente a IDE do Arduino (é necessário fechar e reabrir para carregar as preferências). E na aba “Ferramentas” siga até o menu “Placa” e click em “Gerenciador de Placas”. No “Gerenciador de Placas” filtre as saídas digitando “esp32”. Selecione o pacote e click em instalar:

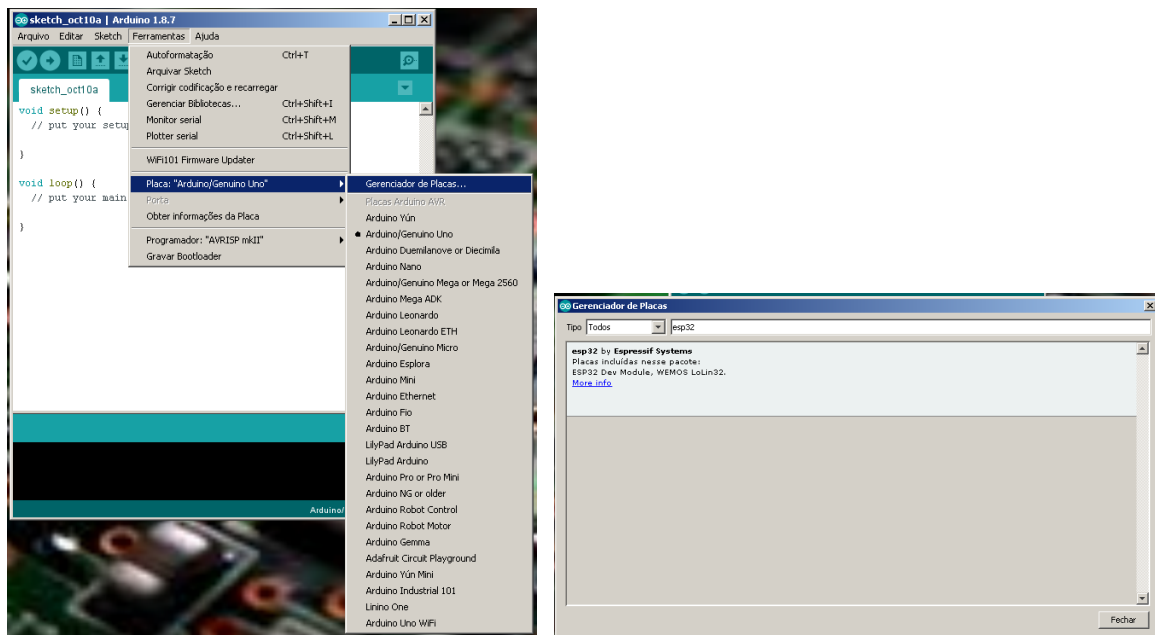


Figura 46 – Visualizando as Placas do ESP32

Fonte: Autor

- Após a instalação click em “Gerenciador de Placas” e você terá a lista de novas placas instaladas na IDE. Selecione a placa que você está usando e já poderá executar testes nesta placa.

B.3 INSTALANDO O DRIVER DE COMUNICAÇÃO

Geralmente o driver é reconhecido pelo sistema operacional do computador. Caso isso não ocorra siga os passos:

- Navegue até o sitio da Silicom Labs³ e baixe o driver adequado ao seu sistema operacional:
- Descompacte e execute o driver baixado:
- Agora já está tudo pronto para a operação do ESP32 com a IDE do Arduino.

³ <<https://silabs.com>>

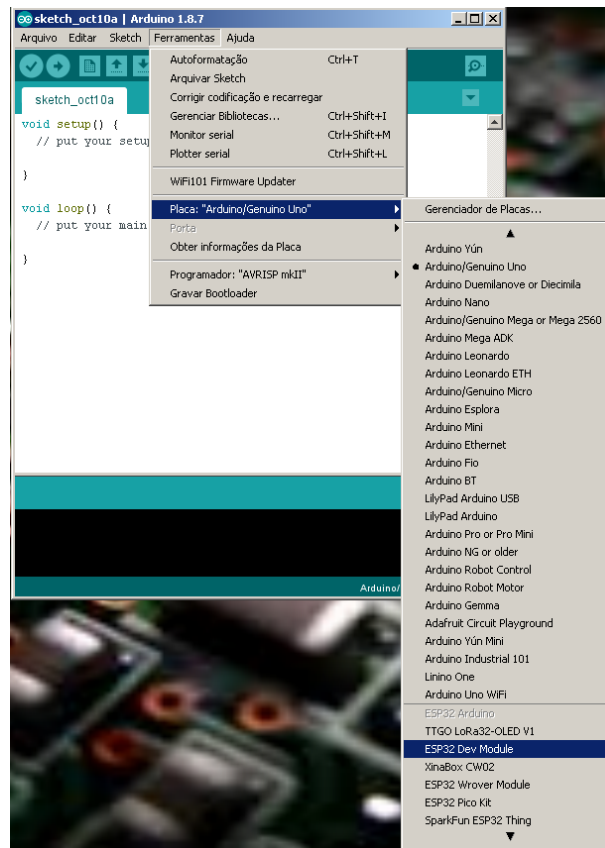


Figura 47 – Buscando as Placas do ESP32

Fonte: Autor



Figura 48 – Baixando o driver CP2102

Fonte: Autor

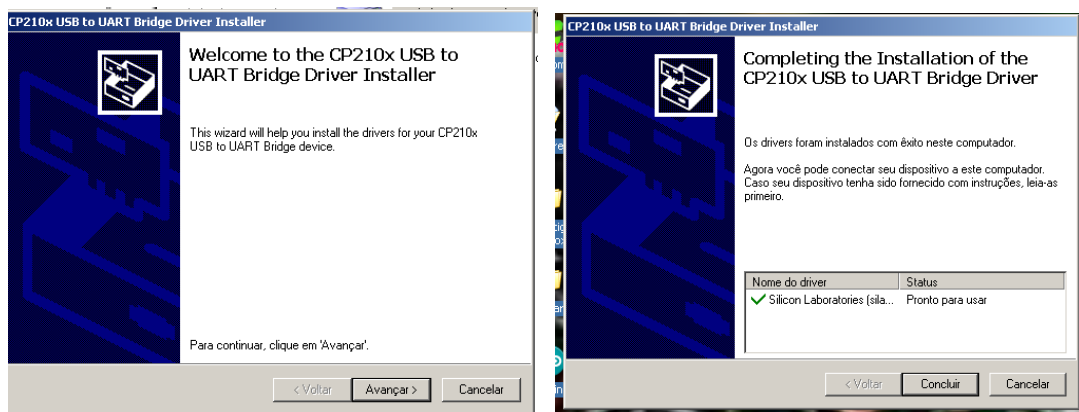


Figura 49 – Executando a Instalação do Driver

Fonte: Autor

APÊNDICE C – CÓDIGO DO KIT

O programa foi dividido em 6(seis) partes para facilitar sua compreensão e alterações posteriores.

O arquivo TCC.INO no qual está a descrição do programa está mostrado abaixo:

```

1  /*
2  Este programa foi criado para o trabalho de conclusão de curso de Bacharel em Engenharia de Controle
   e Automação.
3  Instituto Federal de Minas Gerais - Campus Betim.
4  Aluno Lúcio Mário Almeida Lopes.
5  O programa de Scorm faz a leitura do módulo através de GetWebData no endereço http://192.168.4.1/ na
   porta 80.
6
7  Os dados obtidos são:
8  GET /H   Liga o LED da placa para teste de conexão.
9  GET /L   Desliga o LED da placa.
10 GET /G0  Faz a leitura analógica do GPIO39
11 GET /G1  Faz a leitura analógica do GPIO36
12 GET /G2  Faz a leitura analógica do GPIO34
13 GET /G3  Faz a leitura analógica do GPIO35
14 GET /G4  Faz a leitura analógica do GPIO33
15 GET /G5  Faz a leitura analógica do GPIO32
16 GET /G6  Faz a leitura digital do GPIO26
17 GET /G7  Faz a leitura digital do GPIO25
18
19
20 */

```

O arquivo Bibliotecas.ino no qual estão as bibliotecas do está mostrado a seguir:

```

1 #include <WiFi.h>           //necessário para usar o WiFi.
2 #include <WiFiClient.h>    //necessário para aceitar clientes Webserver.
3 #include <WiFiAP.h>        //necessário para usar o modo Access Point.

```

O arquivo Objetos está mostrado a seguir:

```

1 // Configurando a rede no modo AP.
2 const char *ssid = "TCC-LUCIOLOPES";
3 const char *password = "24112019"; //a data da primeira prova de ENADE que o IFMG-Betim participou,
   somos nota 5.
4 //iniciando o servidor na porta 80
5 WiFiServer server(80);

```

O arquivo Pinos.ino no qual estão as definições dos pinos está mostrado a seguir:

```

1 #define LED_02 2           // Definindo os pinos usados no projeto.
2 #define GPIO36 36
3 #define GPIO39 39
4 #define GPIO34 34
5 #define GPIO35 35
6 #define GPIO32 32
7 #define GPIO33 33
8 #define GPIO25 25
9 #define GPIO26 26

```

A seguir temos o arquivo setup.ino que contem o setup do programa.

```

1 void setup()
2 {
3   pinMode(LED_02, OUTPUT); //configurando o modo dos pinos.
4   pinMode(GPI036, INPUT);
5   pinMode(GPI039, INPUT);
6   pinMode(GPI034, INPUT);
7   pinMode(GPI035, INPUT);
8   pinMode(GPI032, INPUT);
9   pinMode(GPI033, INPUT);
10  pinMode(GPI025, INPUT);
11  pinMode(GPI026, INPUT);
12  //
13
14  Serial.begin(115200);
15  Serial.println();
16  Serial.println("Configurando o Access Point");
17
18  // Iniciando o Wifi no modo Access Point
19  WiFi.softAP(ssid, password);
20  IPAddress myIP = WiFi.softAPIP();
21  Serial.print("Endereco IP do Access Point: ");
22  Serial.println(myIP);
23  server.begin();
24
25  Serial.println("Servidor Iniciado");
26 }

```

E a seguir está o looping principal do programa.

```

1 void loop() {
2   WiFiClient client = server.available(); // aguarda clientes
3
4   if (client) { // Entrando cliente, escreve na porta serial e cria uma
5     String "currentLine" para armazenar dados entrando.
6     Serial.println("Novo Cliente");
7     String currentLine = "";
8     while (client.connected()) { // Looping de conexao e leitura das requisicoes dos
9       clientes
10      if (client.available()) {
11        char c = client.read();
12        Serial.write(c);
13        if (c == '\n')
14        {
15          if (currentLine.length() == 0)
16          {
17            // O final do HTTP responde com outra linha em branco:
18            client.println();
19            // Sai do loop
20            break;
21          } else { // se houver nova linha, apaga a variável currentLine:
22            currentLine = "";
23          }
24        } else if (c != '\r') { // adiciona à variável currentLine o que for recebido
25          currentLine += c; // add it to the end of the currentLine
26        }
27
28        // Check to see if the client request was "GET /H" or "GET /L":

```

```

27     if (currentLine.endsWith("GET /H")) {
28         digitalWrite(LED_02, HIGH);           // GET /H Liga o LED
29     }
30     if (currentLine.endsWith("GET /L")) {
31         digitalWrite(LED_02, LOW);           // GET /L Apaga o LED
32     }
33     if (currentLine.endsWith("GET /G0")) {
34         int entrada = analogRead(GPI036);    // GET /G0 Ler o Valor de GPI036
35         client.print(entrada);
36     }
37     if (currentLine.endsWith("GET /G1")) {
38         int entrada = analogRead(GPI039);    // GET /G1 Ler o Valor de GPI039
39         client.print(entrada);
40     }
41     if (currentLine.endsWith("GET /G2")) {
42         int entrada = analogRead(GPI034);    // GET /G2 Ler o Valor de GPI034
43         client.print(entrada);
44     }
45     if (currentLine.endsWith("GET /G3")) {
46         int entrada = analogRead(GPI035);    // GET /G3 Ler o Valor de GPI035
47         client.print(entrada);
48     }
49     if (currentLine.endsWith("GET /G4")) {
50         int entrada = analogRead(GPI032);    // GET /G4 Ler o Valor de GPI032
51         client.print(entrada);
52     }
53     if (currentLine.endsWith("GET /G5")) {
54         int entrada = analogRead(GPI033);    // GET /G5 Ler o Valor de GPI033
55         client.print(entrada);
56     }
57     if (currentLine.endsWith("GET /G6")) {
58         int entrada = digitalRead(GPI026);    // GET /G6 Ler o Valor de GPI025
59         client.print(entrada);
60     }
61     if (currentLine.endsWith("GET /G7")) {
62         int entrada = digitalRead(GPI025);    // GET /G7 Ler o Valor de GPI026
63         client.print(entrada);
64     }
65     if (currentLine.endsWith("GET /tudo")) {
66         String json = fazjson();              // gera o json
67         client.print("Content-Type");
68         client.print(",");
69         client.print("application/json");
70         client.print(json);
71     }
72 }
73 }
74 // close the connection:
75 client.stop();
76 Serial.println("Cliente Desconectado.");
77 }
78 }
79
80 String fazjson(){
81     int entg0 = analogRead(GPI036);
82     int entg1 = analogRead(GPI039);
83     int entg2 = analogRead(GPI034);
84     int entg3 = analogRead(GPI035);
85     int entg4 = analogRead(GPI032);

```

```
86  int entg5 = analogRead(GPI033);
87  bool entg6 = digitalRead(GPI026);
88  bool entg7 = digitalRead(GPI025);
89  String resultado = "{";
90  resultado = resultado + "\"G0"+"\":\":"+entg0+"\",";
91  resultado = resultado + "\"G1"+"\":\":"+entg1+"\",";
92  resultado = resultado + "\"G2"+"\":\":"+entg2+"\",";
93  resultado = resultado + "\"G3"+"\":\":"+entg3+"\",";
94  resultado = resultado + "\"G4"+"\":\":"+entg4+"\",";
95  resultado = resultado + "\"G5"+"\":\":"+entg5+"\",";
96  resultado = resultado + "\"G6"+"\":\":"+entg6+"\",";
97  resultado = resultado + "\"G7"+"\":\":"+entg7+"\",";
98  resultado = resultado + "}";
99  return resultado;
100 }
```

APÊNDICE D – FERRAMENTA ONLINE PARA FAZER SCORM - ISEAZY®

Podemos gerar **SCORM** através de ferramentas online do tipo **WYSIWYG** que permitem a publicação do conteúdo gerado em **SCORM** e utilizá-lo. Para efeito de facilitar o entendimento vou demonstrar a forma rápida de gerar um conteúdo **SCORM** em uma ferramenta online. A ferramenta escolhida foi o ISEAZY®¹ que possui um acesso gratuito que permite a criação de conteúdo **SCORM**.

Após um breve cadastro na ferramenta pode se gerar cursos de forma totalmente visual, dispensando habilidades de programação. O ISEAZY® conta ainda com vários modelos que podem ser usados conforme a necessidade e vontade. A Figura 50 mostra o painel com os projetos criados dentro da ferramenta.

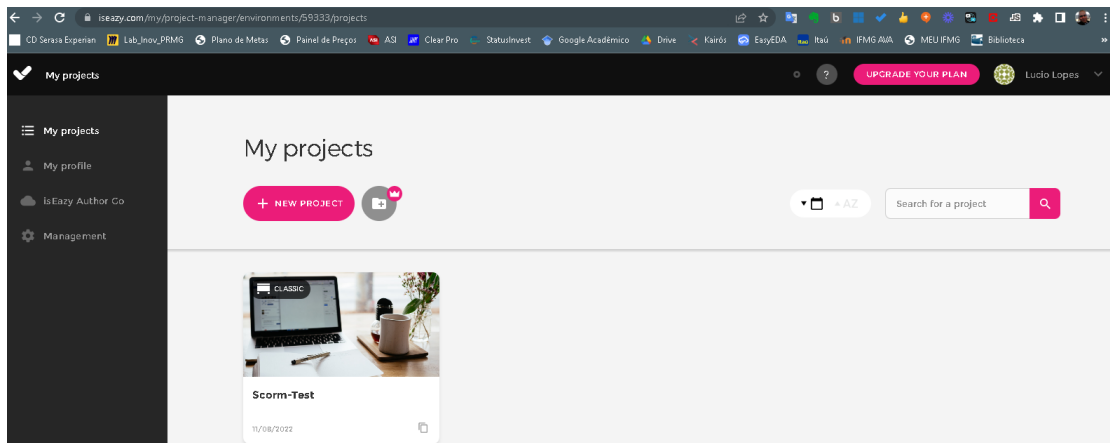


Figura 50 – Painel de Projetos no ISEAZY

Fonte: Autor

Selecionando o projeto desejado será mostrado um cardápio de tarefas à esquerda conforme a Figura 51.

Selecionando a opção "Distribution" aparece a tela que permite a distribuição do código conforme a Figura 52.

Selecionando a opção "LMS" aparece a tela para tela de opções de publicação SCORM conforme a Figura 53.

Fazendo as escolhas entre a versão do SCORM e a monitoramento de progressão, basta acionar "Publish" para publicar e possibilitar o "DOWNLOAD" e utilização do código gerado conforme mostrado na Figura 54.

¹ <<https://iseazy.com>>

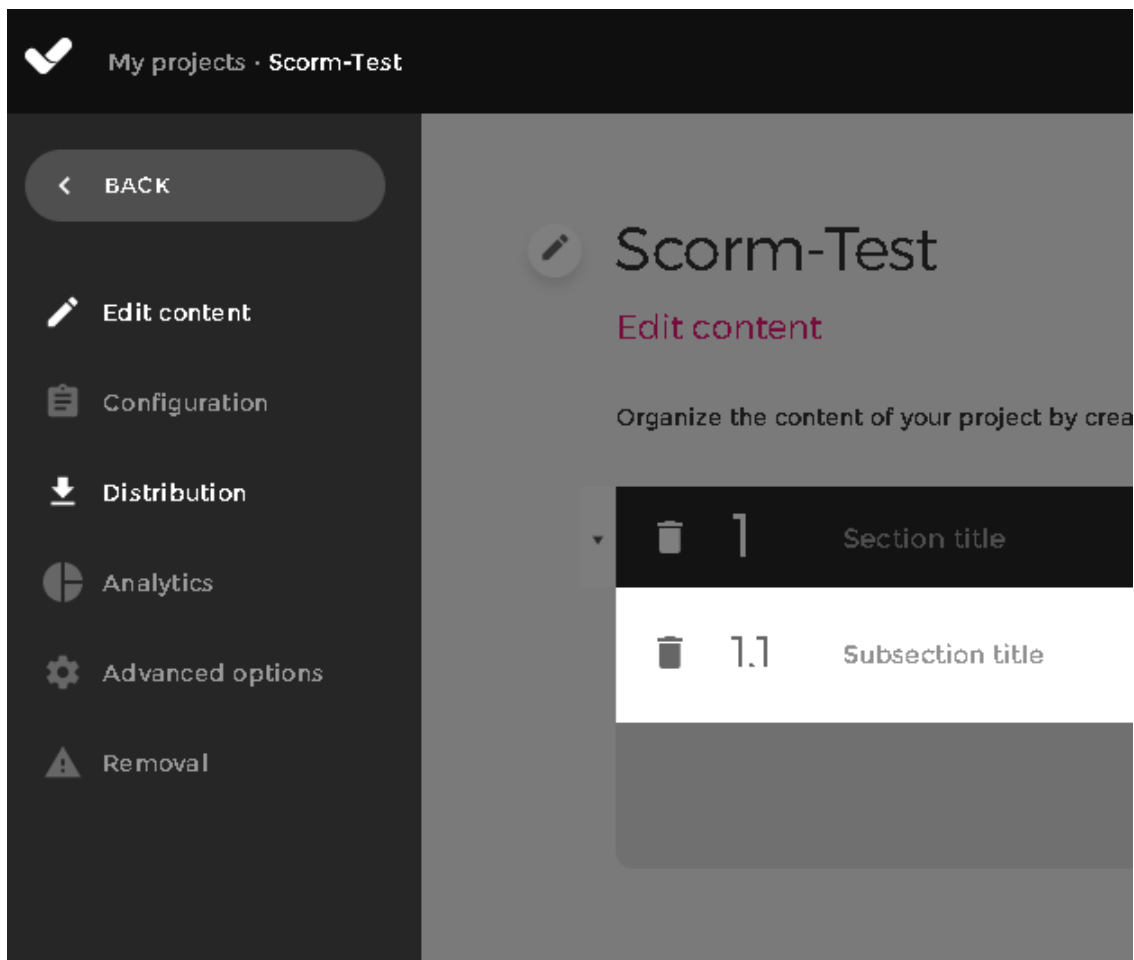


Figura 51 – Menu do Projeto no ISEAZY

Fonte: Autor

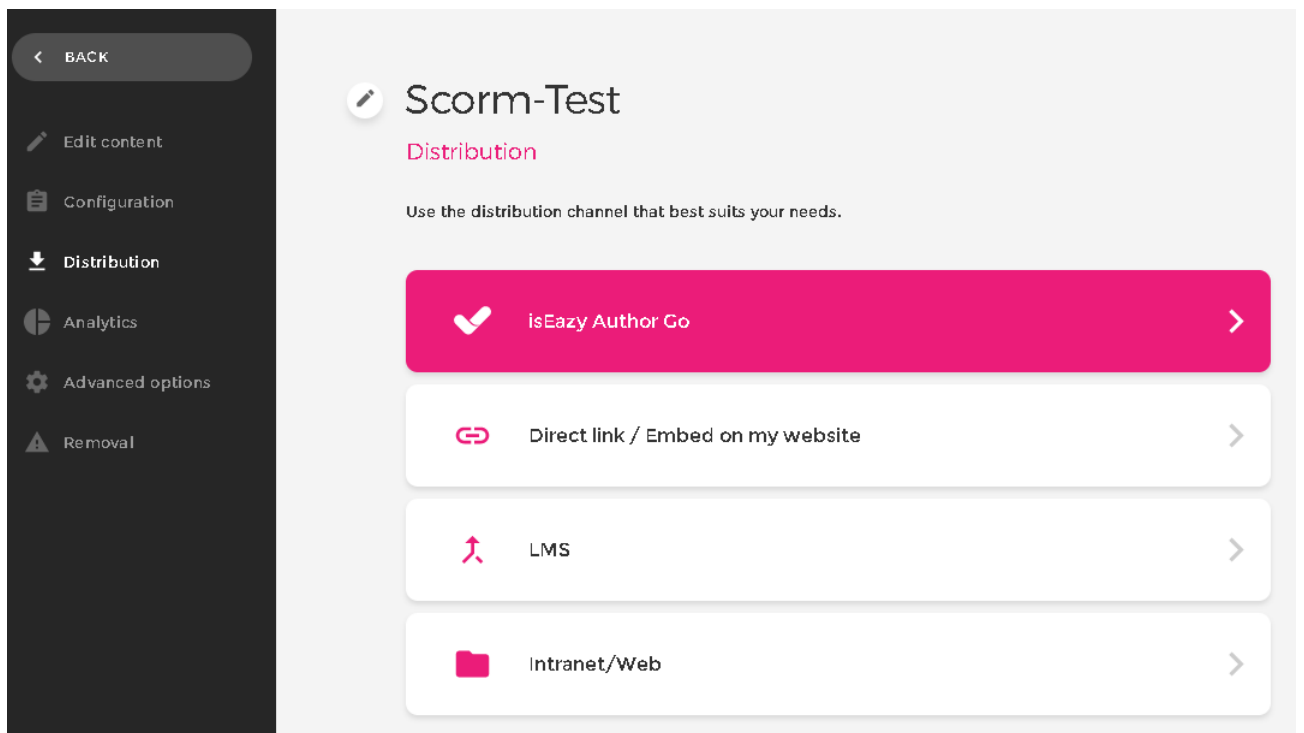


Figura 52 – Menu de Distribuição no ISEAZY

Fonte: Autor

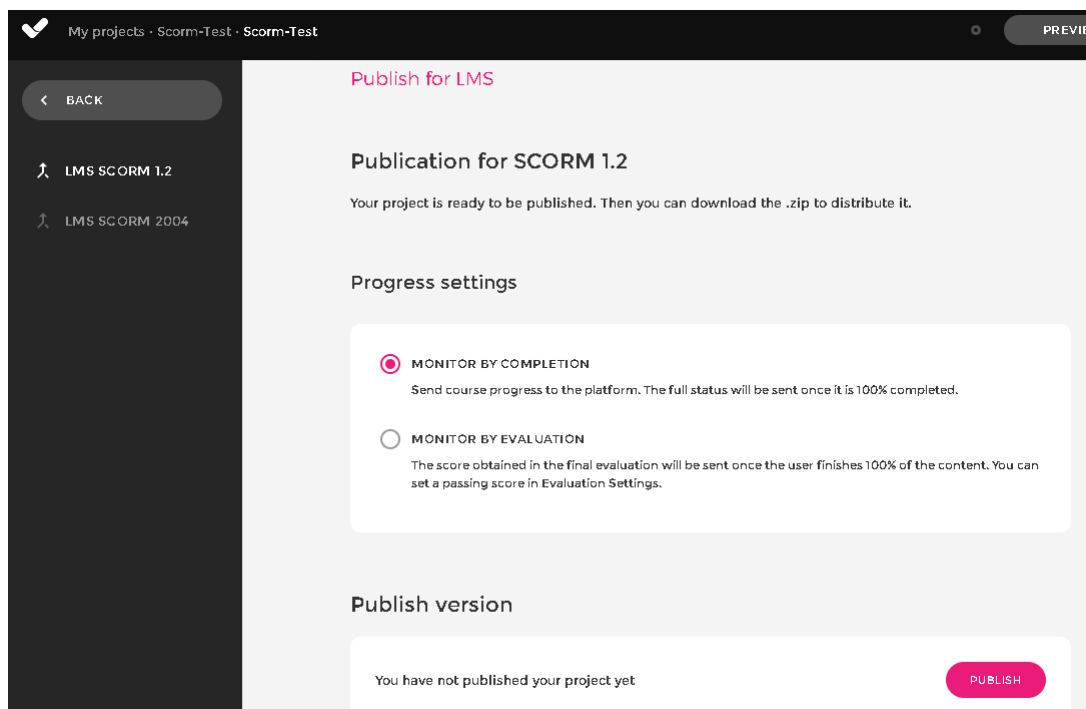


Figura 53 – Menu de Publicação no ISEAZY

Fonte: Autor

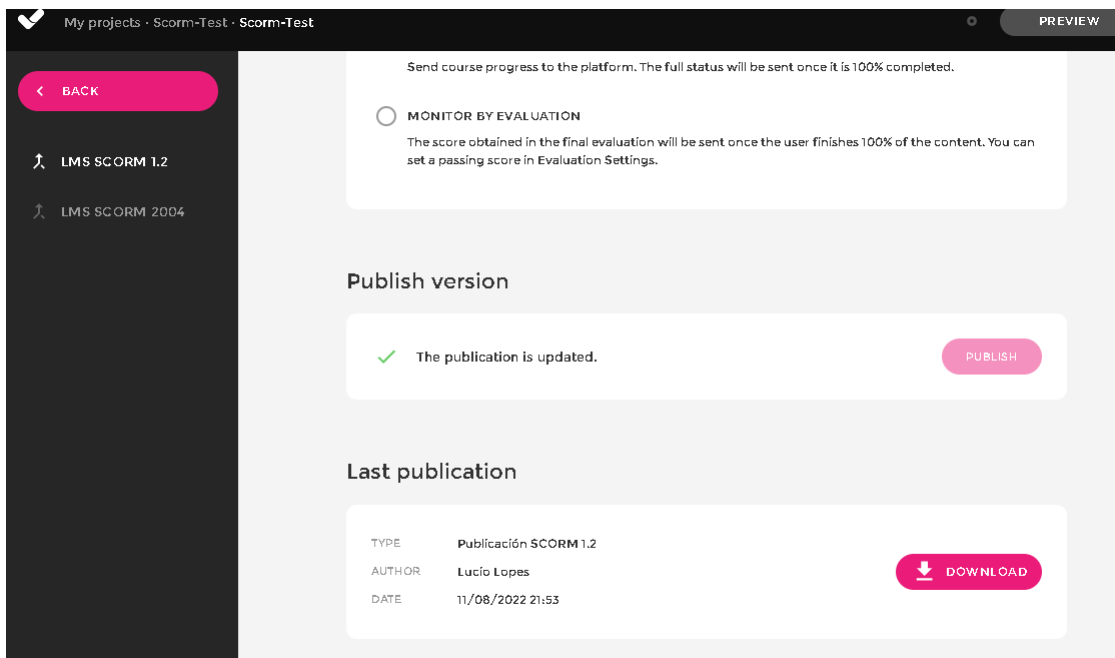


Figura 54 – Tela de Download no ISEAZY

Fonte: Autor