

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

HELOISA VARGAS MEGDA DE OLIVEIRA
MARCELA SANAE HONMA

Criação de material didático de robótica educacional para professores utilizando
Tinkercad Circuitos e Arduino como ferramentas digitais

São Carlos
2022

HELOISA VARGAS MEGDA DE OLIVEIRA
MARCELA SANAE HONMA

Criação de material didático de robótica educacional para professores utilizando
Tinkercad Circuitos e Arduino como ferramentas digitais

Monografia apresentada ao Curso de
Engenharia Mecatrônica, da Escola de
Engenharia de São Carlos da Universidade de
São Paulo, como parte dos requisitos para
obtenção do título de Engenheira Mecatrônica.

Orientador: Prof. Dr. Daniel Varela
Magalhães

São Carlos
2022

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

O48c Oliveira, Heloisa Vargas Megda de
Criação de material didático de robótica educacional para professores utilizando Tinkercad Circuitos e Arduino como ferramentas digitais / Heloisa Vargas Megda de Oliveira, Marcela Sanae Honma; orientador Daniel Varela Magalhães -- São Carlos, 2022.

Monografia (Graduação em Engenharia Mecatrônica) --- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2022.

1. Robótica educacional. 2. Protagonismo do aluno. 3. Programação em blocos. 4. Eletrônica básica. I. Oliveira, Heloisa Vargas Megda de. II. Honma, Marcela Sanae. III. Título.

FOLHA DE AVALIAÇÃO

Candidatas:

HELOISA VARGAS MEGDA DE OLIVEIRA

MARCELA SANAE HONMA

Título: Criação de material didático de robótica educacional para professores utilizando Tinkercad Circuitos e Arduino como ferramentas digitais

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia de São Carlos da
Universidade de São Paulo
Curso de Engenharia Mecatrônica.

BANCA EXAMINADORA

Professor Daniel Varela Magalhães
(Orientador)

Nota atribuída: 10,0 (Dez)

Daniel Varela Magalhães

(assinatura)

Professora Maíra Martins da Silva

Nota atribuída: 10,0 (Dez)

Daniel Varela Magalhães

(assinatura)

Eng. Luiz Alves Antônio Neto

Nota atribuída: 10,0 (Dez)

Daniel Varela Magalhães

(assinatura)

Média: 10,0 (Dez)

Resultado:

Data: 21/07/2022.

Este trabalho tem condições de ser hospedado no Portal Digital da Biblioteca da EESC

SIM NÃO

Visto do orientador,

Daniel Varela Magalhães

AGRADECIMENTOS

Aos nossos pais e irmãs, que nos deram essa oportunidade e nos apoiaram para realização deste trabalho;

Ao nosso orientador Daniel e a todos os professores, que nos ensinaram e continuam ensinando a tantos alunos com muita dedicação e esforço;

Aos amigos, pelo companheirismo e pela troca de experiências, que nos permitiram crescer não só como indivíduos, mas também como formandas;

À Universidade de São Paulo e à Escola de Engenharia de São Carlos, pelas inúmeras oportunidades que promove no campus;

Ao Projeto Semente, que instigou a vontade de atuar na área de educação associada à tecnologia e proporcionou experiências valiosas dentro e fora de sala de aula como voluntárias do projeto;

Ao Projeto Pequeno Cidadão, que confiou na proposta desse trabalho e contribuiu diretamente em sua construção.

RESUMO

Oliveira, H. V. M.; Honma, M. S. **Criação de material didático de robótica educacional para professores utilizando Tinkercad Circuitos e Arduino como ferramentas digitais.** Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2022.

Esse trabalho tem como principal objetivo aproximar professores e, conseqüentemente, alunos da robótica educacional no ambiente escolar. Para isso, foi construído um material didático de introdução à programação em blocos e eletrônica básica voltado para professores que têm o interesse e a possibilidade de promover esse tipo de aula para alunos com idades de 13 e 14 anos, mas que não necessariamente possuem conhecimento específico nessas áreas. As ferramentas digitais escolhidas para desenvolver o trabalho foram o Tinkercad Circuitos e o Arduino em virtude de acessibilidade, abrangência de aplicações, praticidade e integração entre elas. O material consiste em 14 planos de aulas e 4 guias relacionados aos primeiros passos em ambas ferramentas. Para as aulas, buscou-se trabalhar o protagonismo do aluno, estimulando a criatividade, a discussão, a colaboração e o desenvolvimento de projetos. Os resultados foram obtidos pela elaboração do próprio material e pela avaliação de uma psicóloga e uma coordenadora pedagógica do Projeto Pequeno Cidadão, uma iniciativa que atua com formação educacional de jovens. Em suas perspectivas, o material proposto atende aos quesitos de escolha das ferramentas, abordagem de conteúdo e visão do papel do professor e do aluno, mas reforçam a importância do suporte aos professores que não são familiarizados com robótica educacional.

Palavras-chave: Robótica educacional. Protagonismo do aluno. Programação em blocos. Eletrônica básica.

ABSTRACT

Oliveira, H. V. M.; Honma, M. S. **Creation of didactic material in educational robotics for teachers using Tinkercad Circuits and Arduino as digital tools.** Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2022.

The main goal of this paper is to bring teachers and, consequently, students closer to educational robotics in a school environment. Therefore, a didactic material about introduction of blocks-based programming and basic electronics for teachers who have interest and possibility to promote this type of class for students aged 13 to 14 years, but not necessarily have specific knowledge in these areas. The digital tools that were chosen to develop the project were Tinkercad Circuits and Arduino due to accessibility, wide variety of applications, convenience, and integration between them. The material consists of 14 lessons plans and 4 guides related to the first steps in both tools. For the classes, it encourages the student protagonism, stimulating creativity, discussion, collaboration, and project development. The results were obtained by the creation of their own material and the review of a psychologist and an educational coordinator from the *Projeto Pequeno Cidadão*, an initiative that collaborates with citizen formation of young people. From their perspectives, the proposed material attends to the points of selection of platforms, content structure and roles of teacher and student, but they reinforce the importance of supporting teachers who are not familiar with educational robotics.

Key words: Educational robotics. Student protagonism, Blocks-based programming. Basic electronics.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 - Diagrama dos 4Ps da aprendizagem criativa | 23 |
| Figura 2 -Placa Arduino Uno Rev3 | 32 |
| Figura 3 - IDE do Arduino | 32 |
| Figura 4 - Placa micro:bit v2 | 33 |
| Figura 5 - Ambiente de desenvolvimento Makecode | 34 |
| Figura 6 - Placa Raspberry Pi 3 Model B | 35 |
| Figura 7 - Ambiente operacional Raspbian | 36 |
| Figura 8 - Área de montagem dos circuitos | 37 |
| Figura 9 - Área de montagem e de programação em bloco | 37 |
| Figura 10 - Área de gerenciamento de projetos e aulas | 38 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 - Listagem dos guias do material didático e seus respectivos objetivos | 26 |
| Tabela 2 - Listagem dos guias do material didático e seus respectivos objetivos | 28 |
| Tabela 3 - Preço pesquisados das placas | 36 |

SUMÁRIO

| | |
|--|----|
| 1. INTRODUÇÃO | 18 |
| 2. OBJETIVO | 20 |
| 3. BASE TEÓRICA | 21 |
| 3.1 Construcionismo | 21 |
| 3.2 Construtivismo como influência | 21 |
| 3.3 O Construcionismo e seus conceitos | 22 |
| 3.4 Aprendizagem criativa | 23 |
| 4. METODOLOGIA | 25 |
| 5. DESENVOLVIMENTO | 31 |
| 5.1 Comparação entre placas disponíveis | 31 |
| 5.1.1 Arduino | 31 |
| 5.1.2 micro:bit | 33 |
| 5.1.3 Raspberry Pi | 34 |
| 5.2 Comparação entre plataformas disponíveis | 36 |
| 5.2.1 Tinkercad | 36 |
| 5.2.2 s4a | 38 |
| 5.3 Escolha da placa e da plataforma | 39 |
| 6. RESULTADOS | 41 |
| 6.1 Avaliação | 41 |
| 7. CONCLUSÃO | 43 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 44 |
| APÊNDICE A – Material didático do curso “Robótica Legal” | 47 |

1. INTRODUÇÃO

Entender que é natural a transformação substancial do mundo de tempos em tempos em como ele vigora é concordar com a modernização das estruturas de cultura, mobilidade, interações sociais, comunicação, entre outros. No entanto, diferente dos séculos passados, em que a tecnologia da época demorava cerca de 100 anos para ser consolidada no sistema, hoje temos uma sociedade imersa na interatividade, conectividade, expansão da internet, em que as informações chegam numa velocidade nunca antes vista. Estamos na era digital, em que o acesso mais fácil à tecnologia reflete na extrema dependência de ter conhecimento no uso de ferramentas tecnológicas para se inserir e se desenvolver na sociedade.

Apesar das gerações mais novas, nascidas a partir dos anos 2000, já estarem mais familiarizadas com o uso de celular ou tablet, a estrutura de aprendizado na maioria das escolas brasileiras ainda segue o método de ensino tradicional, no qual o professor é o protagonista e detentor do conhecimento e esse é transmitido por meio de aulas expositivas, livros e cadernos, sem a utilização de novas ferramentas tecnológicas. Embora se tenha incentivo sobre o uso da tecnologia em prol do aprendizado nas escolas, são ações pontuais extracurriculares, em que os programas são fora do horário das aulas e não abrangem todos os alunos ou são aqueles que se adaptam aos materiais disponíveis no ambiente (19).

A teoria construcionista diz que o aprendizado é mais eficaz quando se tem a oportunidade da experimentação, por meio de projetos, resolução de problemas ou construção de objetos (1), potencializada com o uso de computadores. É nesse cenário que entra a Robótica Educacional, onde é possível trazer a inclusão das crianças no mundo digital, não somente como aparato tecnológico utilizado no dia-a-dia, mas também como ferramenta de aprendizado que potencializa e prepara os alunos para essa mesma sociedade em constante transformação.

A inserção da tecnologia na aprendizagem dos jovens ainda é um processo lento no Brasil em virtude do modelo tradicional aplicado. Tal cenário é impactado pela formação de professores, em que a adaptação à era digital que estamos inseridos é insuficiente e o padrão professor protagonista e alunos passivos ainda é seguido. Nos currículos dos cursos de Pedagogia é raro, senão inexistente, uma matéria que integre as necessidades didáticas com as ferramentas tecnológicas que os futuros professores poderão vir a utilizar (19). Além disso, tem-se a questão financeira, visto que é necessário investimento para compra de material para ensino de robótica e também capacitação de seus educadores para inserirem

as tecnologias em suas aulas a fim de atrair seus alunos e prepará-los de forma mais condizente com o contexto atual (20).

Diante desses fatos, o trabalho em questão visa facilitar o contato da robótica educacional com os professores, desmistificando a ideia de ser um conteúdo de difícil aprendizado por meio de um material didático e simplificado, que instigue o educador a aprender mais sobre programação e eletrônica e inserir essas e outras ferramentas em suas aulas, tornando-as mais divertidas e dinâmicas.

Além disso, o trabalho busca promover um método que agrega metodologias ativas e ferramentas tecnológicas de baixo custo que auxiliem o aprendizado do aluno, estimulando a curiosidade e criatividade, e que também o atraiam para esse assunto cada dia mais em pauta atualmente.

Existem muitas áreas no Brasil que ainda sofrem com a falta de inovação tecnológica, e a educação é uma delas. Sem uma base que possa desfrutar dos benefícios dessas ferramentas, é difícil desenvolver uma sociedade que acompanhe o crescimento tecnológico mundial. Assim, oferecer alternativas pensadas na realidade brasileira e que se adequem com o sistema educacional atual é ajudar o progresso da sociedade.

As experiências acadêmicas extracurriculares das alunas na robótica educacional ajudaram na construção da metodologia do trabalho, que seguiu por três etapas. Iniciou-se pela pesquisa de conteúdo, fundamentando-se em abordagens educacionais, seguida pela estruturação do material didático, com discussões do formato de aula, conteúdo, escolha da plataforma e da placa. A terceira etapa se deu pela elaboração e revisão dos guias e dos planos de aula, sendo um dos objetivos auxiliar o professor no primeiro acesso às ferramentas e introdução a área.

Dessa forma, o trabalho está organizado em tópicos. Inicia-se pela fundamentação teórica, que apresenta o incentivo do uso das metodologias de ensino; segue pela narrativa da metodologia usada na construção do projeto, prossegue a análise e seleção de ferramentas digitais disponíveis no mercado e finaliza com uma avaliação do material por profissionais da área de educação. O material didático completo está anexado no final do trabalho.

2. OBJETIVO

O objetivo deste trabalho é criar um material didático que contribua para a inclusão de conteúdos relacionados à ciência e tecnologia nas escolas brasileiras. Nesse sentido, o material proposto tem o propósito de dar apoio a professores que possuem interesse em trabalhar com introdução à programação e eletrônica em sala de aula por meio do uso de ferramentas digitais. Além do conteúdo em si, objetiva-se também trazer nas aulas o protagonismo do aluno, em que diversas abordagens são estimuladas e desenvolvidas durante o curso.

3. BASE TEÓRICA

3.1 Construcionismo

Concebida por Seymour Papert na década de 1960, o construcionismo é uma abordagem educacional que defende que o processo de aprendizagem é muito mais eficiente quando as crianças fazem parte da construção do conhecimento, tendo contato com a situação e desenvolvendo soluções independentes. Somado a isso, Papert estudou e elaborou o pensamento computacional, considerando os computadores também como um instrumento de aprendizagem, algo bem visionário na sua época.

O matemático nasceu na África do Sul e mais tarde mudou-se para os EUA, onde lecionou no Massachusetts Institute of Technology (MIT) e criou a linguagem de programação LOGO voltada para educação. É nesse contexto influenciado pelas ideias construtivistas de Piaget que Papert se torna o precursor da Robótica Educacional.

3.2 Construtivismo como influência

O Construtivismo foi idealizado por Jean Piaget, um biólogo suíço reconhecido pelo seu trabalho com desenvolvimento infantil, que posteriormente serviu de base para diversos estudos na área de psicologia e pedagogia (3). A teoria discute que o conhecimento de cada aluno é resultado de sua própria construção pessoal, onde o aprendizado é concebido de forma gradual e em que a cada novo fundamento aprendido é oriundo de conceitos anteriores (4).

Além disso, o construtivismo tem como ideal de que a interação com o ambiente social e físico é importante para o processo de aprendizagem, pois é por meio de estímulos que as crianças criam esquemas mentais e formulam hipóteses para tentar resolver a situação (5). Para isso, a educação deve criar métodos que estimulem essa construção (6).

No espaço de ensino, a teoria construtivista também traz a forma de interação entre o professor e aluno. Sendo o estudante o protagonista no processo de aprendizagem, onde seu nível de amadurecimento e assimilação é respeitado, o professor é visto como um mediador e facilitador na troca de experiências e informações (4), que irá orientar e estimular o aluno para que este realize as suas próprias descobertas e construa seu conhecimento (5).

3.3 O Construcionismo e seus conceitos

Como uma extensão do Construtivismo, o Construcionismo concorda que o processo de aprendizado é por meio do próprio contato do aluno com o mundo, e que o mesmo é o construtor de suas próprias estruturas cognitivas (5). No entanto, a busca de Papert se dá nos meios de aprendizagem para a construção do conhecimento, onde a criação de um produto aproxima e concretiza o aprendizado, com a visualização, externalização, discussão, examinação, admiração e análise do artefato. Papert acredita que esta é a principal característica do Construcionismo, por permitir examinar mais de perto a ideia da construção mental (7).

A ferramenta que Papert encontrou para fundamentar sua teoria foi o computador, que juntamente com a linguagem de programação LOGO que criou, denomina a sua teoria pela qual o aprendiz constrói, por intermédio de um computador, o seu próprio conhecimento. (Papert,1986).

A linguagem de programação LOGO explora conteúdos espaciais e geométricos, na qual a criança conduz uma tartaruga por meio de comandos simples: lineares (para frente 10, para trás 10), de rotação angular (esquerda 90, direita 90), entre outros. É nessa construção de caminhos que possibilita a externalização dos conceitos abordados, pois a criança tem a oportunidade de explorar diferentes alternativas, testar suas hipóteses, errar, analisar seus erros e entender como o computador compreende as instruções dadas.

Dentro do ambiente computacional, o professor é um direcionador, onde ele propõe situações ao aluno, acompanha o progresso de exploração e encoraja a busca de caminhos alternativos. Não mais como antigamente, em que era um detentor do conhecimento, o professor agora é um mediador que, com auxílio do computador, instiga a curiosidade dos alunos e os orienta para que busquem suas próprias soluções.

Papert também define dimensões que servem para estruturar um ambiente de aprendizagem num espaço computacional (8). Ao final dos seus estudos, ele define cinco:

- Dimensão pragmática: o conteúdo a ser elaborado deve ser para um fim prático, de forma que o aluno sinta que está desenvolvendo algo útil;
- Dimensão sintônica: Ao permitir que o aluno escolha o tema que será trabalhado, faz com que o projeto se torne mais relevante, transmitindo uma relação de sintonia do aluno com o conteúdo abordado;

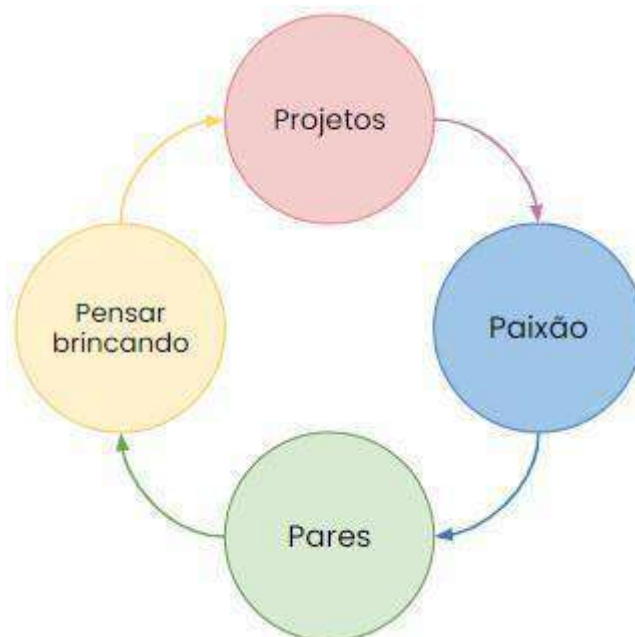
- Dimensão sintática: Por meio do seu desenvolvimento cognitivo, o aluno pode seguir com o aprendizado sem a necessidade de passar por conteúdos anteriores, mostrando a facilidade da criança em compreender o ambiente educacional;
- Dimensão semântica: O aluno interage com os componentes que tenham significado para ele, e não necessariamente seja cunho formal;
- Dimensão social: O dia-a-dia do aluno é inserido no conteúdo das atividades, de maneira que ele interaja com o ambiente que está imerso.

Utilizando essas dimensões para desenvolver atividades, há boas chances de que a construção do aprendizado do aluno seja feita de forma correta (8).

3.4 Aprendizagem criativa

Conceito que tem como base o construtivismo e o construcionismo, a aprendizagem criativa foi fundamentada por um grupo de pesquisadores do MIT Media Lab, liderado por Mitchel Resnick, que teve contato com Seymour Papert. Ela foca na construção dos ambientes de aprendizagem baseado em quatro pilares, chamado 4 Ps da aprendizagem criativa: paixão (*passion*), pares (*peers*), pensar brincando (*play*) e projeto (*project*).

Figura 1 - Diagrama dos 4Ps da aprendizagem criativa



Fonte: Autoral

A abordagem é integrada por um pensamento espiral: dadas opções de criações, a criança imagina o que quer fazer, identifica-se com o significado do projeto (paixão), inicia a construção, brincando e explorando suas alternativas (pensar brincando), compartilha suas ideias e caminhos com seus pares, de forma colaborativa, assim construindo seu projeto. Dessa forma, ela reflete a respeito do processo e seus intuítos, o que por sua vez reinicia a busca de novas ideias, passando por mais experiências e desenvolvendo seu aprendizado (18).

Paixão é um pilar da aprendizagem criativa que remete ao entusiasmo e que instiga a curiosidade do aluno. Como eles são incentivados a buscar temas que lhe interessam, relacionado ou não às experiências pessoais ou do grupo que estão inseridos, é possível ver uma turma mais envolvida, pois o assunto os aproxima do desenvolvimento do projeto.

As interações sociais são mais trabalhadas no pilar intitulado de Pares. É nesse cenário que acontecem a troca de ideias, busca de soluções por meio de colaboração dos colegas e inspiração nas criações de outros grupos. Com ambientes mais colaborativos, é possível trabalhar em ideias que não seriam viáveis se fossem executadas sozinhas.

É no Pensar brincando que a criança inicia a exploração e experimentação livres, por meio de materiais ou conteúdos não estruturados, instigando a imaginação. É no ambiente lúdico que a visão é expandida, ideias novas são experimentadas, limites são testados e o erro é visto como a busca de soluções inovadoras.

Por fim, o Projeto é um pilar que estrutura o processo de construção de algo em torno de uma ideia (18). Considerando que as pessoas aprendem mais quando estão envolvidas no desenvolvimento de um produto e em um ambiente colaborativo, na aprendizagem criativa não é o projeto finalizado que quantifica o aprendizado, mas sim todo o processo de construção deste, de forma que ele atue como um direcionador na busca de conhecimento.

4. METODOLOGIA

No planejamento inicial do trabalho, o objetivo era criar o material, aplicar o curso para uma turma de alunos durante um semestre e, por fim, ajustá-lo e aprimorá-lo conforme a experiência das aulas. A aplicação seria feita com crianças e adolescentes que fazem parte do Projeto Pequeno Cidadão, uma iniciativa pública e privada que busca orientar jovens em vulnerabilidade social através de uma formação socioeducativa com duração de 4 anos. O projeto busca a equidade educacional, oferecendo garantia alimentar e ambiente seguro e desenvolvendo competências por meio de esportes, leitura e tecnologia. Especialmente em ciências exatas, existem parcerias com iniciativas de professores e estudantes da USP São Carlos, como o Projeto Semente EESC-USP, que oferecia módulos relacionados à introdução ao pensamento computacional, desenvolvimento de produtos e projetos. Entretanto, em decorrência do isolamento social durante a pandemia da COVID-19 e das medidas de segurança epidemiológicas adotadas dentro do campus, não foi possível realizar um curso piloto durante esse período. Assim, a solução alternativa encontrada foi criar o material, pedir uma avaliação de especialistas em educação do Projeto Pequeno Cidadão e revisar o material. Desse modo, mesmo sem o desenvolvimento na prática, seria uma forma de obter a visão de profissionais cuja área de atuação não é necessariamente de tecnologia e ciências exatas, mas com embasamento e experiência pedagógica.

O primeiro passo foi construir o material, que foi desenvolvido a partir de quatro etapas: pesquisa, estruturação, elaboração e revisão. Como já tínhamos alinhado de criar um material relacionado à introdução de programação e eletrônica para alunos do 9º do Ensino Fundamental II, foram pesquisados exemplos de materiais didáticos e artigos na internet associados aos temas. Existem diversos artigos disponíveis em sites de busca sobre o uso didático de plataformas de programação e eletrônica voltado para diferentes públicos e com motivações e direcionamentos específicos de cada estudo. O material que foi utilizado como referência na abordagem e na estrutura das aulas foi *Creative Computing Curriculum* (23), um guia criado por membros do time de pesquisa da *Harvard Graduate School of Education* que reúne ideias, estratégias e atividades para introduzir computação criativa utilizando o Scratch. Baseado na aprendizagem criativa e no construcionismo, o guia destaca-se pelo estímulo da criatividade, abordagem didática, clareza dos objetivos dos conteúdos, cobertura do nível introdutório, intermediário até avançado e método avaliativo da aprendizagem dos alunos. Além disso, a experiência de participar e coordenar turmas do Projeto Pequeno

Cidadão e do Projeto Semente contribuiu para um maior entendimento na dinâmica de aulas práticas, além da interação com os alunos da turma.

A segunda etapa foi estruturar o formato, os temas e a abordagem das aulas. Para que o professor pudesse acessar as ferramentas e explorar seus recursos, foram planejados montar guias à parte das aulas de como se cadastrar no Tinkercad e como fazer a instalação da IDE do Arduino. Em relação ao formato das aulas, definiu-se que a duração do curso seria relativa a um semestre escolar com 14 aulas de uma hora cada. Vale ressaltar que se trata de uma sugestão de formato, sendo importante o professor adaptá-la de acordo com o contexto da turma e da instituição escolar. Já no quesito de temas das aulas, discutimos quais conteúdos seriam relevantes para introduzir a programação e a eletrônica, considerando que os alunos têm conhecimento em matemática e física referente ao seu ano escolar pela Base Nacional Comum Curricular (BNCC) e que há alunos que já tiveram contato com programação e eletrônica ou não. Acerca da abordagem das aulas, definimos nos basear nas teorias construtivistas e construcionistas e da aprendizagem criativa, como foram descritas na seção teórica. O objetivo seria a aprendizagem centrada no aluno, o qual é estimulado a ter consciência no seu processo de aprendizagem, além de criar, testar e ajustar projetos práticos de forma colaborativa. Para alcançar esse nível de engajamento do aluno, seria importante então contextualizar as atividades com situações significativas do cotidiano, incentivar as interações e colaborações e trazer dinâmicas lúdicas que estimulam a criatividade dos alunos.

A terceira etapa foi elaborar os guias e os planos de aula. Como mencionado anteriormente, o intuito do guia é orientar o professor no primeiro acesso às ferramentas e deixá-lo familiarizado com a interface e os recursos de cada uma. Nesse sentido, foram mapeados três guias de passo-a-passo e mais um com informações importantes e boas práticas de circuitos eletrônicos. Optou-se por organizar esse último para servir como material geral de consulta nas aulas. A tabela abaixo apresenta identificação, tema e objetivo de cada guia.

Tabela 1 - Listagem dos guias do material didático e seus respectivos objetivos

| Identificação | Tema | Objetivo |
|----------------------|---------------------|-----------------------------------|
| Guia 1 | Como se registrar e | Mostrar o passo-a-passo de como o |

| | | |
|--------|--|---|
| | criar um novo projeto no Tinkercad | educador e o aluno podem se registrar, explicar as funcionalidades no ambiente de criação de projeto. |
| Guia 2 | Como criar uma sala de aula no Tinkercad | Mostrar o recurso da sala de aula na plataforma e explicar o funcionamento da sala na visão do educador e do aluno. |
| Guia 3 | Como baixar e usar a IDE do Arduino | Mostrar o passo-a-passo de como instalar a IDE no computador e como carregar um programa do Tinkercad na IDE para viabilizar o teste com protótipo. |
| Guia 4 | Informações importantes e boas práticas no Circuitos Tinkercad | Compilar informações, boas práticas e termos usuais considerados importantes no desenvolvimento de um projeto eletrônico. |

Fonte: Autoral

Em relação aos planos de aulas, a estrutura seguiu um padrão básico de material didático, sendo organizada em 4 tópicos gerais: objetivos, recursos materiais, atividades propostas e avaliação da aula. Para a criação do conteúdo, foram seguidas algumas diretrizes gerais para aplicar a visão de educação centrada no aluno proposto pelas teorias que embasam esse trabalho. A sequência de aulas foi estruturada de tal forma que houvesse um progresso não só na abordagem de conteúdos, mas também na autonomia dos alunos em relação ao desenvolvimento de um projeto. É essencial que os alunos não dependam das explicações passo-a-passo do educador para realizar as práticas propostas em aulas; o objetivo é que eles gradativamente discutam e reflitam sobre os caminhos de como explorar um problema e construir uma solução; tudo isso com a orientação e acompanhamento de um docente. Além disso, busca-se também contribuir para que eles adquiram consciência do seu próprio papel na sua formação educacional e na participação ativa na melhoria das aulas por meio das dinâmicas das aulas e as pesquisas de percepção do aluno.

A tabela abaixo apresenta identificação, tema e objetivo de cada aula.

Tabela 2 - Listagem dos guias do material didático e seus respectivos objetivos

| Identificação | Tema | Objetivo(s) |
|----------------------|--|---|
| Aula 1 | Primeiro contato | <p>Criar um ambiente confortável para interação</p> <p>Obter um visão geral da turma sobre contato com programação, elétrica e eletrônica</p> <p>Trazer conceitos básicos de programação de forma lúdica e educativa</p> |
| Aula 2 | Conhecendo a placa | <p>Introduzir placas, sobretudo o Arduino, mostrando aplicações, explicações gerais e suas partes</p> <p>Servir como momento exploratório para despertar a curiosidade no tema e tirar possível receio da complexidade do curso</p> |
| Aula 3 | Conhecendo a plataforma (parte 1) | <p>Introduzir a plataforma do Tinkercad Circuitos, discutindo sobre exemplos de projetos e explorando a interface</p> |
| Aula 4 | Conhecendo a plataforma (parte 2) | <p>Retomar as funcionalidades do Tinkercad Circuitos introduzidas na aula anterior</p> <p>Apresentar e discutir sobre os blocos de programação</p> |
| Aula 5 | Introdução à eletrônica básica e pisca LED | <p>Abordar conceitos de elétrica e eletrônica</p> <p>Montar um projeto de circuito simples</p> |
| Aula 6 | Introdução à entrada e saída digital | <p>Introduzir os conceitos de entrada e saída digital</p> <p>Construir um projeto de circuito aplicando esses conceitos, retomando a prática feita na</p> |

| | | |
|---------|---|---|
| | | aula 5 |
| Aula 7 | Introdução a entrada e saída analógicas e comparação de sistema digital vs. analógico | Introduzir conceitos de sistemas analógicos e discutir diferenças entre sistemas digitais e analógicos em contextos do cotidiano Montar um projeto de circuito aplicando esses conceitos, em que o aluno tenha maior percepção das etapas gerais na construção de um projeto |
| Aula 8 | Conhecendo o sensor ultrassônico | Introduzir as principais características do sensor ultrassônico hc-sr04 Montar um projeto de circuito utilizando o componente abordado com maior independência na concepção e construção do projeto |
| Aula 9 | Conhecendo o servo motor | Orientar os alunos a procurarem informações sobre o componente (no caso, servo motor), compilar os aprendizados Montar um projeto de circuito a partir do conhecimento adquirido nas pesquisas de forma mais independente |
| Aula 10 | Revisão | Retomar alguns conceitos importantes abordados nas aulas até agora Reforçar o pensamento crítico, a capacidade de resolver problemas e o trabalho em equipe |
| Aula 11 | Projeto - idealização | Contextualizar e discutir a situação-problema para que entendam a motivação do projeto Estimular os grupos a buscarem por uma |

| | | |
|---------|---|--|
| | | solução à situação-problema fornecida |
| Aula 12 | Projeto - desenvolvimento no ambiente virtual | Desenvolver a ideia do projeto na prática, construindo a montagem virtual do circuito e a programação Retomar os processos que foram abordados nas aulas anteriores de como fazer essas atividades propostas sem a necessidade de instruções passo-a-passo de como fazer o projeto rodar. |
| Aula 13 | Projeto - prototipagem | Construir o protótipo do circuito simulado na aula anterior, além de fixá-lo ao objeto associado. Testar e fazer ajustes até que o projeto funcione adequadamente |
| Aula 14 | Projeto - apresentação final e conclusão do curso | Apresentar o projeto de cada grupo, destacando o desenvolvimento de cada etapa e os aprendizados gerais Abordar sobre a importância do feedback e aplicá-lo Obter um feedback final da percepção do projeto e do curso no geral |

Fonte: Autoral

A quarta e última etapa desse processo de criação foi revisar o material para certificar a coerência das aulas e se, de forma geral, cumpre o objetivo proposto.

5. DESENVOLVIMENTO

5.1 Comparação entre placas disponíveis

5.1.1 Arduino

De acordo com o site oficial da plataforma “O Arduino é uma plataforma eletrônica open source baseada em hardware e software” (9). Criada em 2005 na Itália, no Ivrea Interaction Design Institute, por cinco pesquisadores que tinham por objetivo facilitar a prototipagem, focando em baixo custo e de forma simples, o Arduino possui a parte de hardware, compreendida pela placa eletrônica baseada em um microcontrolador e a parte de software, um Ambiente de Desenvolvimento Integrado - IDE (*Integrated Development Environment*), onde os códigos são desenvolvidos na linguagem de programação de C++.

As placas de Arduino possuem pinos analógicos e digitais que podem ser transformados em entradas e saídas do projeto em mente, por exemplo ler o sinal de entrada de um botão sendo pressionado e transformar em um sinal de saída para um buzzer ou um LED. A sua alimentação pode ser feita pela entrada USB, que também é a conexão entre a placa e o software, ou pela fonte externa, que funciona a cerca de 5V.

Além dos pinos analógicos e digitais, as placas de Arduino também apresentam em comum pinos GND (Ground), que seria o pino de referência, o pino VIN, para alimentar a placa através de bateria externa e os pinos de 3.3V e 5V, que fornecem as respectivas tensões para circuitos externos.

O modelo mais utilizado atualmente, Uno R3, possui um microcontrolador ATmega328, 14 portas digitais que podem ser usadas como PWM, 6 portas analógicas, uma memória flash de 32KB, uma memória SRAM de 2KB e memória EEPROM de 1KB.

Somando-se a essas configurações, o Arduino ainda possui botão RESET para reiniciar a placa e LED para indicar se a mesma está recebendo energia, porém para a criação de projetos, tanto básicos como mais complexos, é interessante utilizar a vasta gama de componentes eletrônicos, como LEDs, sensores, resistores, capacitores, shields, módulos entre outros disponíveis no mercado.

Figura 2 -Placa Arduino Uno Rev3

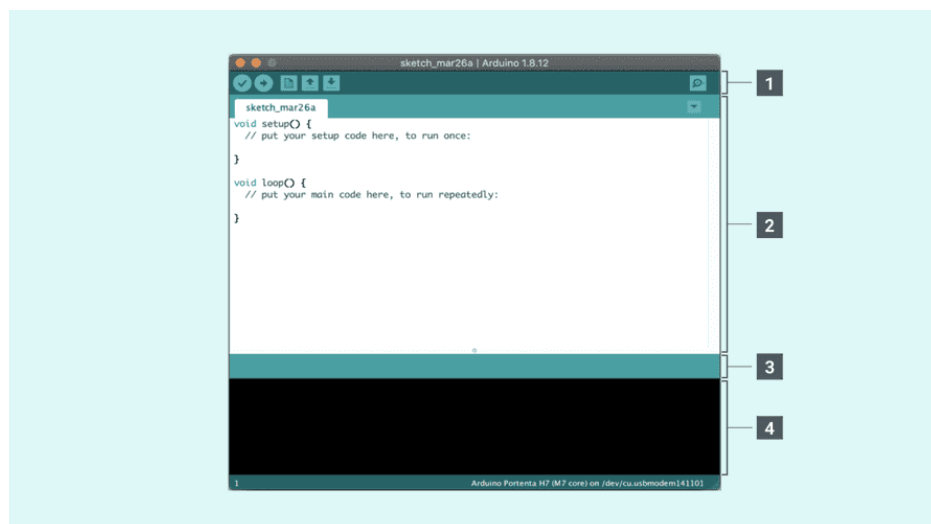


Fonte: Loja Oficial Arduino USA

Todas as placas do Arduino são baseadas em torno de um microcontrolador da família AVR, da Atmel. É por meio dos microcontroladores que ocorrem todos os processos de controle e leitura dos pinos; interface com USB com o computador; de forma que pode-se chamar o microcontrolador de “cérebro” do Arduino.

Para a programação, o Arduino oferece um software próprio, o Arduino IDE (*Integrated Development Environment*) criado em linguagem Java, mas com o objetivo de fazer códigos na linguagem C++. Com uma interface de fácil navegação, é no IDE que além de criar os códigos, é possível compilar, testar e carregar para a placa.

Figura 3 - IDE do Arduino



Fonte: Starting Guide Arduino

As ferramentas que o Arduino oferece fazem com que se abram diversas possibilidades de criação de projetos, desde projetos simples como piscar um LED até automação residencial.

5.1.2 micro:bit

Em 2015 a BBC, uma corporação pública de rádio e televisão do Reino Unido, lançou a micro:bit, uma placa de hardware que tinha como objetivo promover a educação tecnológica, despertando em crianças e jovens o interesse pela ciência, tecnologia e engenharia. Por meio de uma iniciativa em que distribuíram a placa nas escolas públicas do Reino Unido, os alunos poderiam ter o ensino de programação de forma simples e fácil, instigando a curiosidade e criatividade.

Com apenas 52mm x 42mm de tamanho, a placa possui em seu display diversos componentes eletrônicos, como uma matriz de LEDs 5x5, botões, sensores de luz e temperatura, acelerômetro, pinos, conector USB, conector de bateria. Por meio de uma antena interligada na placa, também é capaz de comunicar via Bluetooth e sinais via rádio. Esse layout faz com que a placa seja completa, interativa e de fácil uso.

Figura 4 - Placa micro:bit v2

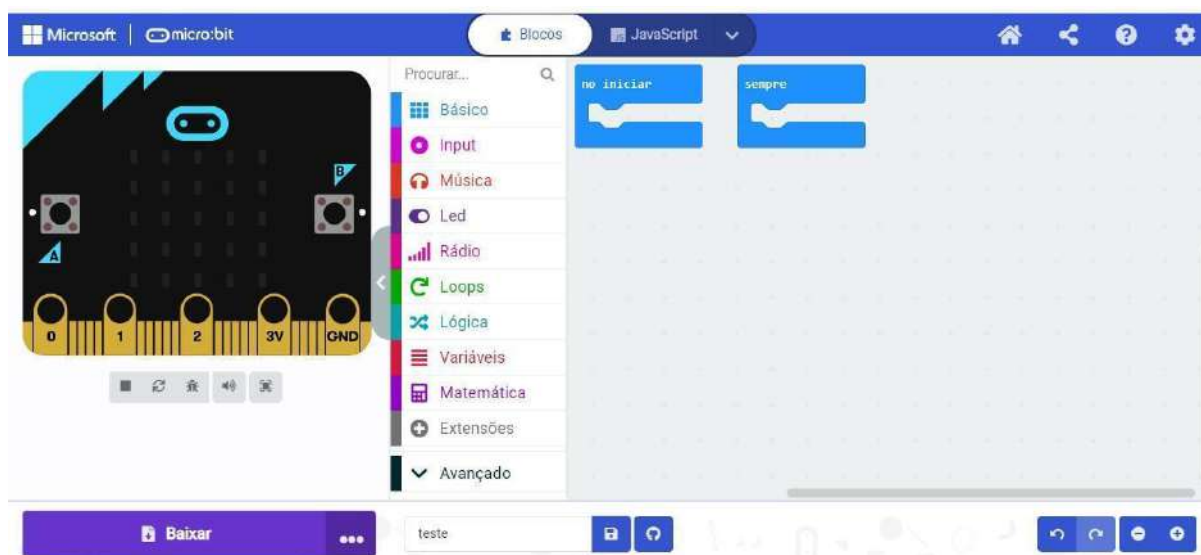


Fonte: Site RoboCore

Para o desenvolvimento dos códigos, a micro:bit oferece ambientes de desenvolvimento para algumas linguagens de programação, como programação em blocos, JavaScript, Python, C/C++ e mobile.

O ambiente de desenvolvimento mais utilizado é o MakeCode para micro:bit, que foi desenvolvido com o apoio da Microsoft. O ambiente segue o estilo Scratch, com a programação em blocos, mas também possui a opção de visualizar o código em JavaScript.

Figura 5 - Ambiente de desenvolvimento Makecode



Fonte: Site MakeCode micro:bit

Contendo diversos componentes em sua placa e somando-se a uma ampla gama de linguagens de programação para criar códigos, a micro:bit é um placa interessante para uso didático, oferecendo um ensino de programação divertido.

5.1.3 Raspberry Pi

Com a Raspberry Pi Model B, primeiro modelo da placa lançado em 2012, a Fundação Raspberry Pi iniciou seu objetivo de tornar a computação acessível a todos por meio de um computador de placa única, de baixo custo e alto desempenho, com software gratuito (14). Quando conectada a periféricos, como mouse, teclado, monitor e cartão microSD, funciona como um computador, com uma interface semelhante a outros sistemas operacionais conhecidos.

Apesar de diversos modelos lançados, todos possuem conexões padrões, variando em tamanho e quantidade, como conector GPIO de 40 pinos, conector display, interface USB, interface HDMI, saída de áudio e vídeo, suporte a câmera, conector de alimentação, além

da entrada para cartão microSD. Atualmente, os modelos mais recentes também possuem conectividade Wifi, Bluetooth e Ethernet.

Figura 6 - Placa Raspberry Pi 3 Model B

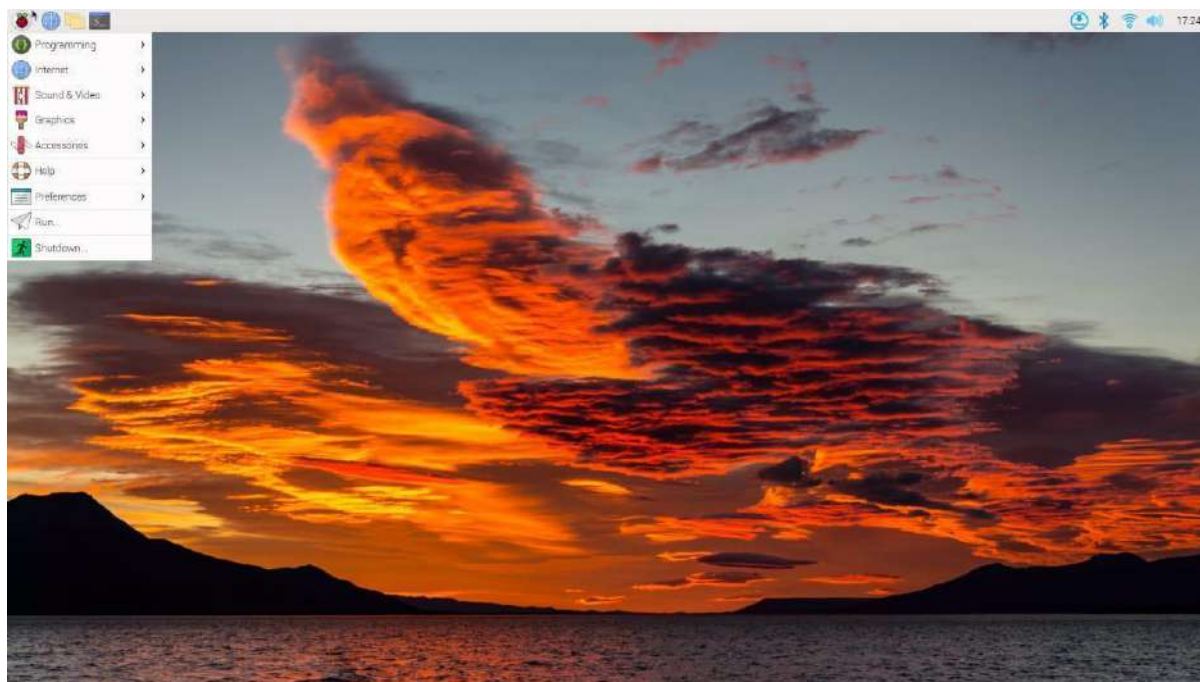


Fonte: Site Filipeflop

O sistema operacional mais utilizado nesse minicomputador é o Raspberry Pi OS (antigamente chamado de Raspbian), e tem como base a distribuição Linux Debian. Apesar da placa receber outros sistemas operacionais, ao se utilizar a Raspberry Pi OS, própria para a placa, ganha-se em otimização e suporte.

Para a programação, o sistema já traz a linguagem Python, porém é possível programar em C/C++, JavaScript, entre outros, sendo necessário fazer instalações no sistema.

Figura 7 - Ambiente operacional Raspbian



Fonte: Autoral

Segue abaixo os valores encontrados em lojas online das placas e kits durante o período de maio de 2022 para fins de comparação.

Tabela 3 - Preço pesquisados das placas

| Placa | Valores |
|-------------------------|-----------|
| Kit micro:bit v2 | R\$415,00 |
| Raspberry Pi 3 Model B+ | R\$450,00 |
| Arduino Uno R3 | R\$130,00 |

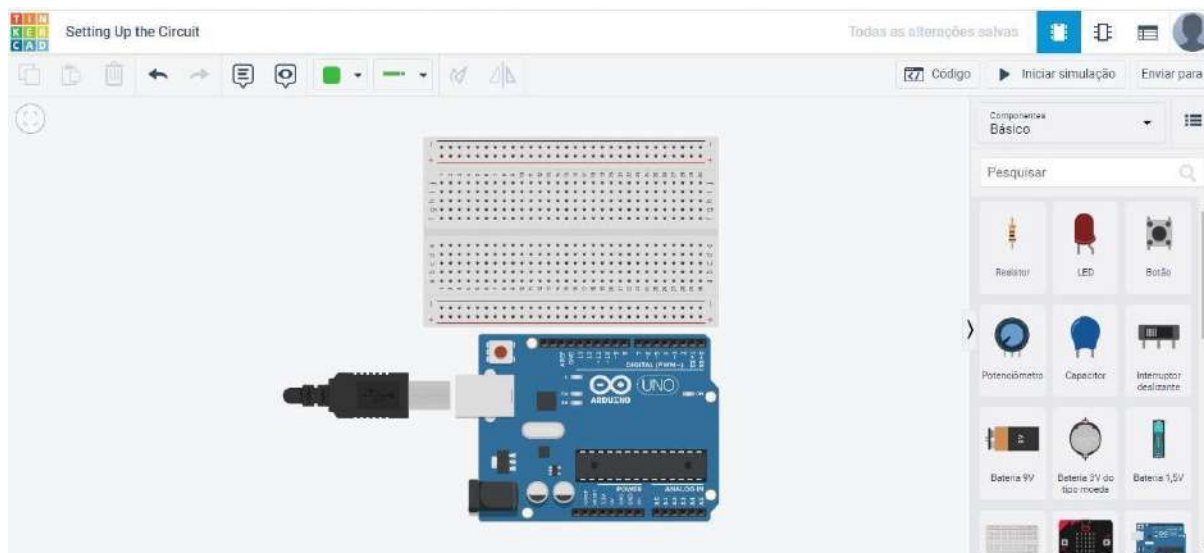
Fonte: Autoral

5.2 Comparação entre plataformas disponíveis

5.2.1 Tinkercad

Idealizado pela Autodesk, o Tinkercad (<https://www.tinkercad.com/>) é uma plataforma online gratuita voltada para ensino de modelagem 3D, eletrônica e programação. Especificamente em relação à parte eletrônica, a plataforma traz um simulador de circuitos em uma interface atrativa, com componentes e bibliotecas de algumas placas de prototipagem, garantindo acesso à educação tecnológica por meio da internet sem a necessidade de ter os recursos físicos à disposição.

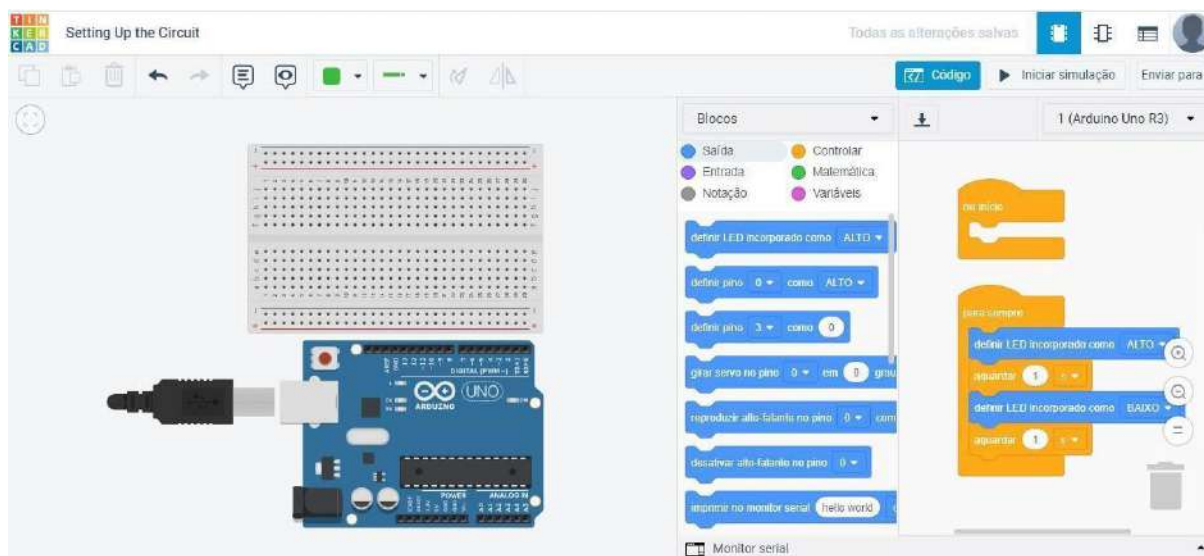
Figura 8 - Área de montagem dos circuitos



Fonte: Site Tinkercad

Na parte de programação, a plataforma tem como base a programação em blocos, mas também oferece visualização e programação em texto na linguagem da respectiva placa em simulação, como C++ para Arduino Uno ou Python para micro:bit.

Figura 9 - Área de montagem e de programação em bloco



Fonte: Site Tinkercad

Além disso, o site tem um sistema de gerenciamento das aulas, com área do educador e a área do aluno, fazendo com que os projetos possam ser organizados,

compartilhados e supervisionados por meio de classes e ampliando os estudos com aulas online ou híbridas. Possui também uma comunidade, onde projetos são compartilhados.

Figura 10 - Área de gerenciamento de projetos e aulas



Fonte: Site Tinkercad

5.2.2 s4a

“S4A é uma modificação do Scratch que permite programação simples da plataforma de hardware aberto Arduino” (17). Com o objetivo de atrair pessoas para programação, a equipe criadora Smalltalk do Citilab utilizou como base o Scratch, uma plataforma online de programação em blocos para idealizar o S4A e inseriu funcionalidades para gerenciar sensores e atuadores e assim viabilizar a interação com as placas Arduino.

Para obter a plataforma, é necessário fazer o download do arquivo na página oficial (http://s4a.cat/index_pt.html) e seguir com as recomendações. Dependendo da placa utilizada na interação com a plataforma, é necessário também baixar mais componentes para um uso adequado.

O S4A tem configurações próprias, como 6 entradas analógicas, 2 entradas digitais, 3 saídas analógicas, 3 saídas digitais e 4 saídas específicas.

É possível abrir projetos do Scratch na plataforma, porém os projetos criados na plataforma não podem ser compartilhados no site do Scratch. O site oficial também disponibiliza conteúdos para ensino, como vídeos, exercícios de exemplos, além de documentação para possíveis dúvidas.

5.3 Escolha da placa e da plataforma

Em relação a placa e as plataformas que as acompanham, comparando os modelos apresentados, foi possível encontrar aquele que se encaixa no objetivo em questão.

A Raspberry Pi é uma placa que possui um hardware potente, o suficiente para rodar um sistema operacional completo, e permite a criação de projetos variados, com diversas conexões de entrada e saída. Porém, além de ter um custo mais elevado, que o torna menos acessível à educação, o processo de configuração e instalação da plataforma Raspberry Pi OS e a própria utilização são mais complexos, o que dificulta o uso para professores e educadores que não são da área e os distancia da ideia de inovação da ferramenta.

Por outro lado, a micro:bit foi criada voltada para o ensino de programação a crianças e jovens. A placa já contém leds e sensores, o que dispensa a necessidade de investir em mais componentes, e com a plataforma MakeCode intuitiva e online, torna o ensino mais simples e versátil. No entanto, o sistema micro:bit é voltado para o aprendizado de computação e programação, e pode dificultar o ensino da parte eletrônica na criação de projetos mais elaborados, pois não há uma estrutura própria na placa para isso. E assim como a Raspberry Pi, seu custo é mais elevado, o que dificulta na questão de acesso e ensino à programação.

Em relação a plataforma S4A, apesar dela ter a interação com Arduino, com a criação do código em blocos e compilação direto na placa, é necessário o download do software no computador e algumas configurações externas caso sejam utilizados componentes eletrônicos que não foram incluídos na plataforma, o que pode dificultar para professores que não estão familiarizados com o mundo de hardware e software. Além disso, existem certas ordenações na interface da S4A em que não há a padronização do idioma português brasileiro, podendo confundir o conteúdo.

Para suprir as necessidades em questão, combinou-se a placa Arduino com a plataforma Tinkercad. Por ser um sistema *open-source* (código aberto), a placa Arduino pode ser reproduzida, diminuindo seus custos e podendo ser adquirida mais facilmente. Suas funcionalidades podem ser expandidas com uso da protoboard e inserção de componentes, gerando mais oportunidades de ensino com projetos mais completos. Junto a isso, a plataforma gratuita e online do Tinkercad possibilita a integração com o Arduino de maneira muito prática, além de trazer um ambiente robusto voltado à aplicação educacional com constantes atualizações de melhoria.

Nesse contexto, a placa Arduino com a plataforma Tinkercad foram juntamente escolhidas, por suas funcionalidades tornarem o ensino de programação mais dinâmico, versátil e acessível.

6. RESULTADOS

Os resultados foram obtidos de duas maneiras: por meio do próprio material criado, com o planejamento de aulas e os guias de acesso ao site Tinkercad e a placa Arduino, e a opinião de duas profissionais da área que atuam em frentes diferentes: uma coordenadora pedagógica e uma psicóloga.

Como robótica educacional é um tema que entrou em pauta recentemente, as entrevistas com as profissionais foram feitas de forma mais abrangente e simplificada, buscando mais a visão e experiência das profissionais do que uma opinião conteudista.

6.1 Avaliação

A coordenadora pedagógica do Projeto Pequeno Cidadão trouxe uma visão estrutural do ambiente educacional. Para ela, o material é inovador, pois traz a junção de dois sistemas gratuitos, a plataforma Tinkercad e a placa open source Arduino, que tem uma flexibilidade às necessidades do ambiente e das aulas, em que podem ser adaptadas num modelo online ou híbrido, dependendo dos recursos disponíveis. Além disso, o material condensa e sistematiza o tema, organizando conteúdos que normalmente estariam dispersos na internet de forma mais didática, o que facilita o acesso aos educadores.

No entanto, a ressalva se dá a capacitação dos educadores. Como os educadores podem não ter recebido uma capacitação para lecionar sobre robótica educacional, era esperado que o material trouxesse esse perfil de conteúdo. Mas, apesar disso, ela enfatizou que o material criado da forma mais introdutória é considerável, pois incentiva os educadores a se interessarem na área e aplicarem nas suas aulas e buscarem capacitação por si próprios. Outra observação se dá que, apesar de se tratar de uma plataforma gratuita e uma placa open source de baixo custo, ainda há o obstáculo financeiro, já que envolve custos por conta da necessidade de computadores, placas e de seus componentes, entre outros.

Pensando mais na questão de desenvolvimento do aluno, a psicóloga do Projeto Pequeno Cidadão enfatizou na avaliação de diagnóstico inicial, intermediária e final que, na sua visão, é uma abordagem importante para acompanhar o aprendizado do aluno e também a efetividade do conteúdo no aprimoramento das habilidades inovadoras que o Projeto trabalha. De fato, a Avaliação de Diagnóstico não é uma forma de mensurar o aluno, mas discernir se a abordagem que está sendo feita está de fato cumprindo com os objetivos estipulados.

A base lúdica nas aulas, vinda do método de aprendizagem criativa, também foi algo positivo que a psicóloga pontuou. Com essa abordagem, busca-se o ensino do raciocínio lógico, criatividade, trabalho em equipe e protagonismo do aluno, quesitos que precisam ser trabalhados com os jovens para o desenvolvimento desses como bons indivíduos. Entretanto, a observação que a psicóloga faz é em relação ao conforto dos educadores que não são da área de tecnologia para aplicar esse conteúdo. Sua sugestão seria fazer um vídeo acompanhando o material, onde seria explicado o conteúdo com exemplificações, de forma a introduzir o mundo de robótica educacional ao educador e o aproximasse do meio, rompendo essa insegurança em relação ao novo conhecimento.

7. CONCLUSÃO

É possível perceber a educação no Brasil ainda não acompanha os avanços tecnológicos, quando se fala sobre o ensino de robótica ou uso de ferramentas tecnológicas para o aprendizado, embora se veja ações pontuais em algumas escolas, o que dá a entender que o assunto possui impactos positivos na aprendizagem e que está ganhando cada vez mais espaço dentro de discussões pedagógicas. O que precisa ser observado é o fato de que a internet e os computadores já estão inseridos de forma crucial na dinâmica mundial e nesse contexto a robótica educacional não tem somente objetivo de formar futuros profissionais na área de tecnologia, mas também de desenvolver cidadãos que se adaptem a essa sociedade em constante transformação. Desse modo, é importante estimular e buscar iniciativas que promovam a implementação nas escolas e atinjam mais crianças e jovens. Juntamente às necessidades das crianças, é necessário um olhar mais apurado em relação aos professores. Para que estes possam aprender as ferramentas tecnológicas e assim educar de forma consciente seus alunos, é importante definir metodologias para tal, disponibilizar materiais, capacitá-los e consolidar métodos dentro das instituições para que os educadores sejam orientados e se sintam seguros para instruir seus alunos. Numa estrutura maior, o ensino superior de educação necessita de uma reformulação com uma visão mais atualizada, integrando a didática pedagógica com as ferramentas digitais.

Dessa forma, por meio de pesquisa e parceria com projeto social, o trabalho buscou elaborar um material didático que pudesse guiar os professores de forma mais simplificada, trazendo tanto o conteúdo específico da área tecnológica quanto o que aquele conteúdo representa na linha de aprendizagem do aluno. A análise das ferramentas digitais também foi voltada para a facilidade de uso e entendimento, tanto do aluno quanto do professor, e em relação ao custo-benefício que profissionais e escolas poderiam aproveitar. Além disso, com a participação de profissionais da educação, com um olhar mais analítico nas práticas pedagógicas, foi possível alinhar o material didático com as necessidades e expectativas do aluno.

REFERÊNCIAS BIBLIOGRÁFICAS

- (1) AIX Sistemas. Aprender, fazendo! O conceito de Seymour Papert e sua contribuição para a educação infantil. Blog da Educação Infantil, 2021. Disponível em: <https://educacaoinfantil.aix.com.br/seymour-papert/>. Acesso em: 20 jun. 2022
- (2) PAPERT, S. Mindstorms: Children, Computers and Powerful Ideas. 2nd. ed. [S.l.]: [s.n.], 1980.
- (3) Jean Piaget: conheça a história do criador da teoria Construtivista. Escola Santi, 2021. Disponível em: <https://escolasanti.com.br/jean-piaget-conheca-a-historia-do-criador-da-teoria-construtivista/>. Acesso em: 30 jun.2022
- (4) Entenda o construtivismo na educação. escola da inteligência, 2018. Disponível em: <https://escoladainteligencia.com.br/blog/construtivismo-na-educacao/> Acesso em: 30 jun. 2022
- (5) Seymour Papert: O Construcionismo, Uma pequena biografia. Slideshare, 2010 Disponível em: <https://pt.slideshare.net/lat730/construcionismo-seymour-papert>. Acesso em: 30 jun. 2022
- (6) BARBOSA, Priscila Maria Romero. O Construtivismo e Jean Piaget. Revista Educação Pública, 2015. Disponível em: [https://educacaopublica.cecierj.edu.br/artigos/15/12/o-construtivismo-e-jean-piaget#:~:text=O%20Construtivismo%20afirma%20que%20o,aluno%20\(Fossile%2C%202010\)](https://educacaopublica.cecierj.edu.br/artigos/15/12/o-construtivismo-e-jean-piaget#:~:text=O%20Construtivismo%20afirma%20que%20o,aluno%20(Fossile%2C%202010).). Acesso em: 30 jun. 2022
- (7) Alves Antonio Neto, Luiz. Estudos de caso com computação física para o ensino de pensamento computacional para o público infantil. Monografia (Graduação em Engenharia Mecatrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2018
- (8) TARSSO GOMES SANTOS, J.; FELIPE SILVA DE LIMA, J. Robótica Educacional e Construcionismo como proposta metodológica para o desenvolvimento de ambientes de aprendizagem significativa. RENOTE, Porto Alegre, v. 16, n. 2, p. 596–605, 2018. DOI: 10.22456/1679-1916.89300. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/89300>. Acesso em: 5 jul. 2022.
- (9) What is Arduino?. Arduino, 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 24 mai. 2022

- (10) Arduino Uno Rev3. Arduino Online Shop. Disponível em: <https://store-usa.arduino.cc/collections/boards/products/arduino-uno-rev3> Acesso em: 24 mai. 2022
- (11) ALJUNDI, Liam, Using the Arduino Software (IDE). Arduino Docs, 2022. Disponível em: (<https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide>. Acesso em: 24 mai. 2022
- (12) Kit BBC micro:bit V2 e Acessórios. Robocore, 2022. Disponível em: https://www.robocore.net/placa-microbit/kit-microbit-v2-e-acessorios?gclid=Cj0KCQjw2MWVBhCQARIsAIjbwoOtTTyrjYBnSCY6CFLsLa8Rt2jwhO_TkFBX5P1rBaFpn8IKxxfSFu0aAh5gEALw_wcB. Acesso em: 24 mai. 2022
- (13) Microsoft MakeCode for micro:bit, 2022. Disponível em: <https://makecode.microbit.org/#editor>. Acesso em: 24 mai. 2022
- (14) About Us. Raspberry Pi Foundation, 2022. Disponível em: <https://www.raspberrypi.org/about/>. Acesso em: 24 mai. 2022
- (15) Raspberry Pi 3 Model B+ Anatel. FilipeFlop, 2022. Disponível em: (<https://www.filipeflop.com/produto/raspberry-pi-3-model-b/>. Acesso em: 24 mai. 2022
- (16) TinkerCard. Autodesk Tinkercad, 2022. Disponível em: <https://www.tinkercad.com/>. Acesso: 24 mai. 2022
- (17) Sobre S4A. Disponível em: http://s4a.cat/index_pt.html. Acesso em: 24 mai. 2022
- (18) ABAC – Associação Brasileira de Aprendizagem Criativa. Aprendizagem Criativa na Prática: A Experiência do Desafio de Aprendizagem Criativa Brasil 2018. São Paulo, 2019. Disponível em <http://aprendizagemcriativa.org/dac2018> Acesso em: 3 jul. 2022
- (19) CAMPOS, F. R. Robótica educacional no Brasil: questões em aberto, desafios e perspectivas futuras. Revista Ibero-Americana de Estudos em Educação, Araraquara, v.12, n.4, p. 2108-2121, out./dez. 2017. Disponível em: <https://periodicos.fclar.unesp.br/iberoamericana/article/view/8778/6944> Acesso em: 03 jul. 2022
- (20) Redação. Robótica nas escolas: impacto pedagógico e futuro profissional. Revista Veja, 2022. Disponível em: <https://veja.abril.com.br/insights-list/robotica-nas-escolas-impacto-pedagogico-e-futuro-profissional/>. Acesso em: 03 jul. 2022

- (21) Redação. A maior vantagem competitiva é a habilidade de aprender. Revista Super Interessante, 2016. Disponível em: <https://super.abril.com.br/tecnologia/a-maior-vantagem-competitiva-e-a-habilidade-de-aprender/>. Acesso em: 3 jul. 2022
- (22) Robótica educacional no Brasil / vários autores ; organizado por Brehme D'napoli Reis de Mesquita. -Ananindeua : Itacaiúnas, 2021. Disponível em: <https://edicoes.editoraitacaiunas.com.br/index.php/ea/catalog/view/9/9/25>. Acesso em: 24 mai. 2022
- (23) Creative Computing Curriculum. Creative Computing Lab, 2011. Disponível em: <https://creativecomputing.gse.harvard.edu/guide/curriculum.html>. Acesso em: 5 out. 2021
- (24) Scratch - Imagine, Program, Share. Fundação Scratch. Disponível em: <https://scratch.mit.edu/>. Acesso: 5 out. 2021

APÊNDICE A – Material didático do curso “Robótica Legal”

MATERIAL DO CURSO “ROBÓTICA LEGAL”

Apresentação do Curso

Seja bem-vindo(a) ao material do curso Robótica Legal!

Para introduzi-lo, é importante destacar alguns aspectos gerais antes de apresentar as aulas em si.

O que é esse material?

É um conjunto de planos de aulas, que se baseia na introdução à programação em blocos e à eletrônica básica por meio do Tinkercad e do Arduino. Tinkercad é uma plataforma web gratuita que possibilita a criação de projetos 3D, eletrônicos e de codificação de modo acessível. Já o Arduino é uma plataforma relacionada à eletrônica baseada em hardware e software com interface acessível e de princípio colaborativo.

Para quem é destinado esse material?

Para professores que gostariam de dar aulas com esses conteúdos, mas não necessariamente tem conhecimento prévio sobre programação e eletrônica.

Por que esse material foi criado?

Porque acreditamos que seja cada vez mais importante que crianças e adolescentes tenham contato com ciência e tecnologia de forma ativa no ambiente escolar para desenvolver raciocínio lógico, capacidade de resolução de problemas, pensamento crítico, trabalho colaborativo, entre outros. E, ao mesmo tempo que os educadores sabem da importância dessa área, muitos têm receio em trazer tecnologia como ferramenta e conteúdo. Por isso, o propósito do material é buscar orientar educadores no desenvolvimento de aulas que trazem tecnologia para dentro da sala de aula (nesse caso, introdução à programação e à eletrônica) baseando-se em metodologias ativas.

Quais são os recursos necessários para o curso?

Para a maioria das aulas, são necessários computadores com acesso à internet e Arduinos com componentes básicos. Cada aula pode ter itens mais específicos por conta do conteúdo abordado.

Como está estruturado o material?

Em 4 guias e 14 planos de aulas sugeridos. Os guias apresentam os primeiros passos relacionados à instalação, cadastro e recursos gerais do Tinkercad e o Arduino. Além disso, um dos guias serve como material de consulta de boas práticas na criação de circuitos.

Em relação às aulas, a primeira parte apoia-se na introdução às plataformas, revisão de conceitos de elétrica e aplicação de conceitos de programação e eletrônica. Já na segunda parte, propõe-se a construção de um projeto que engloba a idealização até a prototipagem e apresentação.

É importante ressaltar que o Arduino e principalmente o Tinkercad possuem atualizações periódicas com o intuito de aprimorar e acrescentar as ferramentas. Nesse sentido, pode ser que alguns recursos tenham sido alterados, mas a base do funcionamento muito provavelmente se mantém.

Além disso, destaca-se que as aulas buscam orientar o professor no desenvolvimento das dinâmicas das aulas, mas é recomendável adaptá-las de acordo com o perfil da turma.

Vamos começar?

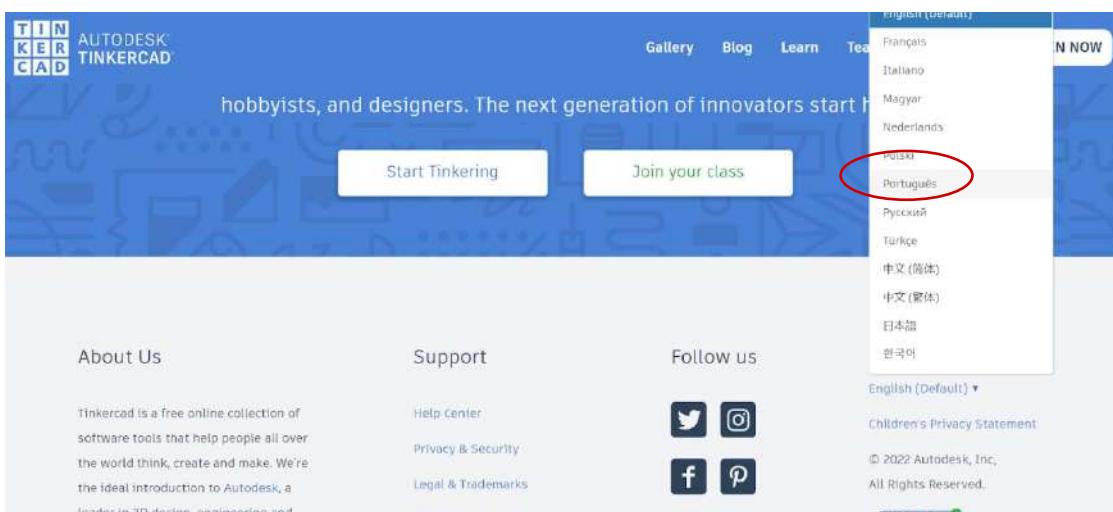
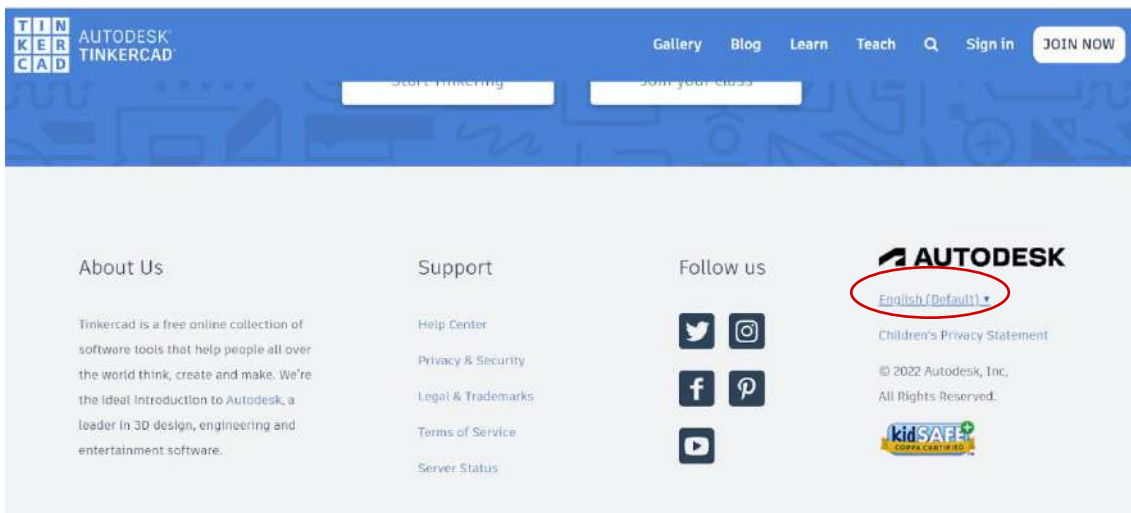
Guia 1: como se registrar e criar um novo projeto no Tinkercad

I. Registrar na plataforma

Para usar os recursos que o Tinkercad oferece, é necessário se registrar no site. Dessa forma, ao digitar no navegador: <https://www.tinkercad.com/>, essa página é aberta:



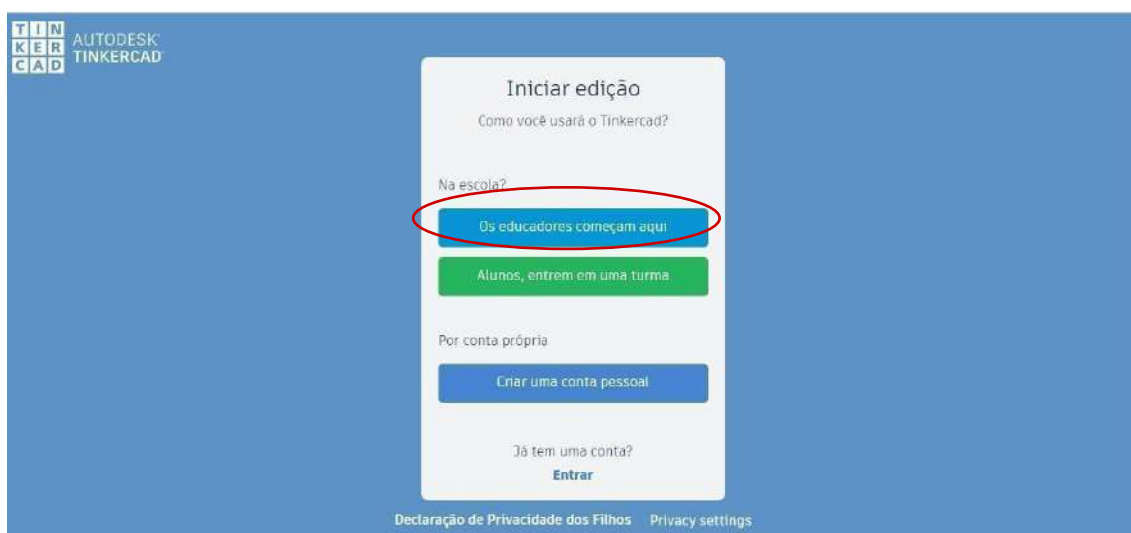
Caso a página esteja em inglês, é possível trocar para português. Para isso, vá até o final da página, clique em “English (default)” e selecione a opção “Português”.



Ao longo da página, aparecerem mais informações sobre as ferramentas do site. Para iniciar o registro na plataforma, clique em “Inscreva-se Agora”, na lateral superior direita da página:



Nas janelas seguintes, siga as instruções de inscrição. Clique em “Os Educadores começam aqui” para que seu acesso tenha os recursos habilitados para esse público.



Em seguida, aparecerá um texto explicativo sobre a segurança dos conteúdos criados pelos alunos. Para prosseguir, clique em *Continuar criando minha conta de educador(a)*.



Um pequeno termo aparecerá sobre permissões e Termos de Privacidade em relação aos alunos e seus responsáveis e à própria plataforma Tinkercad. Após ter lido, confirme a caixa de seleção referente ao termo e aperte “Concordo”.

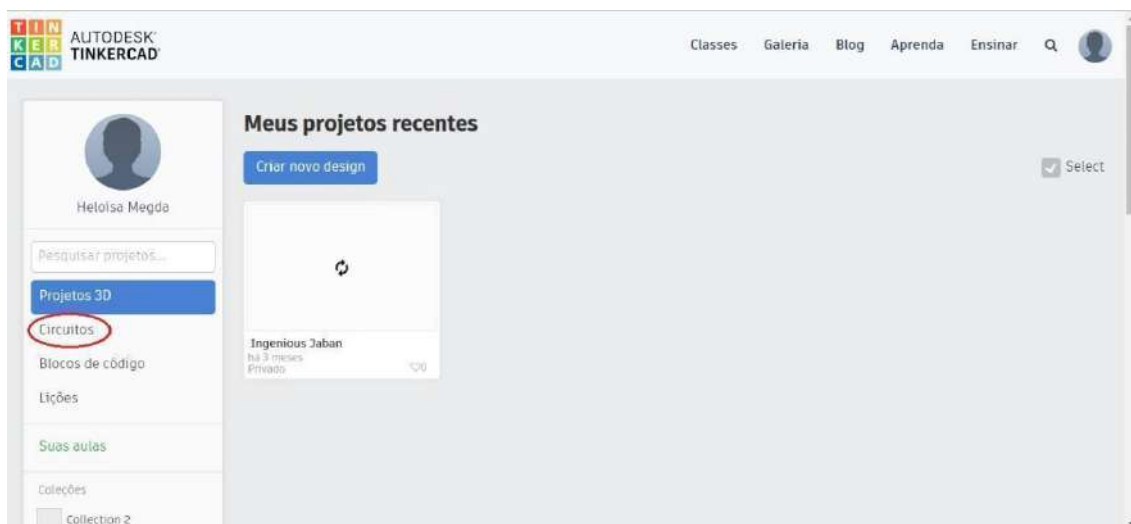


Em seguida, siga com as etapas de inscrição do e-mail. É possível entrar com conta do Google, Apple ou outro domínio.

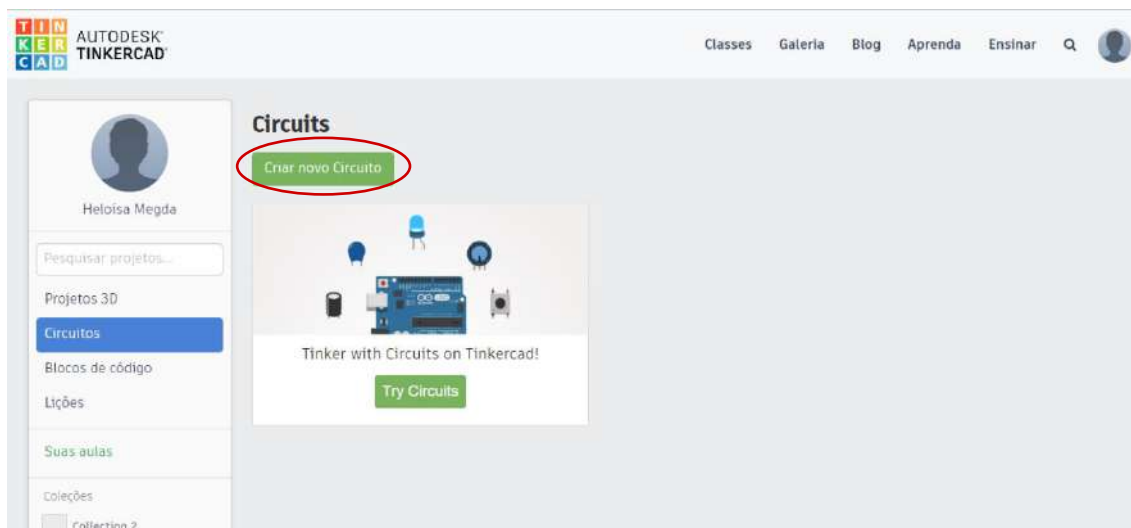


II. Painel

Após ter feito processo de inscrição no Tinkercad, aparecerá o painel, no qual se encontram todos os projetos criados pelo usuário. No canto superior direito, pode-se conhecer mais a plataforma, tanto com conteúdos disponibilizados pelo próprio Tinkercad, como “Blog”, “Aprenda” e “Ensinar”, quanto conteúdos que outros usuários criam, disponíveis no item “Galeria”. O recurso “Classes”, que será abordado no Guia 2, é de extrema relevância para o educador, pois permite acompanhar o desenvolvimento dos projetos de sua turma. Já no lado esquerdo, os projetos são organizados de acordo com as ferramentas de interesse. Para os fins estipulados neste curso, clique no item de interesse “Circuitos”:



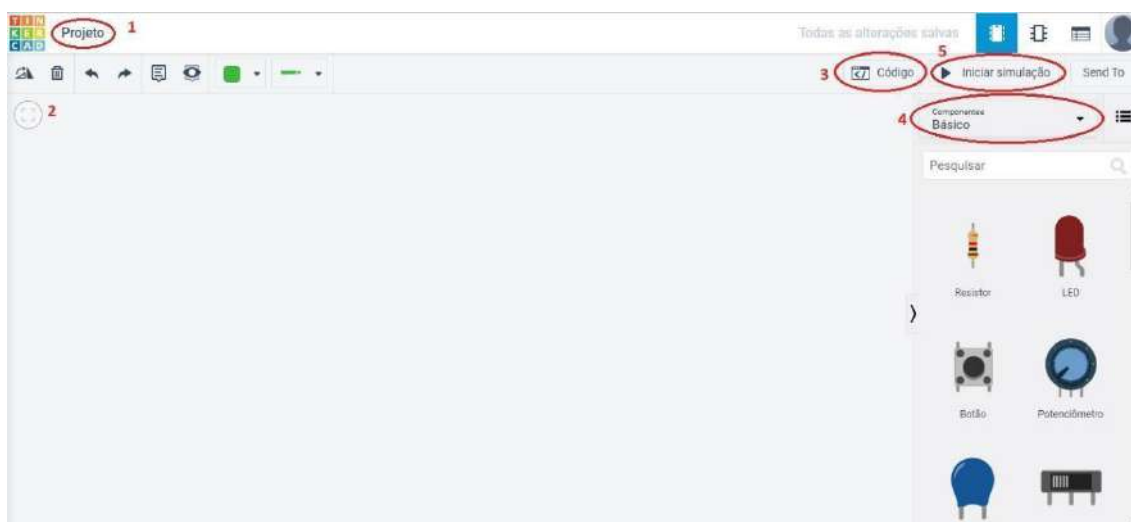
É aqui que os projetos de circuitos estarão disponíveis. Para criar um novo projeto, clique no botão em verde “Criar novo Circuito”.



III. Criando um novo Circuito

É nessa página que as aulas acontecem, desde a montagem do circuito, passando pela programação em blocos, até a simulação final.

Para melhor entendimento, os principais espaços e funcionalidades estão listados e explicados a seguir:



1 - Título do projeto: É nesse espaço que será nomeado o projeto. Ao criar um projeto, um título aleatório é gerado, mas é recomendável renomeá-lo;

2 - Área de montagem do circuito: É onde se cria o circuito, arrastando os componentes para essa área;

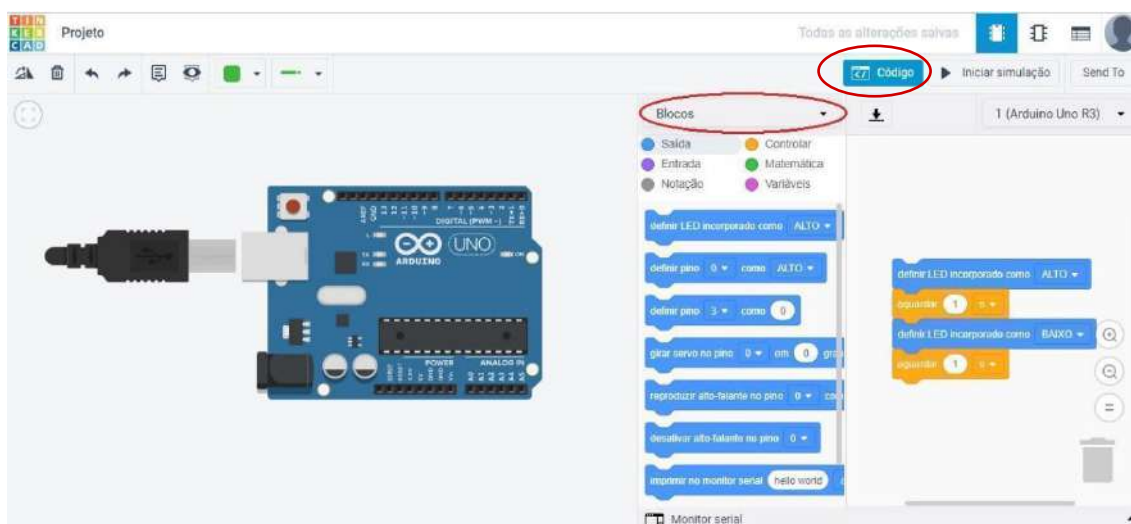
3- Botão “Código” - É o botão que, quando clicado, vai para área que se desenvolver a lógica de programação, por meio de texto ou por blocos;

4 - Aba “Componentes”: Essa área mostra todos os componentes que a plataforma possui, desde componentes básicos até microcontroladores, como no caso do Arduino. São esses componentes que quando arrastados e conectados na área de montagem, dão vida ao projeto. Clicando no ícone à direita dos componentes, é possível trocar da vista de grade para a vista em lista, a qual exhibe os componentes com breves descrições de cada componente;

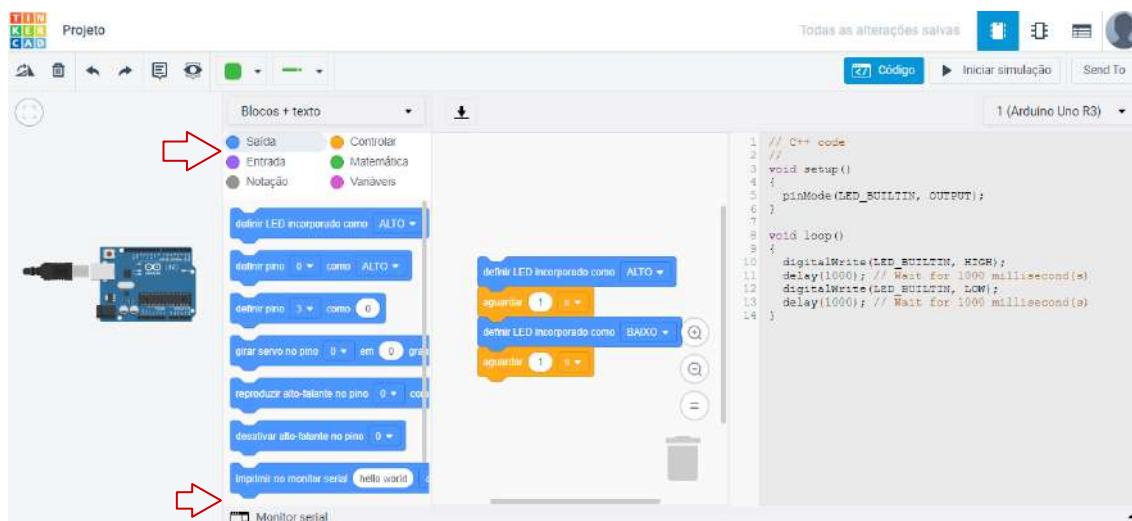
5- Botão “Iniciar Simulação” - Depois de todo o circuito montado e a programação estruturada, é hora de ver o projeto funcionando. Ao clicar no botão, verifica-se se é necessário algum ajuste ou não.

IV. Aprofundando na programação

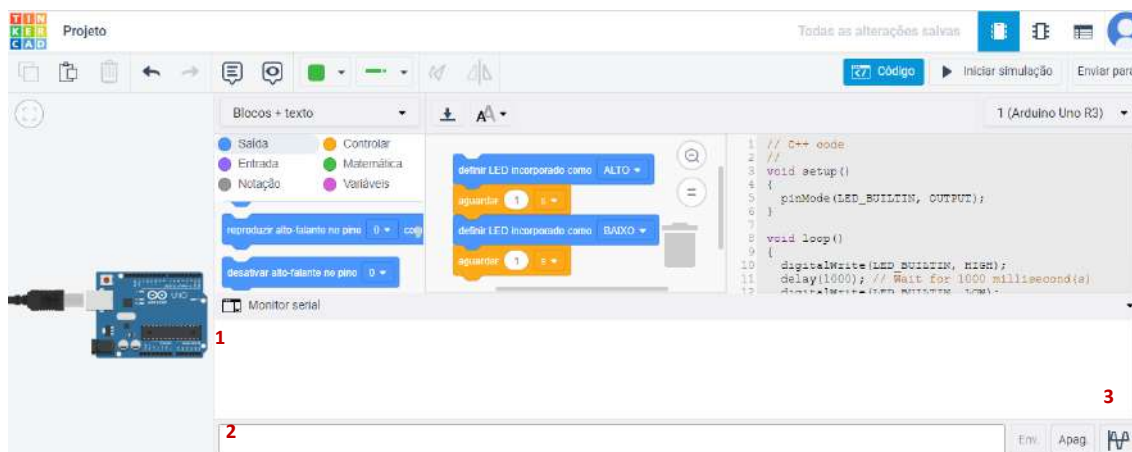
Ao apertar o botão “Código”, a plataforma traz algumas formas para criar a programação do projeto. Como o foco deste curso é programação por blocos, iremos focar nessa configuração. Quando o Arduino Uno R3 é selecionado da Aba de Componentes e arrastado para a Área de Montagem, o Botão Código, quando clicado, mostra a seguinte disposição:



É nessa página que aparecem os blocos de programação que, quando selecionados do espaço à esquerda, arrastados para a direita e conectados seguindo uma lógica, fazem o projeto funcionar. Pela mesma aba, é possível também visualizar o texto em linguagem C++ enquanto programa em blocos selecionando *Blocos + texto*. Além disso, tem o recurso de *Monitor serial* na parte inferior nessa aba. Ao clicar, a visão do monitor é expandida.



Quando programamos com os blocos de entrada e saída relacionados ao monitor serial, ele pode exibir dados enviados ou recebidos em formato de texto no espaço 1 e/ou receber dados que preenchemos no espaço 2 e clicamos no quadrado ao lado *Env*. O ícone 3 mostra a representação gráfica quando se programa a exibição dos dados recebidos.



V. Blocos de programação

Para programação em blocos, o Tinkercad oferece 6 tipos de blocos: Saída, Entrada, Notação, Controlar, Matemática e Variáveis. Esses tipos de blocos caracterizam-se por:

- Saída: comandos que realizam uma ação
- Entrada: comandos que fazem a leitura de algo

- Notação: comentários que auxiliam no entendimento da lógica do programa, mas não interferem no funcionamento do código
- Controlar: comandos relacionados ao tempo, à repetição, etc.
- Matemática: comandos relacionados a operações matemáticas e lógicas
- Variável: comandos que definem uma variável criada pelo usuário

Obs: Na aula 4, esse tópico é abordado mais detalhadamente com exemplos e aplicações.

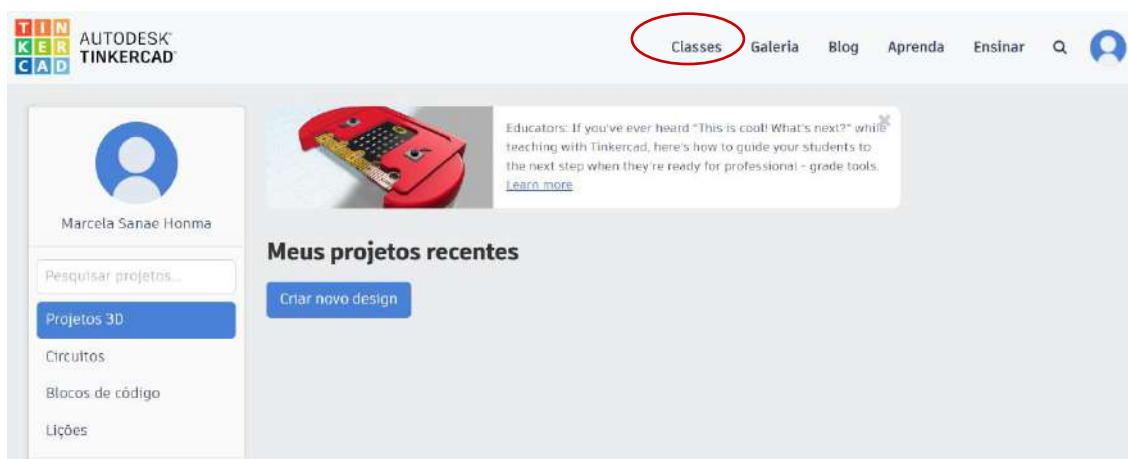
Pronto! Você já sabe as principais funcionalidades do Tinkercad circuitos.

Guia 2: Como criar uma sala de aula no Tinkercad

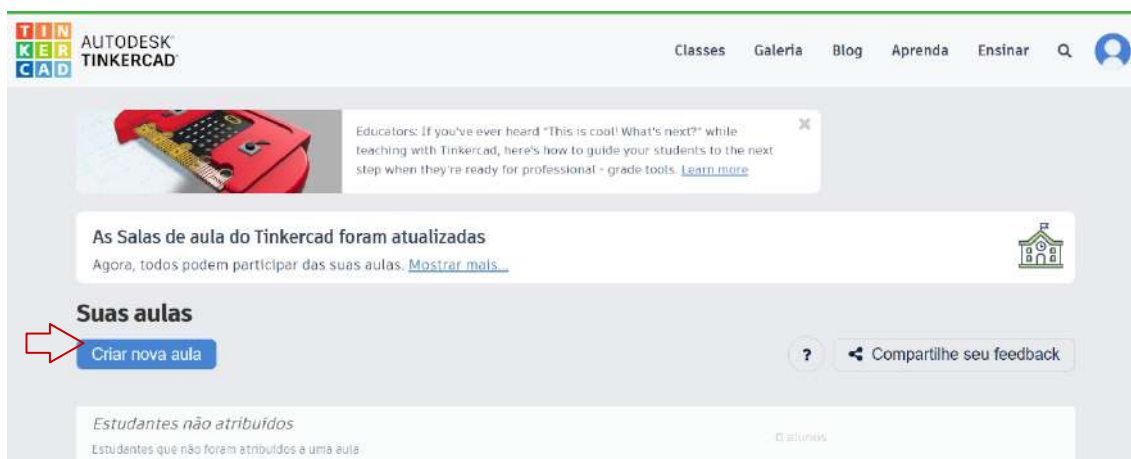
No Tinkercad, existe a funcionalidade da “sala de aula”, na qual o professor pode criar um ambiente virtual para a turma, inserindo atividades e acompanhando o andamento de projetos. Além disso, é possível incluir co-professores. Neste guia, mostraremos como criar uma sala de aula e uma atividade.

I. Criar e configurar uma classe

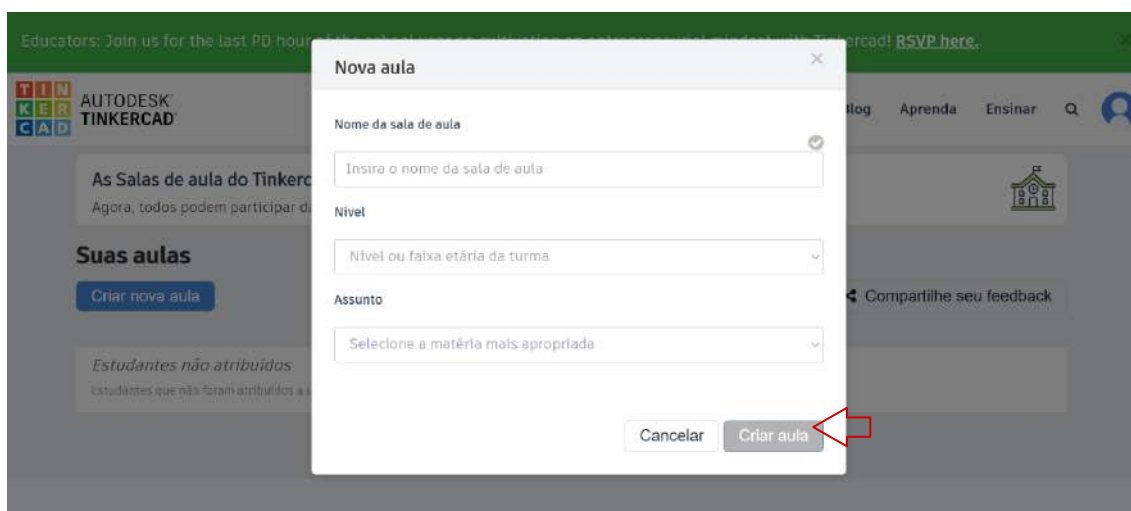
Ao se logar na plataforma, aparecerá o painel inicial, similar ao mostrado abaixo. Clique em *Classes* no canto superior direito.



Aqui, você pode criar sala(s) de aula(s) virtuais! Basta clicar no botão *Criar nova aula*.



Surgirá uma tela para inserir algumas informações básicas da sala de aula e, por fim, clicar em *Criar aula*.



Pronto! A nova sala de aula virtual está criada! Da próxima vez que quiser acessá-la, você clica em *Classes* no painel e a(s) sala(s) criada(s) anteriormente já estará(ão) nessa tela principal. Caso você precise modificar alguma informação básica, clique no ícone da engrenagem à direita, faça a modificação e clique em *Salvar alterações*.

AUTODESK TINKERCAD Classes Galeria Blog Aprenda Ensinar Q

As Salas de aula do Tinkercad foram atualizadas
Agora, todos podem participar das suas aulas. [Mostrar mais...](#)

Suas aulas

[Criar nova aula](#) ? [Compartilhe seu feedback](#)

| | | |
|----------------------------------|----------|-----------------------------|
| Exemplo de sala de aula | 0 alunos | Data de criação: 18/03/2022 |
| <i>Estudantes não atribuídos</i> | 0 alunos | |

Propriedades da sala de aula

Nome da sala de aula*
Exemplo de sala de aula

Nível
14 a 15 anos | 9ª série

Assunto
Tecnologia

Excluir sala de aula
A exclusão desta turma removerá todos os assentos da Turma e seus dados. As contas da turma não serão excluídas. Elas serão movidas para a turma "Não atribuída".

Fechar [Salvar alterações](#)

Para entrar na sala de aula, clique na faixa com o nome da sala.

AUTODESK TINKERCAD Classes Galeria Blog Aprenda Ensinar Q

As Salas de aula do Tinkercad foram atualizadas
Agora, todos podem participar das suas aulas. [Mostrar mais...](#)

Suas aulas

[Criar nova aula](#) ? [Compartilhe seu feedback](#)

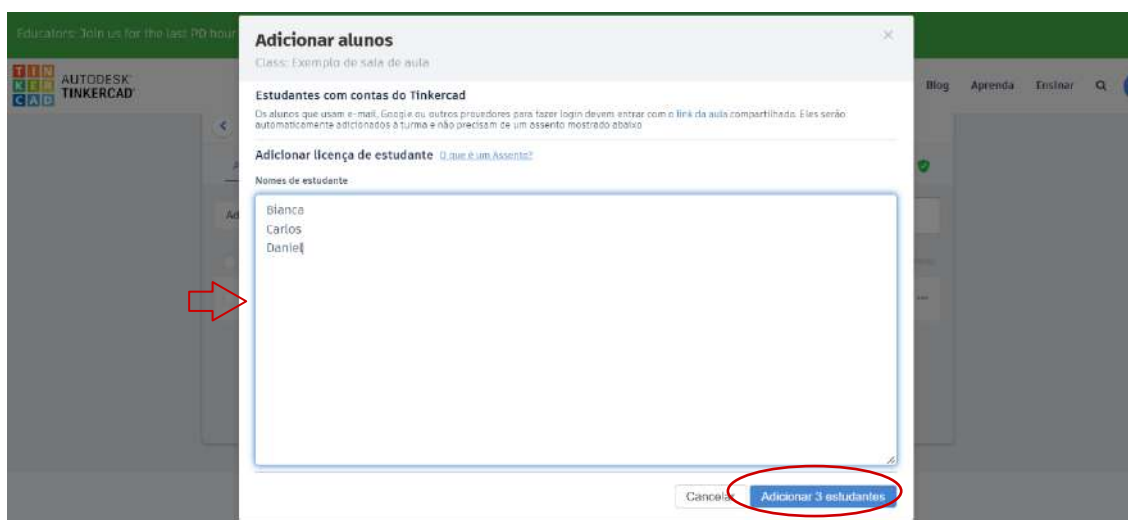
| | | |
|----------------------------------|----------|-----------------------------|
| Exemplo de sala de aula | 0 alunos | Data de criação: 18/03/2022 |
| <i>Estudantes não atribuídos</i> | 0 alunos | |

Você pode começar incluindo os alunos da turma na guia *Alunos*. Para isso, clique no botão *Adicionar alunos*.

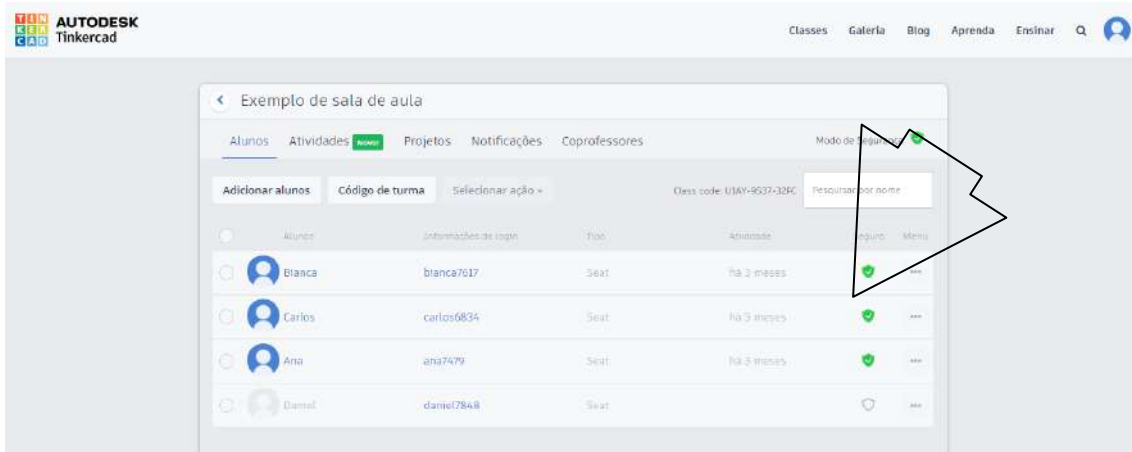
Para incluir os alunos, você pode adicionar um nome por vez e clicar no botão *Salvar alterações* (o apelido é gerado automaticamente pela ferramenta).



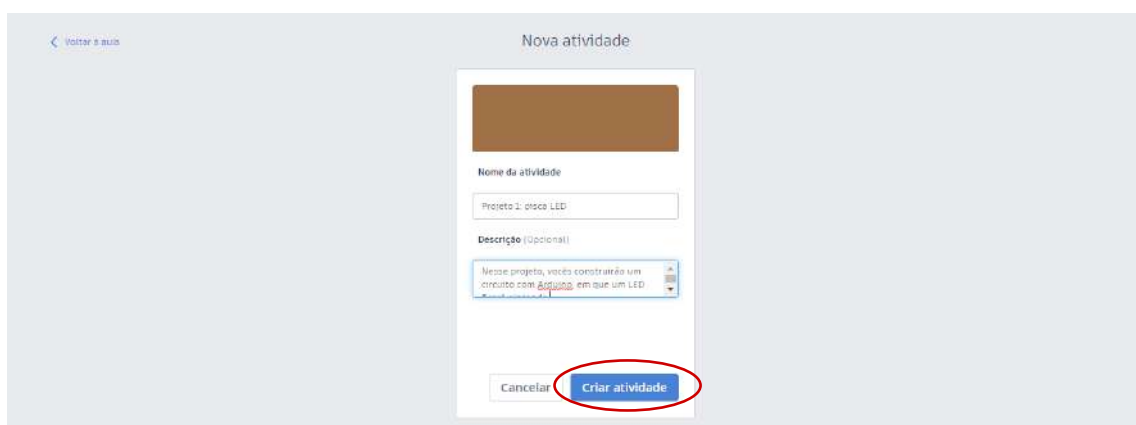
Ou ainda, se tiver vários alunos, pode-se adicionar todos de uma vez só, clicando no botão *Colar uma lista de estudantes*, digitando os nomes dos alunos, um por linha, e clicando em *Adicionar x estudantes*.



Após ter adicionado todos os alunos da turma, a tela na guia de *Alunos* deve mostrar o nome do aluno, o apelido e outras informações, como mostrado abaixo. Quando o aluno entrar na sala pela primeira vez, o símbolo de seguro fica verde.

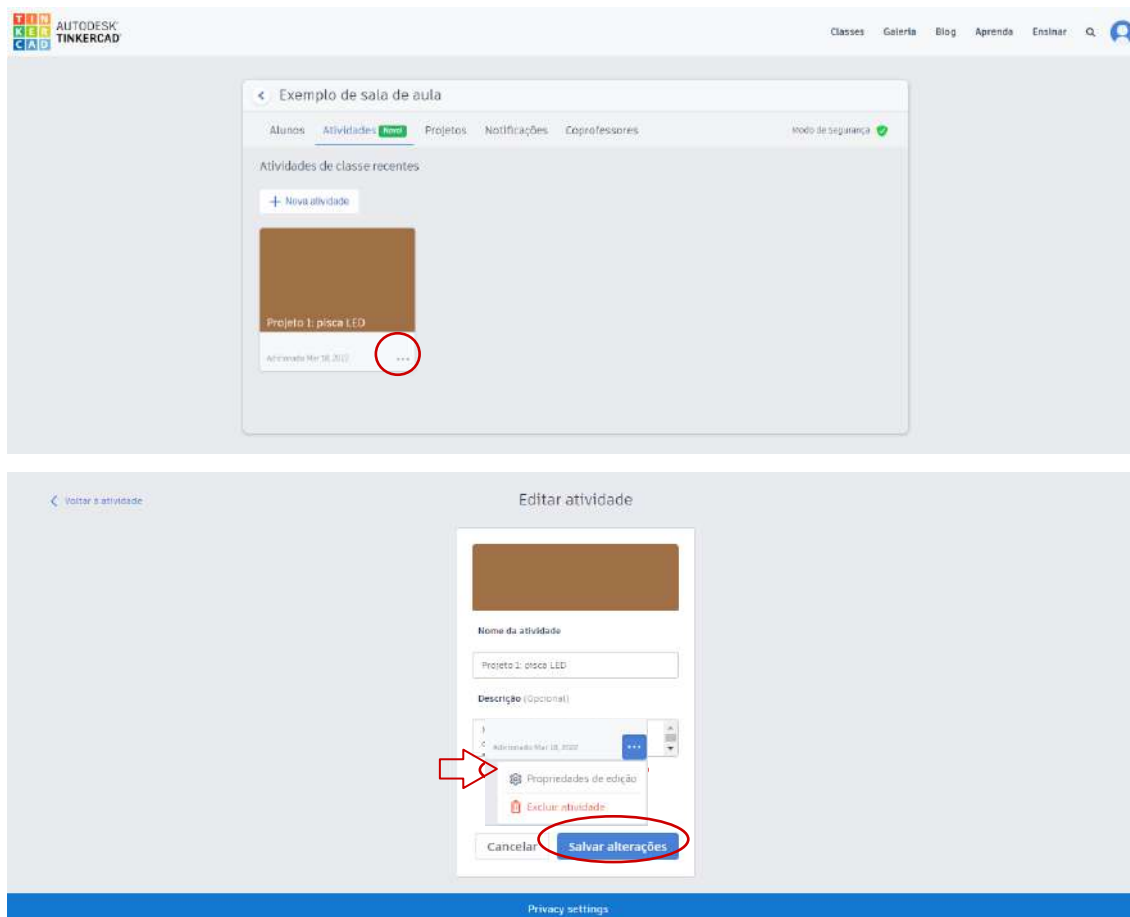


Para criar a primeira atividade, basta ir para a guia de *Atividades*, clicar no quadrado *Criar sua primeira atividade*, inserir informações básicas e clicar no botão *Criar atividade*.



A(s) atividade(s) criada(s) ficará(ão) exposta(s) nessa guia. Assim como na sala de aula, se precisar alterar alguma informação básica, clique nos 3 pontos do quadrado da atividade, depois

em *propriedades de edição*, faça a mudança e clique em *Salvar alterações*.

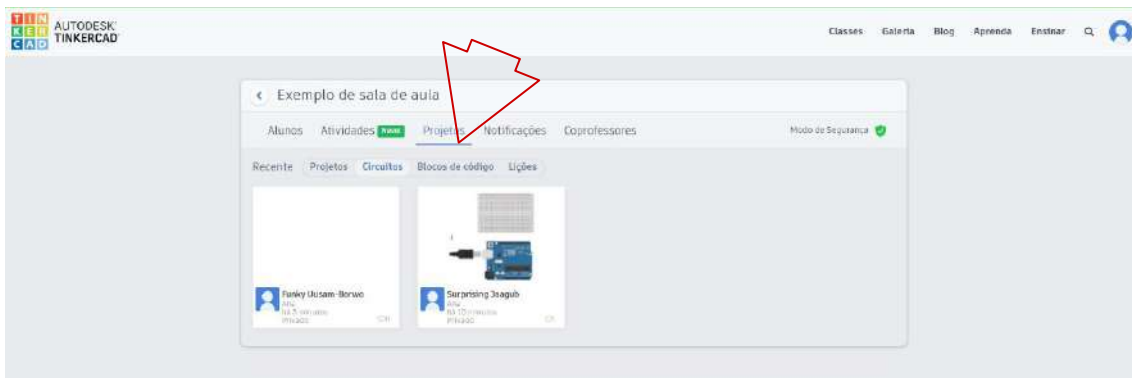


Clicando na atividade criada, aparecerá uma nova tela com as informações inseridas até agora. O ambiente apresenta uma primeira área de projeto(s) que você pode compartilhar com a turma e uma segunda área com os trabalhos dos alunos, organizados por categoria (Projetos 3D, Circuitos, Blocos de códigos ou Lições). Esses trabalhos podem ter sido construídos dentro da sala de aula do Tinkercad ou não.

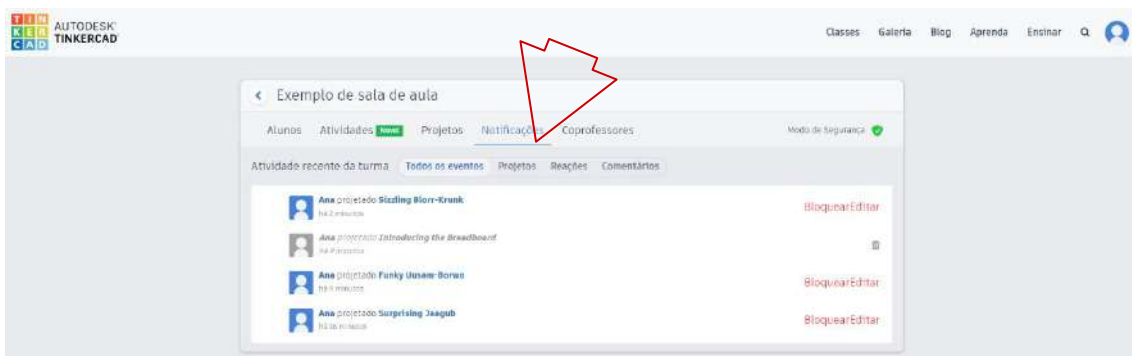
Para você criar um design nessa atividade, clique no botão + *Criar novo design*, escolha o tipo (no nosso caso, *Circuitos*) e seja direcionado para o típico ambiente de criação do Tinkercad. Esse projeto ficará automaticamente no espaço de *Projetos compartilhados com estudantes*, em que eles podem visualizar ou copiar e usar.



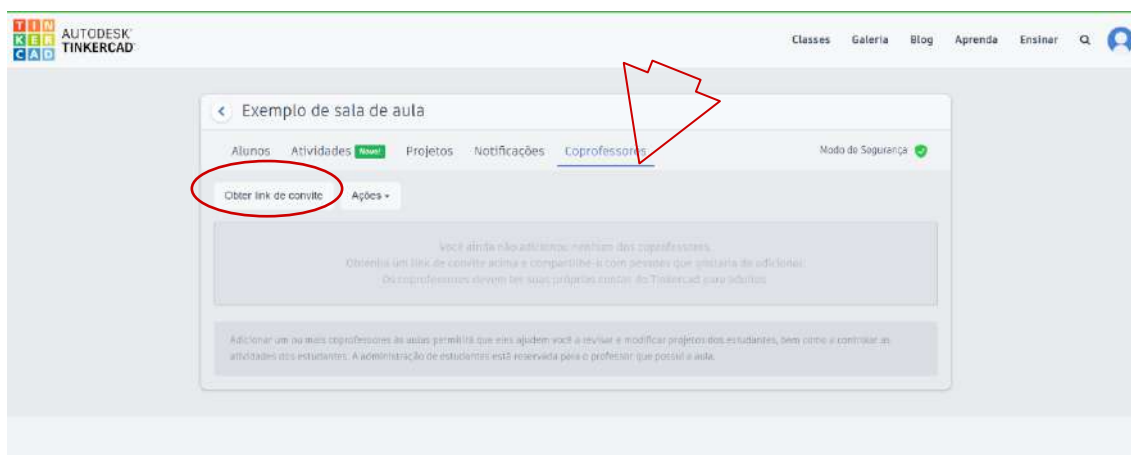
A guia *Projetos* é parecida com a de *Atividades*, porém não aparecem os projetos do professor e não é organizado por atividade. Os trabalhos dos alunos aparecem por categoria e horário de criação do design.



Na guia de *Notificações*, o professor tem acesso ao histórico de ações dos alunos, como mostrado abaixo.



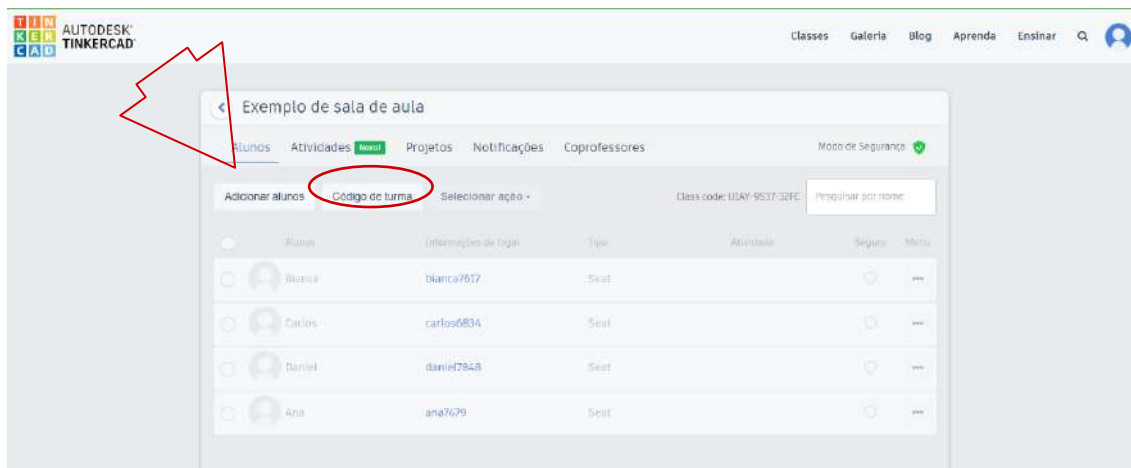
Por último, se você quiser adicionar mais professores na sala, vá para guia de *Co-professores* e depois em *Obter link de convite*. O link será copiado e você poderá enviar para que o(s) demais(s) professor(es) acessem.



II. Acessar a classe

Para que o aluno acesse a sala de aula criada pelo professor, há dois modos de fazê-lo: um mais prático a partir do link da sala e um outro mais extenso por meio do código.

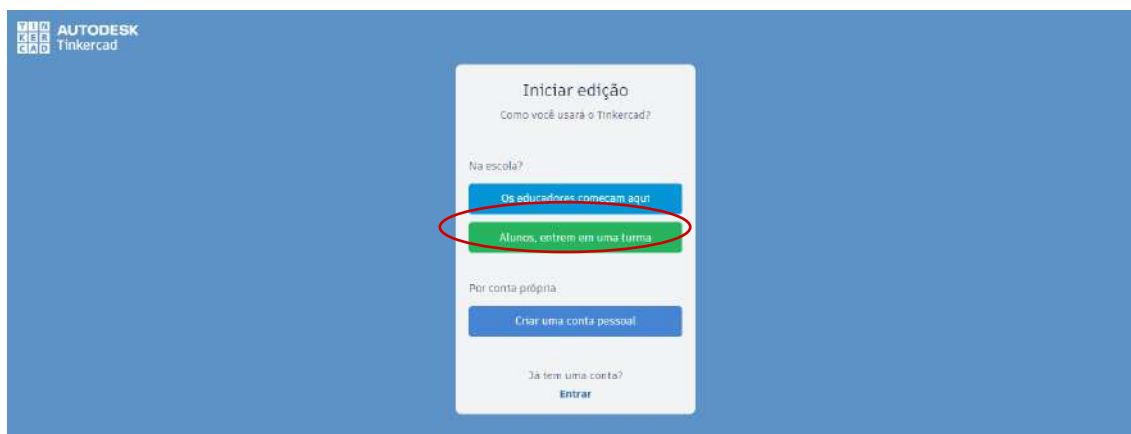
Primeiro, o professor deve fornecer ao aluno o código ou o link da sala de aula criada. Para isso, entre no ambiente da sala de aula e clique no botão *Código de turma* da guia *Alunos*.



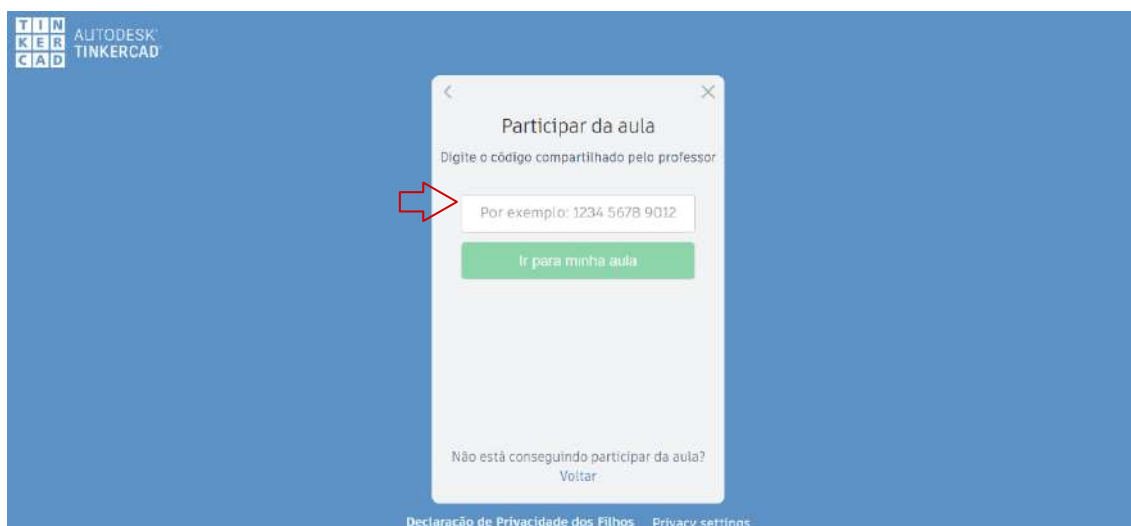
Aparecerá uma nova tela, em que você poderá copiar o código ou o link da sala. Recomendase utilizar o link por conta da redução de passos, mas serão explicados os dois modos.



Para acessar com o código, o aluno deverá seguir um fluxo similar ao do educador: acessar <https://www.tinkercad.com/> no navegador, mudar para o idioma português no final da página, clicar em *Inscreva-se agora* na faixa fixada na parte superior da tela. Dessa vez, deve-se clicar no botão *Alunos, entrem em uma turma*.

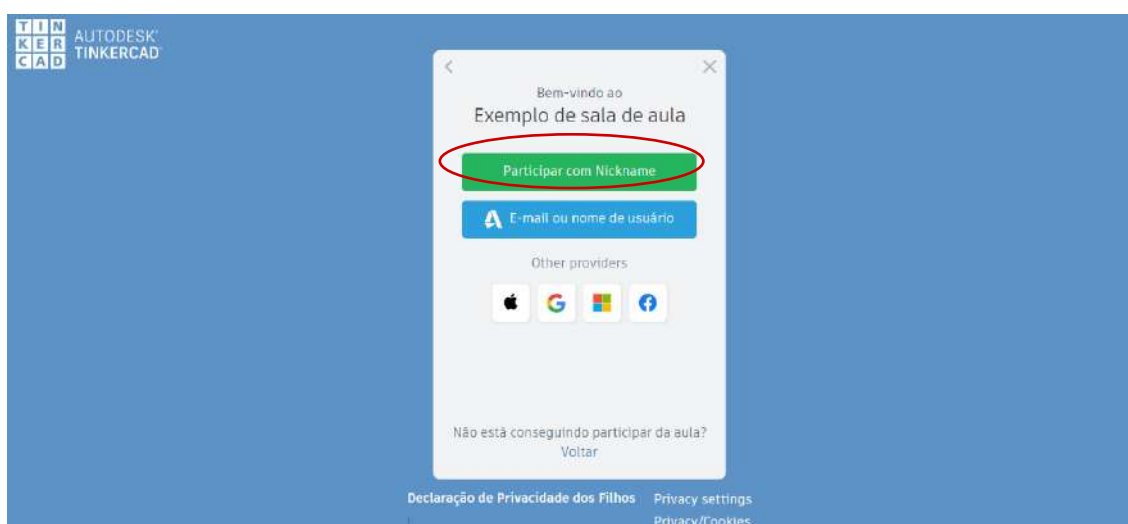


Aqui o aluno digita ou cola o código da aula fornecido pelo professor e depois clica em *Ir para minha aula*.

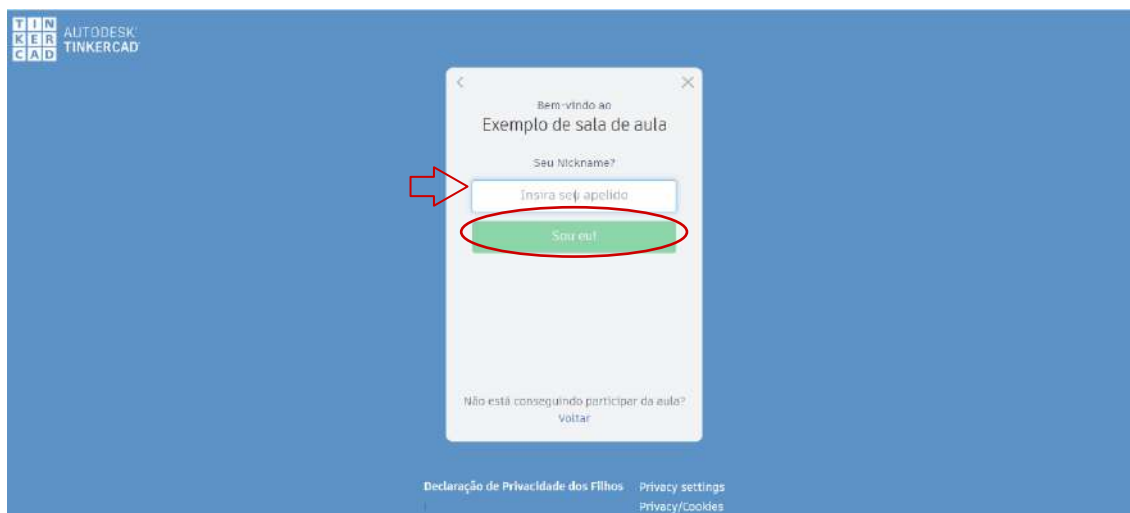


A próxima tela é a identificação do aluno. Caso você tenha fornecido o link da sala de aula, o aluno chega direto nessa página e, por isso, é mais prático.

Aqui o aluno tem 3 opções de acesso. Caso opte por acessar com o apelido, o aluno deve clicar em *Participar com nickname*.

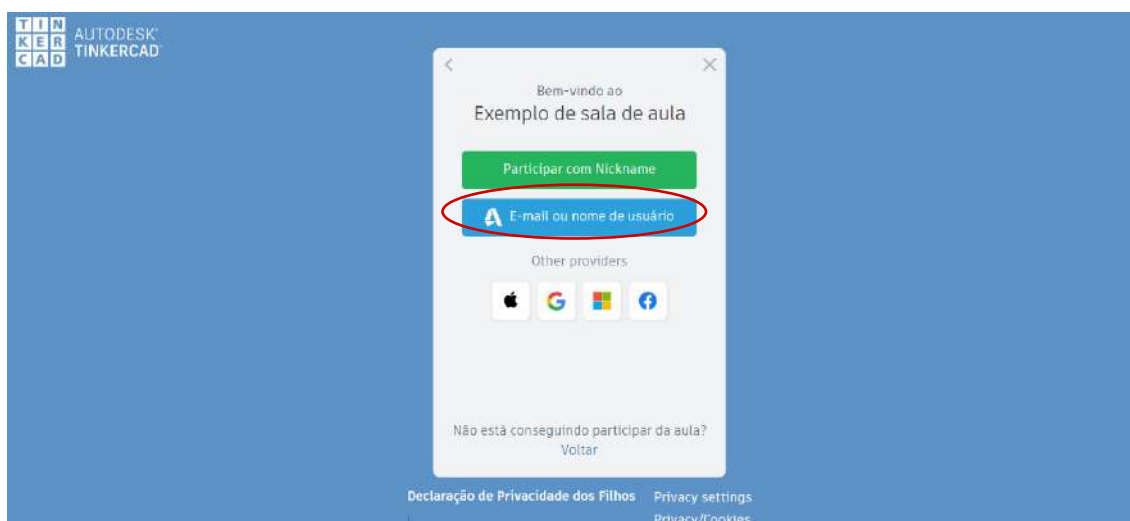


Compartilhe o apelido que foi gerado pela plataforma a partir da inclusão de alunos na sala para que o aluno digite e clique no botão *Sou eu!*, como mostrado abaixo.



O aluno já será direcionado para o painel inicial.

Caso queira criar um acesso do aluno com e-mail e senha específicos para o Tinkercad, o aluno deve clicar em *E-mail ou nome do usuário*.

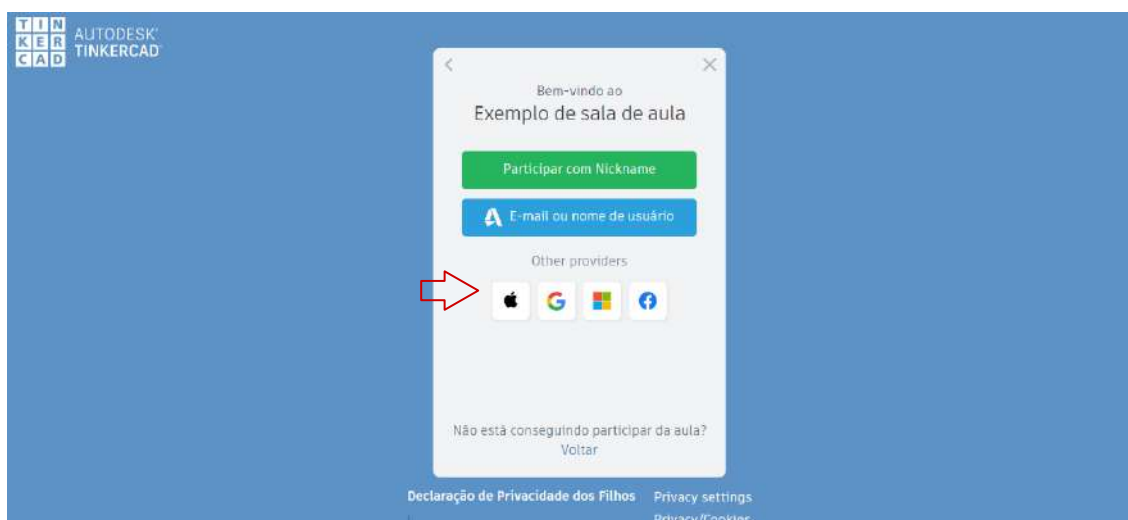


Provavelmente o aluno será novo na plataforma, então é necessário clicar em *Criar uma conta*, preencher os dados de país e data de nascimento, clicar em *avançar*, inserir o nome de usuário, a senha e o e-mail do responsável do aluno e, por fim, clicar em *Criar conta*.

The image displays three sequential screenshots of the Tinkercad registration process:

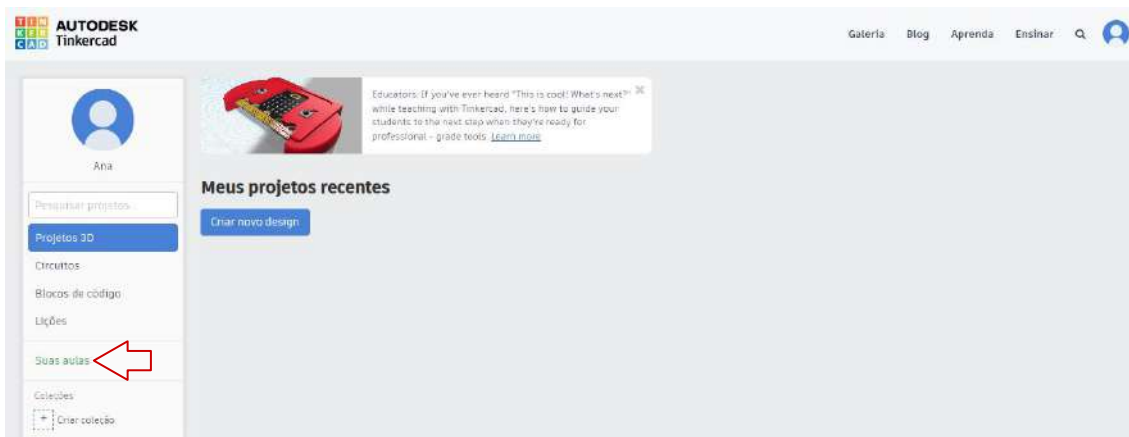
- Left Screenshot:** 'Efetuar login' screen. It features a text input field for 'E-mail ou nome de usuário' and a blue 'AVANÇAR' button. Below the button, there is a link: 'NOVO NA AUTODESK? CRIE UMA CONTA', which is circled in red.
- Middle Screenshot:** 'Criar conta' screen. It includes a dropdown for 'País, território ou região' (set to 'Estados Unidos'), a date picker for 'Aniversário' (Month, Day, Year), and a blue 'AVANÇAR' button. A red arrow points from the 'AVANÇAR' button of the previous screen to this one.
- Right Screenshot:** 'Criar conta' screen. It contains input fields for 'Seu nome de usuário', 'Sua senha', and 'E-mail do seu responsável', followed by a blue 'CRIAR CONTA' button. A red arrow points from the 'AVANÇAR' button of the middle screen to this one.

A última opção é vincular o acesso às credenciais de plataformas como Google, Facebook, Apple, Microsoft, sendo necessário que o aluno já ter uma conta na plataforma ou criar uma. Para isso, o aluno deve clicar no ícone da plataforma, inserir o e-mail e a senha da plataforma, dando acesso desses dados ao Tinkercad.



Em virtude da praticidade, é recomendável que os alunos acessem a sala pelo link e a partir do apelido. Além disso, nessa etapa, é importante que o educador leia os termos de privacidade da plataforma e repasse para os pais e responsáveis informações importantes sobre a política da plataforma.

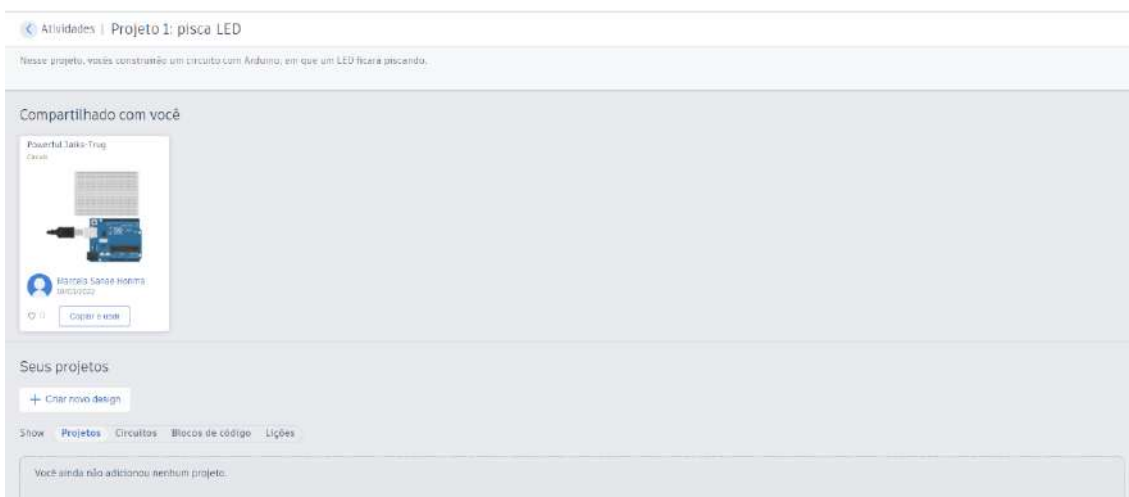
Para acessar a sala de aula, o aluno, depois de ter feito esse processo de inscrição, precisa clicar em *Suas aulas* na coluna à esquerda do seu painel.



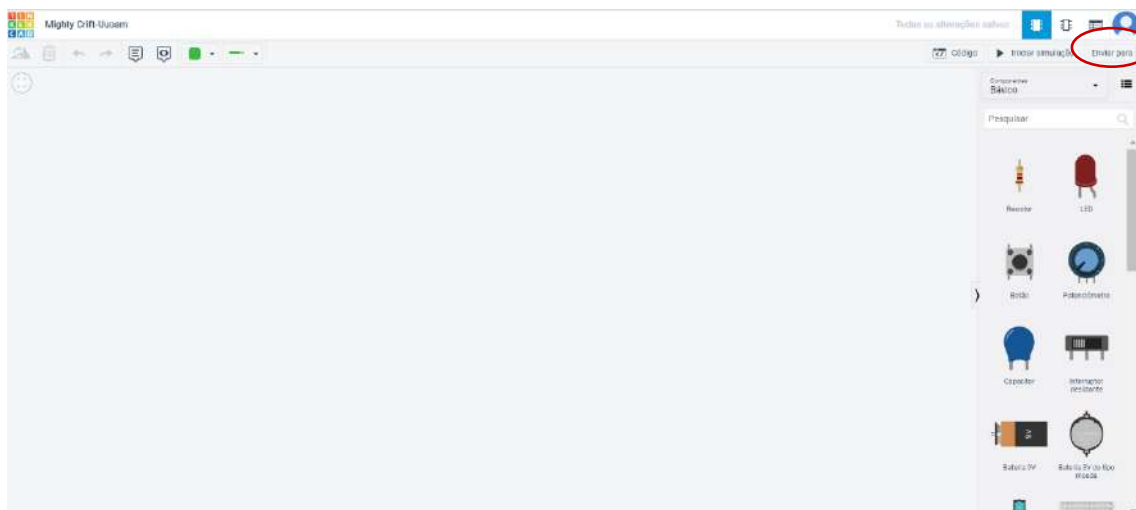
O ambiente da sala de aula do aluno é mais restrito se comparado com a do professor. A(s) atividade(s) da sala de aula são apresentadas na tela inicial.



Ao clicar em uma atividade, observa-se que o aluno tem acesso ao que o professor compartilhou na parte superior e pode criar seus projetos na parte inferior clicando em + *Criar novo design*, sendo apenas compartilhado com professor/co-professor da sala.



O aluno não tem acesso aos projetos dos demais alunos pela sala de aula. Mas se quiserem compartilhar, tem a função *Enviar para* no canto superior direito da área de criação. Assim, pode-se compartilhar de vários formatos, inclusive como imagem e link.



Pronto! Agora você e seus alunos podem acessar e criar projetos em uma sala de aula!

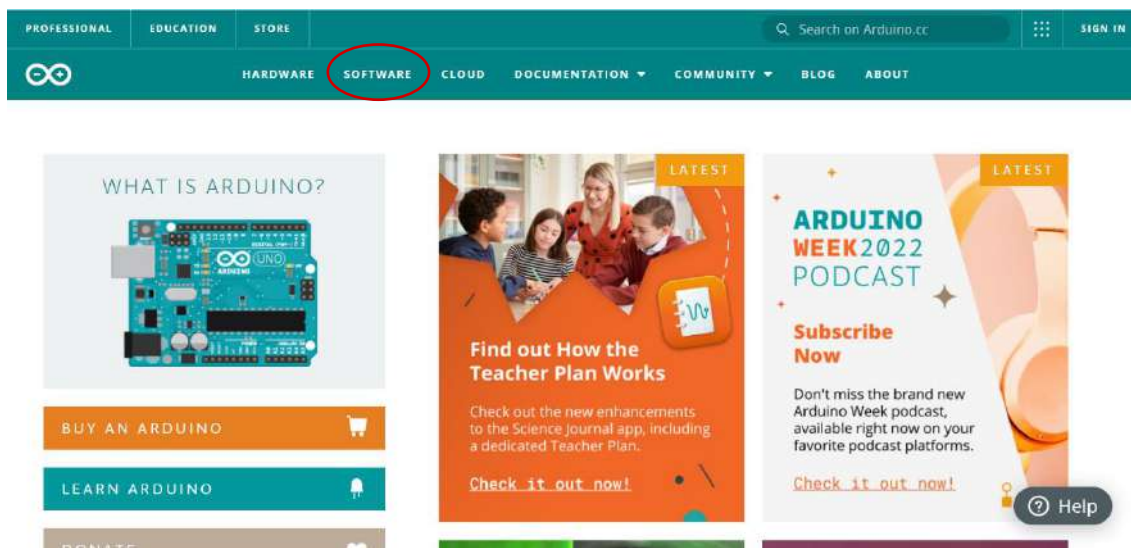
Guia 3: Como baixar e usar a IDE do Arduino

I. Como baixar o IDE do Arduino

Após ter baixado um código da plataforma do Tinkercad, é importante ter o IDE do Arduino instalada no computador para conseguirmos realizar a etapa de prototipagem do circuito. O IDE (em português, Ambiente de desenvolvimento integrado) é o software do Arduino que permite desenvolver, compilar e debugar códigos em um único ambiente.

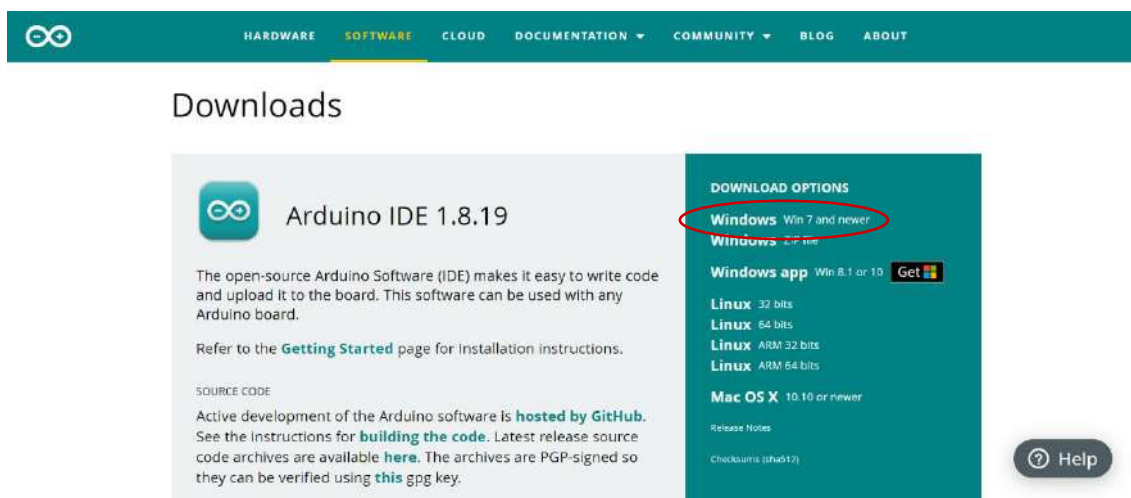
Para fazer o download do IDE, siga os passos a seguir:

Acesse o site oficial do Arduino <http://arduino.cc> e clique em *Software*, na faixa superior da página.

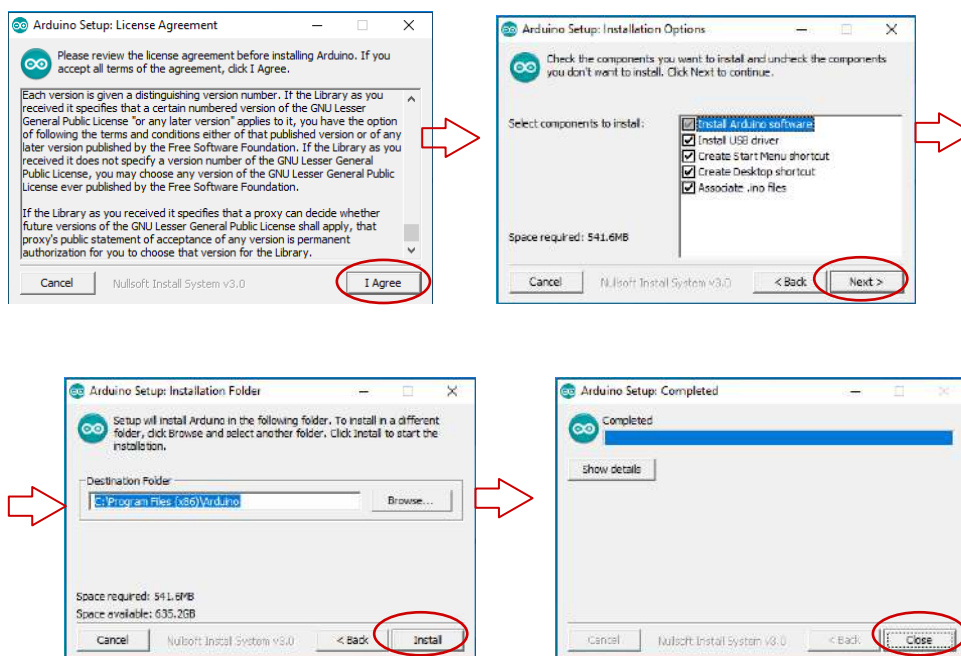


O site disponibiliza a opção de uso online, porém, como seus recursos são limitados, é essencial baixar o IDE. No bloco do *Arduino IDE 1.8.19**, clique na opção de sistema operacional do seu computador (Mac OS, Linux ou Windows). Caso não tenha certeza, consulte as informações nas configurações do seu próprio dispositivo. No caso do Windows, aconselha-se selecionar a 1ª opção *Win 7 and newer* para baixar todos os drivers juntos. Em seguida, você tem a opção de contribuir financeiramente para o Arduino, clicando no botão *Contribute & Download*, ou fazer o download gratuitamente, clicando em *Just download*.

*Pode ser esteja em uma versão mais atualizada do IDE.

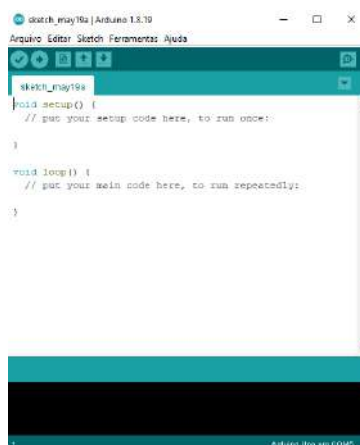


Após o download completo, clique no arquivo para instalar o programa. Aparecem na sequência as telas de aceite dos termos de licença, opções de instalação, pasta para instalação, instalação em si, como mostradas a seguir:



Pronto! IDE do Arduino instalado!

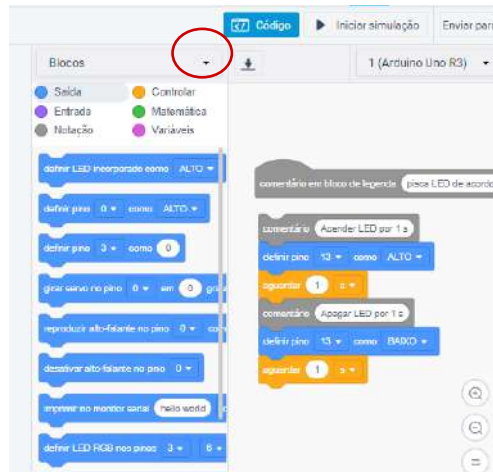
Clique no atalho criado na área de trabalho. Aqui, você pode criar um código na linguagem C++ e explorar os recursos do software.



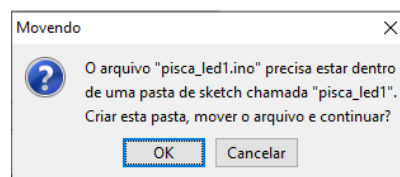
Porém, no nosso caso, em que o código é feito no Tinkercad, iremos apenas utilizar a IDE do Arduino para compilar e carregar o código na placa.

II. Como carregar um código do Tinkercad na IDE do Arduino

Ao finalizar o código no Tinkercad, clique no botão *Código* no ambiente de design, clique no ícone de *Baixar código* (seta para baixo). Vale ressaltar que o Arduino reconhece a linguagem da programação em blocos porque o Tinkercad a correlaciona com a linguagem do Arduino e configura o restante.



Concluído o download, clique no arquivo para que seja aberto. Aparecerá uma janela como mostrado abaixo para que o arquivo seja organizado no local certo. Clique em OK.



Para verificar se o código não possui erros que impossibilitem o carregamento, clique no ícone de *verificar*, indicado por 1 na faixa superior verde escuro da imagem abaixo. Provavelmente, como fazemos simulação e ajustes no Tinkercad, não apresentará mensagens de erro no IDE.



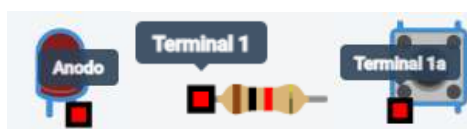
Com a prototipagem do circuito pronta e o cabo USB 2.0 AB conectado à porta USB do Arduino e a uma entrada USB do seu computador, pode-se clicar no ícone de *carregar*, indicado por 2 na imagem acima. Vale ressaltar que, caso o USB não esteja conectado, aparecerá uma mensagem de erro de problema de carregamento para a placa.

Se não tiver erros, aparecerá *Carregado* na faixa inferior e o programa já começará a funcionar!

Para fazer qualquer tipo de alteração no código, no protótipo ou apenas parar o teste, basta retirar uma das extremidades do cabo USB.

Guia 4: Informações importantes, boas práticas no Circuitos Tinkercad e dicionário

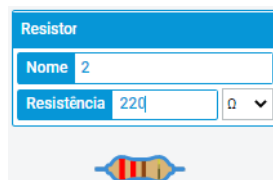
- Ao passar o mouse em cima de cada terminal, pino, etc. no ambiente de montagem virtual no Tinkercad, mostra-se o nome ou a identificação desse item. Exemplos: nomeação dos terminais catodo e anodo de um LED, identificação dos terminais 1 e 2 de um resistor, identificação dos terminais 1a, 2a, 1b, 2b de um botão.



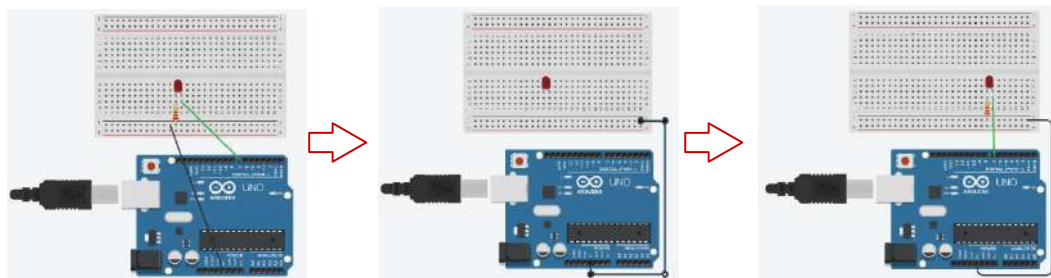
- No caso da protoboard, ao passar o mouse em cima de cada furo, a plataforma mostra se as conexões condutoras são na direção vertical ou horizontal. Assim, torna-se mais fácil confirmar se a disposição dos componentes estão se conectando adequadamente na protoboard.



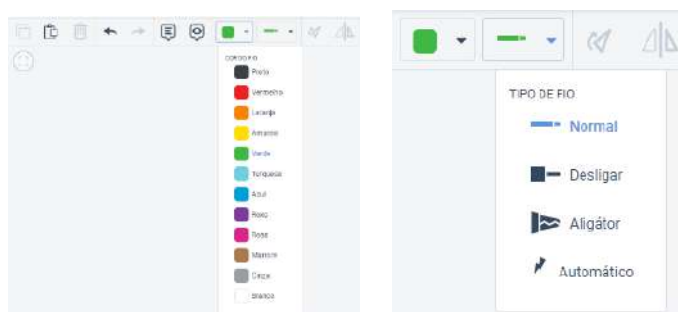
- O resistor não tem diferenciação entre seus terminais, então independe de qual terminal será conectado a cada item, desde que as conexões estejam adequadas.
- Ao clicar em um componente no ambiente de simulação, exceto nos terminais ou pinos, é exibido uma caixa com seu nome, valor e ordem de grandeza, entre outros. É possível alterar esse(s) dado(s) de acordo com seu projeto.



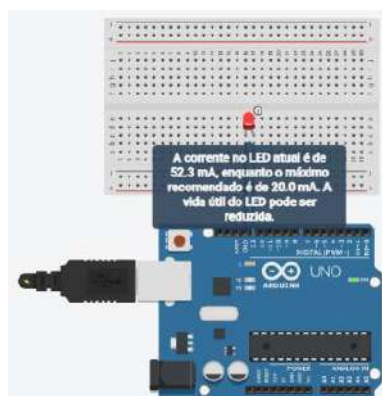
- Ao clicar em um terminal ou pino, cria-se um jumper/fio, que deve ser conectado a um outro terminal ou pino. Para organizar melhor o circuito, pode-se adicionar pontos no jumper clicando 2 vezes sobre o fio. Para excluir pontos desnecessários, basta clicar no ponto e clicar no ícone da lixeira na barra do canto superior à esquerda. Como boa prática, procura-se ajustar, se possível, a disposição dos componentes para evitar cruzar jumpers e passar jumpers sobre componentes. No caso abaixo, foram adicionados pontos no jumper preto para que contornasse o Arduino e os demais componentes foram alinhados com o outro terminal do LED, já que havia espaço. Outra alternativa poderia ser alterar o pino, mas se deve lembrar de modificar no código também.



- Geralmente, tanto no ambiente virtual como físico, optamos por utilizar fios pretos para GND e fios vermelhos para fonte de alimentação. Para os demais componentes, é sugerido fazer associações às cores dos componentes, quando for possível. No Tinkercad, você pode alterar a cor do jumper na barra do canto superior, escolhendo a cor desejada. Você também pode alterar o tipo de jumper, mas na maioria das vezes utilizaremos o normal, como já é pré-definido.



- Caso tenha uma montagem que possa danificar algum componente (por exemplo, um circuito de LED sem um resistor), o Tinkercad sinaliza esse ponto de atenção durante a simulação. Assim, é importante ajustar esses pontos e ficar atento para evitar esses problemas na simulação física do circuito.



- Para os casos de necessidade de ajuste para o funcionamento adequado do circuito, faz-se necessário investigar mais a fundo. Possíveis erros:

- Na montagem virtual:
 - Verificar se o circuito fecha e se todos os terminais estão conectados;
 - Se tiver LED, verificar se o catodo e o anodo estão conectados corretamente;
- Na programação:
 - Verificar se o(s) pinos(s) no código correspondem ao(s) pino(s) correto(s) na montagem;
 - Verificar se todos os componentes estão conectados firmemente.

Dicionário do curso

- **Arduino:** plataforma eletrônica open-source baseada em hardware e software fáceis de usar. Amplamente difundida no mundo todo por ser open-source, acessível e multifuncional. Possui diversas aplicações em diversas áreas, como construir protótipos de robótica, fazer experimentos físicos, construir materiais artísticos interativos, entre outros.
- **Compile:** converter um programa que foi feito em linguagem de alto nível para uma linguagem de máquina (baixo nível); isto é, traduzir códigos que o computador não é capaz de entender.
- **Datasheet:** um documento que fornece as especificações de um determinado produto.
- **Debugar:** consertar erros existentes no código de um programa.
- **Default:** amplamente utilizado na informática, indica algo pré definido, padrão.
- **Hardware:** componentes físicos que armazenam e carregam instruções escritas por meio de software.
- **IDE:** Ambiente de desenvolvimento integrado. Ferramenta que possibilita editar o código, debugar, compilar em um único ambiente.
- **Linguagem C++:** linguagem de programação open source orientada a objetos, podendo ser linguagem de alto e baixo nível.

- Open-source: tratam-se de softwares com código aberto, ou seja, qualquer pessoa pode ver, modificar e distribuir códigos. Apoia-se em comunidades que colaboram com o enriquecimento do software.
- Shield: placas que podem ser conectadas ao Arduino e outras placas, adicionando funcionalidades de forma prática e confiável.
- Software: série de instruções, dados e programas utilizados para operar um computador e executar tarefas específicas.
- Tinkercad: plataforma web gratuita e fácil de usar que estimula a criatividade e inovações a partir de criação de projetos relacionados ao 3D, eletrônicos e programação

Aula 1: Primeiro contato

□ Objetivos da aula sugerida

- Criar um ambiente confortável para que os alunos interajam;
- Auxiliar o professor a entender o contexto da turma em relação aos possíveis conhecimentos prévios na área;
- Trazer alguns conceitos básicos de programação de forma lúdica e educativa aos alunos.

□ Recursos da aula

- Dispositivos com acesso à internet;
- Projetor (opcional);
- Papéis e canetas (opcional).

□ Descrição das atividades

- Dinâmica quebra-gelo (opcional)
 - Objetivo: promover uma interação inicial entre professor e alunos, principalmente se não se conhecem.
 - Existem várias dinâmicas de quebra-gelo, mas uma sugestão seria todos formarem em círculo e cada um se apresentar e trazer uma curiosidade ou fazer uma pergunta e passar a vez para outra pessoa, jogando um objeto para ela.
- Avaliação de Diagnóstico de nível conhecimento lógica de programação e eletrônica básica, além de expectativas dos alunos
 - Objetivo: trazer um recurso para o professor ter uma noção geral do conhecimento dos alunos em relação a temas gerais do curso e auxiliar no direcionamento das aulas. Vale ressaltar que a ideia não é questionar se o aluno sabe descrever termos técnicos. Além disso, entender ou mesmo provocar a reflexão dos alunos sobre o que esperam do curso.
 - Duração: 30 min
 - Formato: formulário online. A sugestão de estrutura do diagnóstico inicial é mostrada no final desta aula.

- Breve apresentação do curso:
 - Objetivo: instigar de forma rápida as inúmeras possibilidades desses recursos. Pode-se mostrar projetos mais complexos para atrair e inspirar os alunos.
 - Duração: 10 min.
 - Formato: projetor ou plataforma já utilizada pela escola (ex. google classroom).
 - Mostrar alguns projetos feitos no tinkercad e no arduino. Não é necessário explicar o que é cada termo, já que haverá uma aula para cada recurso.Exemplos:
 - Arduino: <https://www.youtube.com/watch?v=2sAT4x944pI>
 - Tinkercad - pomodoro simples:
<https://www.tinkercad.com/things/3azcokBqZqH>

- Dinâmica que mostre um processo importante em programação e/ou robótica.
 - Sugestão 1: Descrever uma atividade simples cotidiana, por exemplo, como se prepara um pão com manteiga; como escova os dentes; como faz um embrulho de presente, entrar na sala e sentar etc. O ideal é que seja algo que dê para realizar dentro da sala de aula.
 - Objetivo: mostrar para os alunos a importância de descrever as ações de uma tarefa de forma sequencial e específica.
 - Duração: 20 min.
 - Sugestão do formato: o professor deseja completar uma tarefa e pede o auxílio dos alunos para que eles deem ordens do que precisa ser feito até que essa tarefa seja concluída. Um voluntário é chamado para começar (escrevendo a sequência e depois falando) e o professor segue essas ações, mostrando possíveis interpretações diferentes. Caso o professor não chegue ao resultado final esperado, passa a tarefa para um próximo aluno. Isso mostra que, da mesma forma na dinâmica, o computador não saberá daquilo que não estiver na sequência e não for bem especificado.
 - Referência: <https://www.youtube.com/watch?v=Ct-1OOUqmyY&feature=youtu.be>
 - Sugestão 2: Descreva o que você está vendo para que o outro imite/desenhe/etc.
 - Objetivo: idem ao de cima.

- Duração: 20 min.
 - Sugestão de formato: formam-se duplas. Uma pessoa da dupla será responsável por visualizar um desenho, um vídeo, etc., enquanto a outra não pode ter acesso a esse material, porém irá imitar/desenhar aquilo que for passado pelo que visualizou. Depois de alguns minutos de interação, mostrar qual era o material, se for necessário. Mostrar que o imitador/desenhista seria o computador, que recebe um código/dado mas precisa de instruções específicas para saber o que executar.
- Para a próxima aula: se o aluno tiver acesso à internet até a próxima aula, pedir para cada um ver/rever fotos e vídeos que utilizam microcontroladores - como Arduino, Micro:bit e Raspberry) e questionar "O que eles têm em comum? O que permite que tal solução seja mais inteligente, automatizada?" etc. A ideia não é se limitar à existência de microcontroladores apenas, mas de identificar aspectos, componentes comuns nos projetos eletrônicos. Exemplos de materiais:
- [youtube.com/watch?v=Ko8UjSGVzuw](https://www.youtube.com/watch?v=Ko8UjSGVzuw) (1:07)
 - <https://www.youtube.com/watch?v=KA2UfTo1Yr4&list=PL7CjOZ3q8fMdz0AN6lagBDV1OMBz2cFkk&index=5>
 - https://www.youtube.com/watch?v=6a_rykLIN3k

Vamos mergulhar no curso de Robótica Legal?

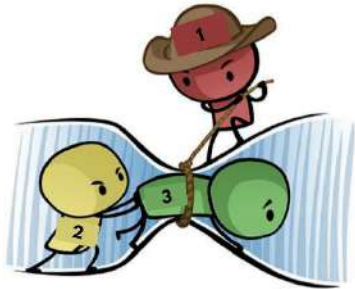
A seguir faremos algumas perguntas relacionadas ao nosso curso. Não tem problema se você não souber as respostas! O objetivo aqui é entender melhor como iremos desenvolver as aulas ao longo do curso :)

| Perguntas | Respostas |
|---|--|
| Você já teve contato com programação e/ou eletrônica anteriormente? | <input type="radio"/> Já tive contato com programação <input type="radio"/> Já tive contato com eletrônica <input type="radio"/> Já tive contato com programação e eletrônica <input type="radio"/> Não, vou começar agora! |

| | |
|---|--|
| <p>Se você respondeu anteriormente que já teve contato, poderia explicar brevemente como foi? Aqui você pode escrever qual(is) plataforma(s) foi(ram) utilizado(s), o que foi abordado (ex. estrutura condicional (se e se não) , loop, acender LED, etc.), o que você aprendeu, o que chamou a sua atenção, sua percepção e interesse nessa área. Seja sincero(a)!</p> | <p>Aberta</p> |
| <p>Você já ouviu falar de Arduino? E Tinkercad?</p> | <p><input type="radio"/> Já ouvi falar sobre Arduino <input type="radio"/> Já ouvi falar sobre Tinkercad <input type="radio"/> Já ouvi falar sobre Arduino e Tinkercad <input type="radio"/> Não, vou conhecer agora!</p> |
| <p>Se você respondeu que já ouviu falar de Arduino e/ou Tinkercad, descreva brevemente o que você sabe. Por exemplo: trata-se de um aplicativo, um site ou uma loja. A sua função é...</p> | <p>Aberta.</p> |
| <p>Sua amiga está indo à padaria a pé e precisa atravessar uma rua que tem semáforo convencional. O que ela deveria fazer nessa situação? Mais de uma alternativa pode estar correta.</p> | <p><input type="checkbox"/> Se o sinal de pedestre estiver verde, ela não pode atravessar a rua. Se não, ela também não pode atravessar a rua. <input type="checkbox"/> Se o sinal de pedestre estiver verde, ela pode atravessar a rua. Se não, ela não pode atravessar a rua. <input type="checkbox"/> Se o sinal de pedestre estiver verde, ela pode atravessar a rua. Se não, ela também pode atravessar a rua. <input type="checkbox"/> Enquanto o sinal de pedestre estiver verde, ela pode atravessar.</p> |

| | |
|--|---|
| | [] Nenhuma das alternativas acima. |
| <p>Seus primos mais novos estão aprendendo a somar números e você teve a ideia de fazer um programa para deixar essa aprendizagem mais divertida! Os itens abaixo são comandos de uma possível estrutura para esse programa, mas estão bagunçados. Você poderia indicar a sequência para que o programa funcione? Dica: não precisa reescrever os comandos. Basta ordenar a partir das letras de A a H.</p> <p>A. Se, na 2ª tentativa, o número digitado não for o resultado correto, mostrar na tela "O resultado dessa soma é z."</p> <p>B. Voltar para a tela "Oi, vamos praticar soma de números!"</p> <p>C. Se não, mostrar na tela "Esse não é o resultado dessa soma. Tente novamente!"</p> <p>D. Mostrar na tela a pergunta de soma que foi exibida anteriormente.</p> <p>E. Mostrar na tela "Oi, vamos praticar soma de números!"</p> <p>F. Se, na 2ª tentativa, o número digitado pela pessoa for o resultado correto, mostrar na tela "Parabéns! Você acertou!"</p> <p>G. Se o número digitado pela pessoa for o resultado correto, mostrar na tela "Parabéns! Você acertou!"</p> <p>H. Mostrar na tela "Quanto é: $x + y$?"</p> <p><i>Fonte: autoral.</i></p> | Aberta (sequência das letras). |
| <p>Agora um pouco sobre elétrica: Qual caminho a corrente elétrica percorre quando ligamos um circuito eletrônico?</p> | <p>() Sai do polo positivo e vai para o negativo</p> <p>() Sai do polo negativo e vai para o positivo</p> <p>() Não sei ainda, mas vou aprender no curso</p> |
| <p>A imagem abaixo ilustra 3 grandezas físicas importantes quando tratamos de eletricidade. Qual seria a correspondência dos números com as grandezas? Fique atento: o que está acontecendo com cada</p> | <p>() 1: Diâmetro; 2: potencial gravitacional; 3: Vazão</p> <p>() 1: Tensão/potencial elétrico; 2: Corrente; 3: resistência</p> |

grandeza na figura é essencial para responder essa pergunta!



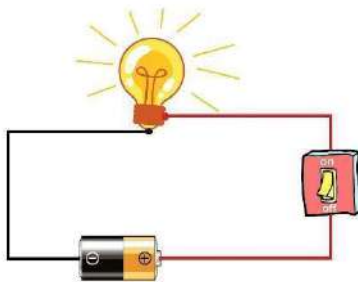
Fonte: Adaptação embarcados.com.br/wp-content/uploads/2016/07/Tensão-corrente-resistencia-destaque.png.webp

() 1: Resistência; 2: Tensão/potencial elétrico ; 3: Corrente

() 1: Vazão; 2: diâmetro; 3: potencial elétrico

() Não sei ainda, mas vou aprender nesse curso!

Você e seus amigos querem construir um projeto de circuito simples para acender uma lâmpada. Primeiro, é importante saber como funciona esse circuito... qual explicação seria a mais adequada para esse caso?



Fonte: <https://pt-static.z-dn.net/files/d21/b6828d8cba027174213da179fa3fc5c7.png>

() São necessários apenas lâmpada, interruptor e fio condutor para realizar esse projeto. A fonte de energia é opcional. Conectamos os componentes com fio condutor e a corrente pode mudar de direção no circuito (do polo positivo para o negativo ou o contrário).

() Para acender uma lâmpada, é essencial ter, além da lâmpada, uma fonte de energia e fio condutor. Podemos incluir um interruptor para conseguirmos acender ou apagá-la. Em relação à corrente elétrica, ela sai do polo positivo e vai para o polo negativo.

() São necessárias duas fontes de energia, uma que tenha polo positivo e outra que tenha polo negativo. Assim, une-se as duas no circuito, permitindo que a corrente saia do positivo e vá para o

| | |
|---|--|
| | <p>negativo. Com isso, a lâmpada acenderá.</p> <p>() Para acender uma lâmpada, vamos precisar de lâmpada, interruptor, fio condutor e bateria. Conectamos a bateria (fonte de energia), o interruptor e a lâmpada a partir dos fios condutores e, assim, a corrente sai do polo negativo para o polo positivo e acende a lâmpada.</p> <p>() Não sei, mas vou aprender nesse curso!</p> |
| <p>É a última! O que você espera desse curso? Fique à vontade para escrever suas expectativas! Pode ser em relação à aprendizagem, à dinâmica das aulas, etc. Além disso, se sentir confortável, fale com seus colegas e seu professor do que está achando no curso durante as aulas. Seria ótimo ter esse retorno de você!</p> | <p>Aberto.</p> |

Avaliação da aula

- Os alunos conseguiram fazer o diagnóstico no tempo previsto?
- Os alunos entenderam a importância da sequência e especificações de programação pela dinâmica? Participaram com perguntas e/ou respostas?

Aula 2: Conhecendo a placa

□ Objetivos da aula sugerida

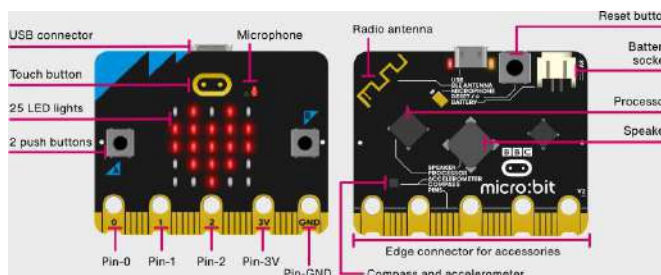
- Introduzir as placas aos alunos, dando enfoque ao Arduino, mostrando aplicações, explicações gerais e suas partes.
- Servir como momento exploratório para despertar a curiosidade e tirar possível receio da complexidade do curso.

□ Recursos da aula

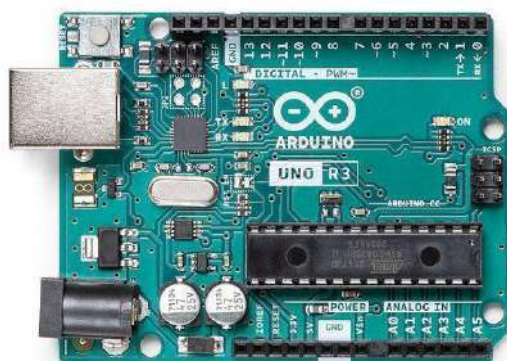
- Computador e projetor
- Arduino(s)
- Papel, lápis, caneta (opcional)

□ Descrição das atividades

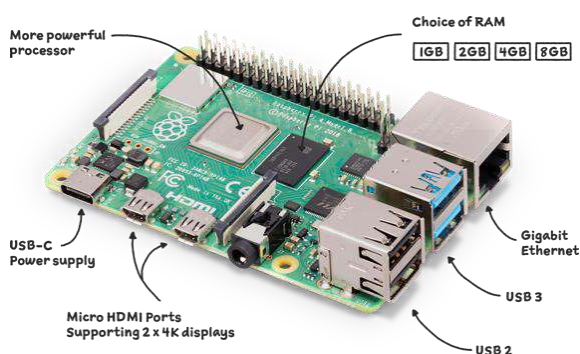
- Introdução com exemplos práticos
 - Objetivo: a partir da atividade final da aula anterior, reforçar aos alunos que as placas eletrônicas estão no cotidiano e discutir algumas das possíveis aplicações.
 - Duração: 10 min.
 - Repassar alguns trechos dos vídeos e fomentar a discussão:
 - O que eles têm em comum? O que comanda esses projetos eletrônicos?
 - A partir desses exemplos, onde mais podemos encontrar essas placas?
- Introdução às placas eletrônicas



Fonte: <https://microbit.org/pt-pt/get-started/user-guide/overview/>



Fonte: <https://store.arduino.cc/products/arduino-uno-rev3>



Fonte: raspberrypi.com/products/raspberry-pi-4-model-b/

- Objetivo: explicar o que são essas placas e para que servem; apresentar as principais placas com uso educacional e suas vantagens e desvantagens; posteriormente, aprofundar nas funcionalidades do Arduino, que será a placa utilizada nesse curso.
- Obs: quando falamos “placas”, estamos nos referindo ao material físico do Arduino, micro:bit e Raspberry Pi. Além dessa parte física (o hardware), cada um deles apresentam a parte de programação (o software), que possuem uma interface própria com linguagem de programação adaptada à placa.
- Duração: 15 min.
- Citar de forma sucinta o que elas têm em comum, os diferenciais de cada uma, quais os principais tipos de aplicação. A sugestão é trazer as definições diretamente da fonte, de forma resumida e simplificada, destrinchando os termos mais complexos.
 - Micro:bit: é um computador de bolso que pode ser programado com editor em blocos ou em texto. Voltado para ensino de crianças e pré-adolescentes. Para funcionar, é necessário, além da Micro:bit, um cabo micro USB e um computador/celular para escrever o programa. Pode ser

utilizado para jogos (com ou sem interação entre mais de uma micro:bit), projetos com medição de temperatura, luminosidade, aceleração, etc.

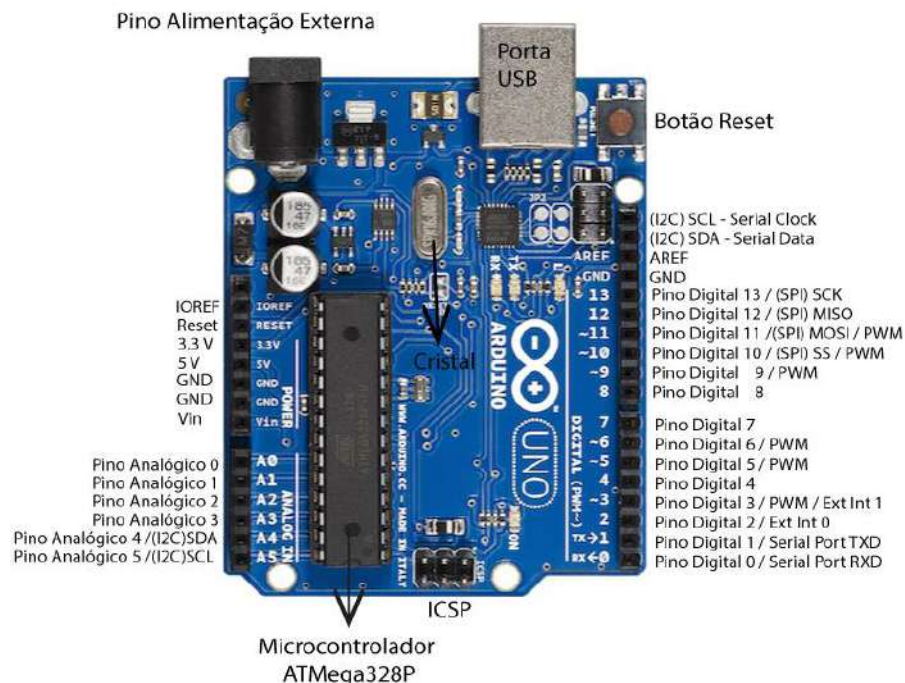
<https://microbit.org/pt-pt/>

- Raspberry PI: é um minicomputador de placa única que, quando conectado a um monitor, um mouse e um teclado, torna-se um computador desktop! Diferente do Arduino, ele pode desempenhar várias atividades, como navegar na internet, fazer vídeos chamadas, controlar uma impressora 3D, etc. Por conta de todas essas aplicações, o valor da placa é a mais elevada. Recomendável para todas as idades, para várias áreas de aplicação, mas requer um conhecimento mais aprofundado de programação. Pode ser programado em Python. raspberrypi.com https://www.filipeflop.com/blog/o-que-e-raspberry-pi/https://www.if.ufrj.br/~gpenello/DisciplinasAntigas/Lab2_2018-1/Guia_Raspberry_Pi_para_Iniciantes_-_FilipeFlop.pdf
- Arduino: não é um mini computador, que desempenha várias funções. *Arduino é uma plataforma eletrônica open-source baseada em hardware e software fáceis de usar. As placas Arduino são capazes de ler entradas - luz acesa de um sensor, um botão pressionado por um dedo, uma mensagem no Twitter - e torná-las em saídas - acionando um motor, acendendo um LED, publicando algo online. Você pode falar para sua placa o que fazer enviando uma série de instruções para o microcontrolador da placa. Para fazer isso, você usa a linguagem de programação do Arduino (baseada em ligações) e no Software Arduino IDE - Ambiente de Desenvolvimento Integrado em pt (baseado em Processamento).*
(...) Professores e estudantes usam [o Arduino] para construir instrumentos científicos de baixo custo, para provar princípios químicos e físicos ou para começar na programação e robótica. Designers e arquitetos constroem protótipos interativos, músicos e artistas usam para instalações e para experimentar novos instrumentos musicais. Makers ("fazedores"), claro, usam para construir vários projetos exibidos na Feira Maker, por exemplo. Arduino é uma ferramenta chave para aprender novas coisas. Qualquer um - crianças, hobbystas,

artistas, programadores - podem começar a mexer seguindo apenas as instruções passo a passo de um kit, ou compartilhando ideias online com outros membros da comunidade do Arduino.", site do Arduino.

- **Destrinchando o Arduino**

- **Objetivo:** trazer de forma interativa quais são as principais partes do Arduino, hardware e software. Será retomado nas aulas posteriores, mas é importante que o aluno tenha noções básicas.
- **Duração:** 25 min:
- Se possível, distribuir um arduino por grupos de 3-5 pessoas para que os alunos possam manusear durante a explicação. Pode-se utilizar um papel com o Arduino impresso ou pedir para que os alunos façam um esboço. Posteriormente, poderia ser utilizado como material de apoio nas próximas aulas.
- **Parte I - Explorar o hardware:** identificar as principais partes do Arduino e mostrar suas funções.



Fonte: blog.smartkits.com.br/arduino-uno/

- **GND (Ground):** terra. Pino de referência de tensão (0V), possibilitando medir tensão de um ponto qualquer. Além disso, tem a função de fechar os circuitos.

- 5V: fonte de alimentação. Fornece tensão para alimentar placas que podem ser adicionadas para expandir as funcionalidades do Arduino (shields), microcontrolador do Arduino e componentes externos.
 - A0, A1, A2, A3, A4, A5 (pinos analógicos): são pinos de entradas analógicas.
 - 0,1,2,3,4,5,6,7,8,9,10,11,12,13 (pinos digitais): são pinos de entradas ou saídas digitais, sendo que os pinos com marcação ~ podem ser programados como saída analógica a partir de um recurso (modulação PWM).
 - LED embutido: está conectado com o pino 13. Dependendo do estado dele (HIGH ou LOW), o LED acende ou apaga.
 - Botão vermelho reset: reiniciar/reconfigurar. Botão que reinicializa a placa de forma externa (sem ser implementada no código), retomando o código do início.
 - Porta USB: pode ser utilizado para alimentar a placa (geralmente vamos utilizar dessa forma), transmite o código para placa e permite a comunicação serial entre placa e o computador utilizado. Fica gravado apenas o último código.
 - Pino alimentação externa: pode ser utilizado para alimentar a placa, caso não esteja usando a porta USB conectada ao computador. Pode-se inserir uma bateria ou uma fonte. Recomenda-se que a tensão seja entre 9-12V.
 - Parte II - Apresentar o software: instigar os alunos a descobrirem/identificarem como essas placas leem as informações e executam as ações. Uma sugestão seria construir um circuito simples e pisca LED, sem entrar nos detalhes da construção, e questionar: o Arduino “saberá” o que fazer? O que está faltando para que realize a ação desejada? Relembrar da dinâmica feita na aula anterior, associando o(s) aluno(s) que deram as direções ao código/programa/ algoritmo e o professor à placa e componentes.
- Após essa discussão, mostrar a interface da IDE do Arduino. O ideal é que o software tenha sido instalado previamente. Explicar que a linguagem utilizada no Arduino é C++ com adaptações. Se for possível, mostrar um código bem simples para que, nessa etapa,

os alunos visualizem o código a título de curiosidade. No próprio IDE, você pode utilizar exemplos de código, como o pisca LED embutido no Arduino. Para abri-lo no IDE, é preciso clicar sequencialmente em *Arquivos > Exemplos > 01. Basics > Blink*, conectar ao Arduino e carregar o código para que o LED comece a piscar. Reforçar que as aulas serão baseadas na linguagem de programação em blocos*, na qual se desenvolve a lógica e o pensamento computacional, mas é considerada mais simples e prática. Destacar para os alunos que o curso não é voltado para desenvolver futuros programadores e cientistas da computação; o propósito é despertar o pensamento crítico em projetos, aprender sobre lógica de forma simples e acessível, além de resolver problemas e trabalhar em grupo.

*No tinkercad, tem a possibilidade de construir só em blocos, só em texto ou blocos + texto. Por curiosidade, é válido mostrar a equivalência do código em blocos para texto, porém nosso foco é a programação em blocos.

- Para a próxima aula: se os alunos tiverem acesso à internet antes da próxima aula, pedir para que eles pesquisem e tragam um projeto (foto, vídeo, site, etc.) feito na plataforma do Tinkercad que lhe chamou a atenção.

□ **Avaliação da aula**

- Os alunos interagiram e conseguiram construir o material de apoio?
- Os alunos entenderam a importância dos comandos, da programação para que um projeto funcione?

Aula 3: Conhecendo a plataforma (parte 1)

□ **Objetivos da aula sugerida**

- Introduzir a plataforma do Tinkercad, discutindo sobre exemplos de projetos e explorando a interface do aluno.

□ Recursos para a aula

- Dispositivos com acesso à internet
- Projetor (opcional)
- Papéis e lápis (opcionais)

□ Descrição das atividades

- Momento de compartilhar os exemplos de projetos que chamaram a atenção dos alunos.
 - Objetivo: introduzir de forma dinâmica o que o Tinkercad possibilita, identificando quais são os principais recursos e vantagens da plataforma.
 - Duração: 20 min.
 - A sugestão é dividir a sala em grupos médios - 5, 6 pessoas - para que cada um comente o projeto que encontrou e por que lhe chamou a atenção. Posteriormente, pede-se para que cada grupo comente de 2 projetos que acharam mais interessantes para o restante da turma. Cabe ao professor questionar sobre as possibilidades que o Tinkercad traz a partir dos projetos mencionados, do nível mais básico até mais avançado, e incentivar a interação entre a sala.

- Mão na massa! Primeiros passos: entrar na plataforma e explorar o espaço.
 - Objetivo: introduzir o espaço do aluno do Tinkercad.
 - Duração: 40 min.
 - A partir do Guia 1, mostrar como o aluno vai entrar na plataforma com seu usuário e como ele pode criar um novo projeto. Além disso, abordar sobre alguns recursos interessantes do tinkercad (sala de aula, comunidade, exemplos de projeto). O ideal é que o professor já tenha se registrado e criado uma sala antes da aula (ver Guia 2) para que os alunos já entrem nessa sala.

- Deixar livre alguns minutos para que cada um explore a tela de criação de projeto para ter uma noção dos componentes e dos blocos. Trazer a ideia de *tinker*, de fazer pequenas mudanças para melhorar ou reparar algo; ou seja, criando e ajustando ideias.

- Se for possível, mostrar um projeto simples já pronto (exemplo: pisca 2 LEDs), destacando que foram feitos: a montagem virtual, a programação em blocos + texto e a simulação (preferencialmente com a visualização de mensagens no monitor serial).

☐ **Avaliação da aula**

- Os alunos conseguiram explicar o exemplo trazido para o grupo? Conseguiram transmitir para a sala posteriormente?
- Foi possível que todos entrassem no seu ambiente e explorassem a interface?

Aula 4: Conhecendo a plataforma (parte 2)

☐ **Objetivos da aula sugerida**

- Retomar as funcionalidades do Tinkercad Circuitos introduzidas na aula anterior
- Apresentar e discutir sobre os tipos de blocos e possíveis aplicações

☐ **Recursos para a aula**

- Dispositivos com acesso à internet
- Projetor (opcional)
- Papéis e lápis (opcionais)

☐ **Descrição das atividades**

- **Destrinchando os blocos**
 - **Objetivo:** retomar e explicar cada grupo e formatos de blocos, para que os alunos tenham um conhecimento básico de programação em blocos e da interface do Tinkercad antes de programar os primeiros códigos. Fazer alguns exercícios para fixar os conteúdos explorados.
 - **Grupos de blocos:** explicar que os blocos são divididos entre grupos, indicados por cores. Sugere-se trazer analogias com exemplos mais próximos do nosso cotidiano.
 - **Duração:** 20 min.
- **Entrada e saída:**

Os aparelhos eletrônicos e máquinas no geral têm como base esses dois termos. Em um sistema, algo entra, sofre uma transformação e sai um resultado. Exemplos: quando vamos mandar uma mensagem para alguém pelo celular, a entrada seria a sua digitação no teclado do celular e a saída seria o aparecimento dos caracteres digitados; quando vamos dar um passo à frente, a entrada seria vontade, necessidade de se movimentar e a saída seria a resposta do cérebro para executar a ação dos membros inferiores.

Saída

No tinkercad, os blocos azuis (saída) são aqueles que possibilitam determinado estado ou ação a partir de uma especificação anterior. Por exemplo, um LED que acende e apaga definido por comandos com determinado intervalo de tempo.

Entrada

Os blocos roxos (entrada) são aqueles que permitem a leitura de determinada situação, que servirá para algum estado ou ação posterior. Por exemplo, quando alguém aperta o botão no circuito (entrada), o Arduino lê essa informação e faz alguma ação, como, por exemplo, acender um LED.

● **Notação:**

Notação

Uma sugestão seria comparar um mesmo código, preferencialmente construído no Tinkercad, com e outro sem comentários. Ressaltar que é uma prática muito difundida entre pessoas que programam para que elas relembrem a lógica do programa ou outras pessoas compreendam melhor a lógica do programa. Apontar que esses comentários não interferem na compilação do código.

● **Controlar:**

Controlar

Esses blocos têm a função de indicar a recorrência de ocorrer um evento (bloco de ação/estado ou conjunto de blocos com um determinado objetivo), com ou sem uma condição específica. Alguns exemplos: entre a ação de acender e apagar o LED, esperar 1 segundo e repetir esse evento 5 vezes.

Um exemplo do cotidiano seria programar um despertador, que toca até 10 vezes (repetição) a cada 5 min (intervalo) de segunda a sexta (condição).

- **Matemática:**

- Matemática

Esses blocos associam-se a outros blocos e estabelecem várias operações de matemática (-, +, /, x, %, ^, <, ≤, >, ≥, =, ≠) e lógica (e/ou). Além disso, também possibilitam números aleatórios, restrição de intervalos, números absolutos, etc.

Um exemplo é sortear um prêmio em um site de sorteador entre os números 1 e 100.

- **Variáveis:**

- Variáveis

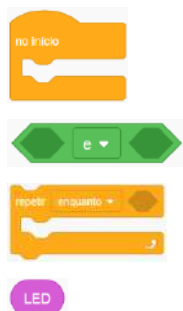
Esse conceito é amplamente utilizado em programação. É utilizado quando queremos definir algo que não está pré-definido na plataforma com a finalidade de armazenar dados, conteúdos que podem mudar. Podemos usar a variável encaixando em outros blocos, definindo ou alterando seu conteúdo. Elas funcionam como se fossem gavetas de um armário, em que podem armazenar esses dados e esses por sua vez podem ser lidos e manipulados. Vamos explorar as variáveis com maior profundidade nas próximas atividades.

Um exemplo de uso de variável seria a pontuação de jogos. Como a pontuação varia constantemente, não é interessante definir todas as possíveis pontuações; por isso pode-se programar de uma forma que, a cada vitória de uma partida, a variável de pontuação acrescenta um.

- Formatos dos blocos: destacar que cada bloco tem seu próprio formato. Isso se deve à necessidade ou não de estar dentro de um outro bloco e da possibilidade de se encaixar a outro bloco, criando uma sequência de blocos para executar o código. Estes formatos têm a função de facilitar a programação, direcionando as possibilidades de encaixe aos usuários.

- Duração: 10 min.
- Formatos existentes:





- Possíveis perguntas para os alunos: Pode-se colocar o bloco de entrada (roxo) “ler pino digital 0” diretamente no bloco de controle (laranja) “repetir enquanto ___”? Por que? Pode encaixar um bloco de uma cor em outro com cor diferente? Por que? Pode-se encaixar o bloco da variável criada “LED” logo embaixo do bloco de saída (azul) “definir pino 0 como ALTO”? Por que? Pode-se colocar a variável “LED” no lugar do 1 no bloco de matemática (verde)?
 - Nesse momento, é importante trazer essas reflexões mais visuais e discutir o que tem uma explicação por trás desses formatos, já que foram pensados nas possibilidades da lógica de programação.
- Exercício rápido para praticar o conteúdo aprendido:
 - Duração: 10 min.
 - Se os alunos quiserem...
 - ...que um evento repita 10 vezes, qual grupo e qual bloco utilizariam?
 - ...que tenha uma condição para acontecer algo (ex. se apertar um botão acende uma luz vermelha, se aperta outro botão acende uma azul), qual grupo e qual bloco utilizariam?
 - ...que tenha que medir a temperatura do ambiente, qual grupo e qual bloco utilizariam?
 - ...e que com essa temperatura medida, tenham que criar uma condição (positiva e negativa) se a temperatura for maior que 50°C?
 - ...que tenham que reproduzir um som por 1 s?
 - ...comentar que “o LED ficará aceso se o sistema estiver funcionando”?
 - Exercício de resolução de problema:
 - Duração: 20 min.
 - Individual ou dupla.

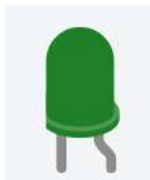
- Q1: Ana e Daniel estão fazendo um código para que uma luz acenda a partir do pino 0 do Arduino quando alguém aperta um botão associado ao pino 2. A partir do que foi discutido nesta aula, você saberia dizer se esse código funcionaria como eles pensaram? Por que? Por enquanto, não precisa dar a sugestão para resolução; nas próximas aulas, você vai aprender como fazer!



- Q2: Ana e Daniel estão fazendo um outro código. Dessa vez, eles fizeram a medição da temperatura corretamente, mas estão com um problema na parte seguinte: gostariam que aparecesse na tela do monitor serial “Está quente! ”, quando a temperatura estivesse maior que 30°C. Você saberia dizer qual é o problema e por que não está funcionando?



- Para a próxima aula: o que é esse componente? Para que serve? O que conseguimos saber sobre esse componente a partir do tinkercad?



□ Avaliação da aula

- Os alunos responderam ao exercício de achar os blocos? Conseguiram achar os blocos adequados para cada caso?
- Os alunos conseguiram entender e responder às questões de resolução de problema durante a aula? Tiveram dificuldades?

Aula 5: Introdução à eletrônica básica e pisca LED

□ Objetivos da aula sugerida

- Introduzir conceitos de elétrica e eletrônica;
- Iniciar a montagem do circuito e lógica por meio de código simples.

□ Recursos para a aula

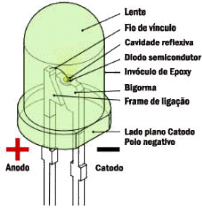
- Dispositivos com acesso à internet
- Projetor (opcional)
- Arduino Uno
- Cabo USB 2.0 A B
- LED
- Jumpers
- Protoboard
- Resistor 220 ohms

□ Descrição das atividades

- Explorando os componentes eletrônicos:
 - Duração: 15 min.
 - Questionar a turma: a partir do que foi explorado até agora, quais seriam os componentes necessários para fazer um projeto de pisca LED? Qual é a função de cada um no circuito?
 - Retomar a pergunta feita na atividade final da aula anterior. Mostrar que na aba de Componentes, eles encontram uma breve descrição de cada componente ao alterar a vista em lista, clicando no botão ao lado da lista suspensa dos grupos de componentes, como mostrado no Guia 1.

| Componente | Função | Observações |
|------------|---|-------------|
| Computador | Desenvolver o código, servir como fonte de | |

| | | |
|----------------------------|---|---|
| | <p>alimentação (USB) e, em códigos que usam o monitor serial, mostrar os dados</p> | |
| Bateria | <p>Outra fonte de alimentação que permite maior mobilidade no projeto</p> | |
| Arduino | <p>Armazenar código, servir como central dos comandos, "cérebro"</p> | |
| Protoboard | <p>Expandir possibilidades de conexão sem necessidade de soldar os fios e componentes</p> | <p>Mostrar como estão conectadas as linhas e colunas. Destacar que no tinkercad, ao passar o mouse sobre a protoboard, torna-se visível essas conexões.</p> |
| LED (diodo emissor de luz) | <p>Variada. Serve como uma pequena lâmpada, convertendo energia elétrica em luminosa. É usada amplamente como indicador de estado (se algo está ligado ou não, próximo ou não).</p> | <p>Explicar que um diodo permite a passagem de corrente por um sentido só. Mostrar que existem os terminais anodo (+) e catodo (-), como identificar cada um e a importância de verificá-los no circuito.</p> |

| | | |
|------------------|---|--|
| | | <p>Estrutura do LED</p>  <p>Fonte: http://www.km4.com.br/eficiencia.php</p> |
| Resistor 220 ohm | <p>Regular tensão ou corrente do circuito para elementos que possuem especificações diferentes. Por exemplo, LED trabalha com máximo de 2V e Arduino 5V, por isso coloca-se um resistor nesse circuito para não queimar o LED. São muito utilizados como sensores também.</p> | <p>Curiosidade: mostrar datasheet, importância</p> <p>Mostrar a conta $U=R \cdot i$ e como chegamos no 150 ohm \rightarrow 220 ohm (comercial) 5V-2V/20mA.</p> <p>Retomar o desenho de tensão-resistência-corrente visto no diagnóstico inicial para reforçar as funções de cada grandeza.</p> <p>Explicar que as cores mudam de acordo com o valor. Mostrar a tabela de cores para entender</p> |

| | | |
|--------------|----------|--|
| | | como funciona o código de cores. |
| Jumpers/fios | Conexão. | Mostrar os tipos de jumpers (macho-fêmea). |

- Construindo um circuito de pisca LED
 - Duração: 30 min.
 - Objetivo: fazer um circuito em que um LED acende e apague a cada 1 segundo. Além disso, mostrar que se trata de um projeto que ajudará a fixar os conceitos apresentados até agora na prática e servirá como base para projetos futuros mais complexos.
 - A sugestão de dinâmica da atividade é que o professor, nessas aulas iniciais, explique e mostre os passos na prática ao mesmo tempo que os alunos também fazem. Nas partes 1 e 2, que envolvem apenas o ambiente virtual, os alunos podem seguir individualmente se houver o número suficiente de dispositivos com acesso à internet. Já na parte 3 de montagem física, é válido desenvolver a prática em grupos em virtude da quantidade de componentes disponíveis e do fomento à discussão em grupo relacionada à construção do circuito real.
 - Consulte o Guia 4 para saber mais sobre os recursos na área de design dos circuitos.
 - Parte 1: Montagem virtual do circuito
 - Criar um novo design dentro da atividade na sala de aula e renomear o projeto;
 - Buscar na aba de componentes e incluir no espaço de simulação 1 arduino, 1 protoboard, um LED e um resistor. Ajustar o resistor para 220 ohms. O LED pode ser da cor desejada;
 - Colocar o LED e o resistor nos pinos na protoboard de forma que: o resistor esteja conectado a um dos terminais do LED, o anodo do LED esteja conectado a um pino digital do Arduino e o catodo a um GND do Arduino.

- Reforçar com os alunos o funcionamento desse circuito (sentido da corrente, funções de cada componente, etc.)
- Parte 2: Programação:
 - Clicar em *Código* na barra do canto superior direito;
 - Como acender um LED pode ser determinado como ALTO e apagá-lo como BAIXO em um sistema digital, utilizaremos o bloco de saída *definir pino x como ALTO/BAIXO*. Para acender, selecione o número do pino que você conectou ao anodo do LED e deixe como ALTO. Para apagar, selecione o mesmo pino e deixe como BAIXO.
 - Para que a mudança de estado do LED fique perceptível, é necessário colocar uma pausa entre os comandos. Por isso, adicione o bloco de controlar *aguardar x s/ms* após cada estado do LED.
 - Pronto! Agora é a hora de simular o circuito. Clique em *Iniciar a simulação* na barra do canto superior direito e, se tiver tudo correto, o LED acenderá e apagará de acordo com o intervalo definido. Não se esqueça de parar a simulação pelo mesmo botão que iniciou a simulação, o qual estará escrito *parar simulação*. Fique à vontade para fazer outras versões desse projeto, alterando cada espera, por exemplo.
 - Faça o download do programa clicando no ícone (similar a ↓) de *baixar código* localizado na barra superior no espaço de criar o código.
- Parte 3: Montagem física/protótipo do circuito:
 - Seguir a montagem virtual. Caso não tenha o(s) mesmo(s) componentes, verificar se esses itens estão dentro das especificações para atender à tensão e corrente máxima. Geralmente, para os LEDs difusos 3 e 5mm, podem-se utilizar resistores de 150 ou 220 ohms, considerando a corrente máxima de 20 mA e a tensão máxima no LED de aproximadamente 2V.
 - Para enviar o código ao Arduino, é necessário antes carregar o código no IDE do Arduino. Caso necessário, reveja o Guia 3. Ao tentar abrir o arquivo baixado, aparecerá uma notificação para alterar seu local, já que o a IDE possui um local específico para armazenar os programas. Clique em ok e prossiga para compilar o código. Caso não tenham nenhum erro, conecte o cabo USB a uma entrada do computador e a outra no Arduino. Carregue o código e pronto! Se tudo estiver

correto, o LED começará a funcionar da forma que foi programado. Se precisar alterar algo, retire um dos terminais do cabo USB para ajustar.

- Foi criada uma sala de aula para servir como apoio ao professor dos projetos sugeridos nas aulas. Para entrar, acesse o link para ter a visão do aluno <https://www.tinkercad.com/joinclass/U1AY9S3732FC> e utilize o apelido *teste4960*. Essa atividade está nomeada como *Projeto 1: pisca LED*

- **Desafio: pisca LED 2.0!**
 - Duração: 15 min.
 - Objetivo: incentivar os alunos a criarem um projeto um pouco diferente do primeiro, na qual é incluído mais um LED no circuito com possíveis combinações de pisca e intervalo de tempo.
 - Nessa atividade, é importante que os alunos tentem fazer sem o passo a passo do professor. Pode ser feito individualmente ou em pequenos grupos.

□ **Avaliação da aula**

- Os alunos entenderam as funções de cada componente utilizado no circuito de pisca LED?
- Os alunos conseguiram acompanhar e entender a construção do circuito de pisca LED?
- Os alunos conseguiram realizar a última atividade? Tiraram possíveis dúvidas?

Aula 6: Introdução à entrada e saída digital

□ **Objetivos da aula sugerida**

- Introduzir os conceitos de entrada e saída digital
- Construir um circuito aplicando esses conceitos

□ **Recursos para a aula**

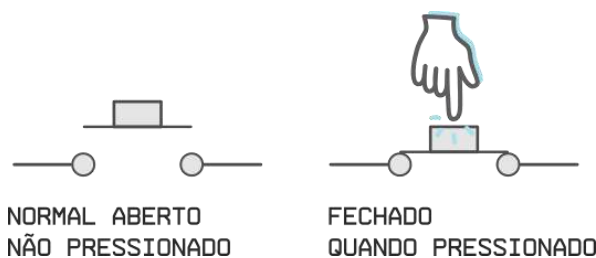
- Dispositivos com acesso à internet
- Projetor (opcional)
- Arduino Uno

- Cabo USB 2.0 A B
- LED
- Jumpers
- Protoboard
- Resistor 220 ohm
- Resistor 10k ohm
- Botão

□ Descrição das atividades

- Pergunta de abertura: pensando no circuito de pisca LED da aula anterior, como podemos acrescentar uma entrada que altere o estado do LED? Quais são os exemplos do cotidiano?
 - Duração: 5 min.
 - Objetivo: instigar os alunos a associarem um botão e um LED com um interruptor e uma lâmpada. Ou seja, uma possibilidade de alterar o estado do LED no nosso circuito da aula anterior é inserindo um botão. Além disso, reforçar os conceitos de entrada (leitura de um estímulo externo) e saída (acionamento, mudança de estado de um componente).

- Explorando um novo componente: botão
 - Duração: 10 min.
 - Objetivo: discutir com a turma sobre o funcionamento de botões
 - Mostrar que o botão é normalmente aberto, ou seja, sem pressioná-lo, não há contato entre seus lados; já ao pressioná-lo, forma-se uma conexão entre os lados. Assim, é possível utilizar esses dois estados (aberto ou fechado) para alterar o estado de um LED (aceso ou apagado).



Fonte: filipeflop.com/aula/kit-maker-arduino-iniciante-interruptor-de-luz/

- Acendendo e apagando LED com botão
 - Duração: 25 min.
 - Objetivo: construir um circuito que, quando o botão é pressionado, o LED acende e quando deixa de ser pressionado, o LED apaga.

- Parte 1: Montagem virtual do circuito
 - Pedir para os alunos construírem a saída digital, similar ao que foi feito na aula anterior. Acompanhá-los na construção para verificar se eles estão com dúvidas.
 - Explicar o passo a passo da inclusão do botão no circuito.
 - Inserir o botão no meio da protoboard, entre as linhas *e* e *f* a fim de ter espaço disponível para incluir componentes associados ao botão. Ressaltar que não faz diferença: conectar ao terminal *a* ou *b* e conectar o GND ou 5V especificamente ao terminal *1* ou *2* do botão. Ou seja, você pode conectar o 5V a qualquer um dos 4 terminais, desde que o GND seja em um dos terminais com número diferente do 5V.
 - Nesses casos em que o nível de entrada é puxado para cima (*pullup*, *HIGH*) ou para baixo (*pulldown*, *LOW*) de acordo com a tensão, geralmente utiliza-se um resistor de 10k ohms.
 - Optando pela lógica de que ao pressionar o botão, será considerado o sinal digital HIGH, o resistor deve estar associado ao GND. Deve-se incluir um jumper para conectar o terminal *a* a uma porta digital. Além disso, no outro terminal, conectar à fonte de 5V. Como tem apenas um jumper vinculado à fonte, pode-se conectar diretamente, sem a necessidade de conectar todos à linha (+) para depois um jumper ser direcionado à fonte de 5V. Já no caso do GND, como tem 2 resistores, torna-se válido encaminhar para a linha (-) e posteriormente conectá-los ao GND do Arduino.

- Parte 2: Programação
 - Assim como na maioria dos códigos, existem várias formas de desenvolvê-los. Aqui, vamos mostrar um exemplo mais simplificado, mas poderia utilizar, por exemplo, uma variável para o botão.

- O primeiro passo seria entender que existe uma condição nesse projeto: se o botão estiver pressionado, deixar LED aceso; se não, deixar LED apagado. Como só há 2 condições, utilizaremos o bloco de controlar *se... outro*, na qual a outra condição de dá por exclusão da primeira;
 - A condição está associada à leitura da porta digital estar no sinal alto. Por isso, inclui-se primeiro o bloco de matemática $... = ...$ no espaço logo após o *se*, adiciona-se o bloco de entrada *ler pino digital x*, sendo x o número do pino relacionado ao botão na simulação e depois pode-se inserir o bloco de matemática *ALTO*.
 - Estabelecida a 1ª parte do condicional, é necessário determinar qual será a ação ou mudança de estado recebido. Nesse caso, seria similar à parte do código da aula anterior: o LED, associado a determinado pino, ficará como ALTO, ou seja, aceso. A espera por x s/ms é opcional.
 - Por fim, na parte do *outro*, adiciona-se o LED como apagado, colocando o pino associado ao LED como BAIXO. A espera por x s/ms também é opcional.
- Parte 3: Montagem física/protótipo do circuito
 - Mesma dinâmica sugerida na aula anterior (aula 5).
 - Desafio: Acendendo e apagando LED com botão 2.0!
 - Duração: 20 min.
 - Objetivo: promover um espaço em que os alunos pensem e discutam lógicas de como fazer uma nova versão do projeto anterior: ao apertar o botão, LED acenda e continue aceso; apenas quando apertar uma outra vez, o LED apaga. Chegar no resultado esperado não é o principal objetivo da atividade.
 - Uma das formas mais simples de desenvolver essa atividade é incluindo mais um botão, na qual um serve apenas para ligar e o outro, para desligar. Caso os alunos estejam com dificuldades em avançar, sugira a adição de mais componentes no circuito. Na parte da programação, deve-se alterar a condição para um dos estados do LED, considerando a leitura do pino associado ao 2º botão.
 - A atividade pode ser desenvolvida individualmente ou em grupo.

- As sugestões de projeto para essa aula encontram-se na atividade *Projeto 2: acender e apagar LED com botão*. Utilizar como apoio apenas ao professor. Link para acessar a sala de aula: <https://www.tinkercad.com/joinclass/U1AY9S3732FC>, utilizando o apelido *teste4960*.

□ Avaliação da aula

- Os alunos relembrou e entenderam o conceito de entrada e saída?
- Os alunos entenderam o funcionamento básico do botão?
- Os alunos conseguiram acompanhar a construção do 1º circuito? Tiveram dificuldades?
- E o 2º circuito? Eles conseguiram refletir e testar ideias para o desafio?

Aula 7: Introdução a entrada e saída analógicas e comparação de sistema digital vs. analógico

□ Objetivos da aula sugerida

- Introduzir conceitos de sistemas analógicos
- Discutir sobre as diferenças entre sistemas digitais e analógicos, ampliando para contextos do cotidiano.

□ Recursos para a aula

- Dispositivos com acesso à internet
- Projetor (opcional)
- Arduino Uno
- Cabo USB 2.0 A B
- LED
- Jumpers
- Protoboard
- Resistor 220 ohms

□ **Descrição das atividades**

- **Aquecimento:** Qual é a diferença entre as imagens do bloco 1 e 2 em relação ao intervalo numérico?

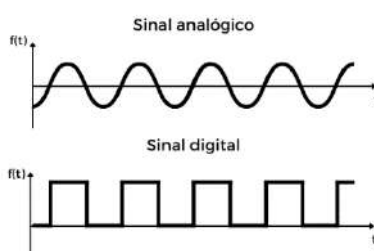


Fontes: <https://pixnio.com/pt/veiculos/carros/painel-de-controle-limite-de-velocidade-velocimetro-odometro-instrumento-carro-dispositivo-controle>
<https://pxhere.com/en/photo/707193>



Fontes: <https://pixabay.com/pt/illustrations/rel%C3%B3gio-digital-n%C3%BAmeros-doze-horas-341253/>
<https://www.lojadomecanico.com.br/produto/146231/3/283/termometro-digital-infravermelho-corporal-32-c---429-c-medical-systel-ly-fl>

- Duração: 10 min.
- Objetivo: mostrar para os alunos exemplos de representações analógicas e digitais do cotidiano e explicar brevemente a característica de cada sinal.



Fonte: autoral.

- Analógico: variam ao longo de um intervalo contínuo de valores
 - Digital: variam por degraus de valores. Em sistemas digitais, haverá apenas 2 degraus: alto e baixo.
-
- Explorando o potenciômetro
 - Duração: 10 min.
 - Objetivo: deixar os alunos explorarem o componente para discutirem tópicos macro levantados até agora.
 - Entregar um potenciômetro para os grupos. Baseado no assunto da aula, questionar como descreveriam um potenciômetro. Analógico ou digital? Usado como entrada ou saída?
 - Perguntas extras:
 - Qual é o intervalo numérico que o potenciômetro trabalha? 0 a 1023.
 - Como funciona? A partir da rotação da haste, que está associada a um cursor no interior do potenciômetro, a resistência varia e, portanto, a tensão no circuito também.
 - Quais são suas aplicações? Calibração de ferramentas fontes de tensão (processo que verifica se os valores de medição de uma ferramenta estão dentro do esperado segundo uma referência); controle de brilho de LEDs, controle de volume de rádios.
-
- Controlando LED RGB pelo potenciômetro
 - Duração: 40 min.
 - Objetivo: construir um circuito que, ao rotacionar o potenciômetro, o LED RGB varie suas cores. Aplicar os conceitos de entrada analógica em um projeto. Incluir o uso do monitor serial para verificação de saídas.
 - Parte 1: Como funcionam os componentes apresentados?
 - O que é um LED RGB?
 - É um LED que une 3 cores de LED em um só: R de red (vermelho), G de green (verde) e B de blue (azul). A partir delas, é possível formar outras cores, como magenta, amarelo, ciano e branco com intensidade variada.

- Possui 4 pernas, sendo 3 para cada cor principal e mais uma que será ligada ao GND ou 5V do Arduino. Para isso, deve-se verificar a especificação do LED: se for LED catodo comum, se conecta ao GND; se for LED anodo comum, ao 5V. Assim como no LED comum, é necessário associar cada LED a um resistor, para que não ultrapasse tensão e corrente máxima.
- Como funcionam as conexões do potenciômetro?
 - É necessário conectá-lo ao GND, à fonte 5V e a um pino no Arduino de entrada analógica, que pode ser A0, A1, A2, A3, A4 ou A5.
- Como será a leitura do potenciômetro e a definição de cores do LED RGB?
 - Como o potenciômetro tem um intervalo de 0 a 1023, a ideia seria dividi-los para que cada cor seja emitida em uma determinada faixa dentro desse intervalo. Há diversas possibilidades de quantidade e escolha de cores, sendo necessário apenas testar na montagem virtual e física se funciona adequadamente. O exemplo abaixo constituirá de 4 opções de cores.
- Parte 2: Montagem virtual do circuito
 - Em relação à dinâmica da atividade, é sugerido que se dê maior autonomia aos alunos para construir a simulação. Assim, o professor indica a estrutura de o que e como montar e depois acompanha o andamento e possíveis dúvidas dos alunos. Uma das possíveis estruturas seria:
 - 1º - Conectar o GND e o 5V à protoboard
 - 2º - Definir a entrada e conectar as pernas do componente. Verificar quais pinos do Arduino serão utilizados de acordo com o tipo (digital, analógico).
 - 3º - Definir a saída e conectar as pernas do componente. Não esquecer de incluir possíveis resistores. Verificar quais pinos do Arduino serão utilizados de acordo com o tipo (digital, analógico).
 - 4º - Revisar o circuito para verificar se todos os componentes estão conectados da forma adequada e com os valores certos. Deixar o circuito

mais organizado possível (evitar passar jumpers por cima do Arduino, fazer esquemas de cores de jumpers, etc. como abordados no Guia 4)

● Parte 3: Programação

- O nível de programação é médio; portanto, seria interessante orientar os alunos a criarem os blocos referentes à primeira cor.
 - Como a cor emitida dependerá da faixa do potenciômetro, utilizaremos condicional (bloco *se*, já que temos mais de duas possibilidades para definir).
 - A condição será estabelecida por determinado intervalo do potenciômetro (entrada analógica). Por isso, colocaremos os blocos de entrada *ler pino analógico x* de tal forma que seja maior que um número e menor que outro. Como são 4 intervalos, pode-se dividir 1024 e teremos as aias definidas para cada cor.
 - Para a saída, a plataforma fornece um bloco específico para o LED RGB. Assim, basta indicar os 3 pinos relacionado a cores e selecionar a cor desejada. Vale ressaltar que as opções de pino no Arduino para esse bloco são reduzidas (apenas as PWM) e, por isso, é importante rever se foram escolhidos os pinos certos para o LED RGB.
 - É importante acrescentar um bloco para conferir se a cor do LED na simulação coincide com o programado. Para isso, utiliza-se o monitor serial, em que mostraremos na tela por escrito qual é a cor programada para aquela faixa. Caso não coincida, recomenda-se modificar a ordem dos pinos no bloco do LED RGB; se ainda assim não refletir, trocar a cor escolhida. Incluir uma espera de tempo para que haja uma pausa entre as impressões no monitor serial. Para visualizá-lo no momento da simulação, deve-se estar com a aba de *Código* aberta e clicar em *Monitor serial* no canto inferior esquerdo. Serão pausadas as impressões ao parar a simulação.
 - Fazer o mesmo processo com as demais cores, atentando-se aos intervalos do potenciômetro. Não esquecer de comentar as partes do código com o bloco de notação.

- Parte 4: Montagem física/protótipo do circuito
 - Idem aos projetos anteriores. É possível ver o monitor serial da própria IDE do Arduino. Na seção de *Ferramentas* do canto superior esquerdo, clique em monitor serial e acompanhe se as cores programadas coincidem com a o protótipo.
- (Extra) E se ao invés de emitir a 4ª cor na faixa especificada, você quer que o LED RGB simplesmente apague? Como poderíamos alterar o código para fazer essa atualização?
 - Aqui o objetivo é que os alunos reflitam e relembrem das aulas anteriores. Nesse caso, não se utiliza o bloco específico para LED RGB, mas pode-se incluir os blocos de saída para definir os 3 pinos como BAIXO, como feito nos últimos projetos. Assim, mostra-se que mesmo com as facilidades dos blocos mais específicos, podemos usar outros blocos mais genéricos para ampliar as funcionalidades.
- As sugestões de projetos para essa aula encontram-se na atividade *Projeto 3: controlar LED RGB pelo potenciômetro*. Utilizar como apoio apenas ao professor. Link para acessar a sala de aula: <https://www.tinkercad.com/joinclass/U1AY9S3732FC>, utilizando o apelido *teste4960*.

□ Avaliação da aula

- Os alunos conseguiram entender a principal diferença entre sinais digitais e analógicos?
- Os alunos entenderam como funciona o potenciômetro e por que se trata de uma entrada analógica?
- Os alunos conseguiram desenvolver o projeto com o auxílio do professor?
- Os alunos tentaram fazer a atividade extra? Tiveram dificuldades em saber qual bloco usar?

Aula 8: Conhecendo o sensor ultrassônico

□ Objetivos da aula sugerida

- Introduzir as principais características do sensor ultrassônico hc-sr04 como estrutura, funcionamento e aplicações.

- Promover um ambiente em que os alunos desenvolvam a construção do circuito proposto de forma mais independente.

□ Recursos para a aula

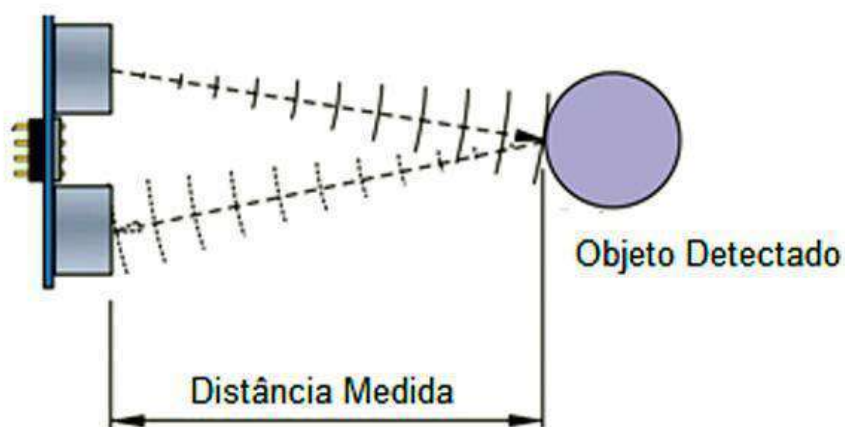
- Dispositivos com acesso à internet
- Projetor (opcional)
- Arduino Uno
- Cabo USB 2.0 A B
- LED
- Jumpers
- Protoboard
- Resistor 220 ohms
- Sensor ultrassônico HC SR04
- Objeto

□ Descrição das atividades

- Introdução ao sensor ultrassônico
 - Duração: 15 min
 - Objetivo: apresentar as características básicas de um sensor ultrassônico (finalidade, funcionamento, aplicações, etc.) para construir posteriormente um circuito com esse componente.
 - A dinâmica sugerida da atividade é trazer uma aplicação do cotidiano (por exemplo, um vídeo de um carro sendo estacionado com sensor de ré), questionar os alunos se eles sabem como funciona e depois mostrar o componente por trás dessa tecnologia. O sensor de ré é diferente do sensor que utilizaremos nesse projeto, porém a lógica de funcionamento é a mesma (cálculo da distância através de tempo e propagação do eco).
 - Informações relevantes sobre o sensor ultrassônico HC SR04
 - O que é? É um sensor que opera com sinais ultrassônicos, ondas com frequências bem altas (40kHz), capaz de medir distância entre o sensor e um objeto sem ter contato entre os dois, com alcance de 2 a 400 cm. O ouvido humano detecta ondas mecânicas até 20 kHz.

- Como funciona? O sensor envia um sinal e informa quanto tempo levou para retornar. A partir desse dado e sabendo que a velocidade do som é de 340m/s, programa-se no Arduino a distância entre o sensor e o objeto. Em relação a sua estrutura, possui geralmente 4 pinos: Vcc, Gnd, Trig e Echo, sendo que o trig (trigger) serve como ponte entre o recebimento do pulso do arduino e a emissão de pulsos do sensor e o echo que retorna os pulsos para o Arduino e o intervalo de tempo entre envio e retorno do sinal ultrassônico.

Sensor Ultrassônico



Fonte: blog.baudaeletronica.com.br/arduino-com-sensor-ultrassonico-hc-sr04/

$$Distancia = \frac{(Tempo_{echo\ nível\ alto} * Velocidade_{som})}{2}$$

- Quais são as aplicações? Detecção de presença, cálculo de comprimento e largura, posicionamento de robôs, detecção de obstáculos.
- Sinalizando a aproximação de um objeto
 - Duração: 25 min.
 - Objetivo: construir um circuito em que determinado LED acende de acordo com a distância entre um objeto e o sensor ultrassônico. Reforçar a estrutura que foi feita na aula passada, em que os alunos possuem maior autonomia no desenvolvimento do projeto.
 - Parte 1: Montagem virtual do circuito

- Cada aluno pode definir como prefere fazer a sinalização pelos LEDs conforme o objeto se aproxima (acrescentar número de LEDs acesos, acender LEDs com cores diferentes, utilizar LED RGB, imprimir mensagem de aviso no monitor serial, etc.)
- Reforçar os passos que os alunos podem seguir para montar o circuito (alimentação e terra, componente (s) de entrada, componente (s) de saída e revisão).
- **Parte 2: Programação**
 - A novidade é que o sensor ultrassônico possui um bloco específico de leitura de distância. Para utilizá-lo, devem ser indicados os pinos de trigger e de echo e a unidade de medida. Além disso, é válido torná-lo uma variável para que, se for necessário alterar os pinos, não precisar trocar de todas as condições.
 - É importante que os alunos tentem fazer individualmente esse código, já que os blocos utilizados são similares aos da aula anterior. Caso estejam com dúvidas, retomar os conceitos para fixar os conteúdos.
- **Parte 3: Montagem física do circuito**
 - Idem às aulas anteriores.
- As sugestões de projeto para essa aula encontram-se na atividade *Projeto 4: sinalizar aproximação de um objeto*. Utilizar como apoio apenas ao professor. Link para acessar a sala de aula: <https://www.tinkercad.com/joinclass/U1AY9S3732FC>, utilizando o apelido *teste4960*.
- **Desafio em dupla: identificação de erros**
 - Duração: 20 min
 - Objetivo: desenvolver a capacidade dos alunos de analisarem um circuito e terem uma linha de raciocínio para testar e corrigir. Discutir e compilar os principais erros que ocorrem em um projeto de construção de circuitos como uma forma de aprendizagem coletiva.
 - Questionar a turma: quais são os principais erros que poderiam acontecer nesse circuito, relacionados à simulação, à programação e ao protótipo? E nos projetos

anteriores? Discutir com toda a sala para listar todos os itens comentados e fazer um compilado como material de consulta. Uma alternativa seria disponibilizar simulações com erros e pedir para eles identificarem e os resolverem.

Avaliação da aula

- Os alunos entenderam o funcionamento básico do sensor apresentado?
- Os alunos conseguem de forma mais independente criar as 3 partes do projeto?
- Os alunos conseguiram listar alguns erros comuns no desenvolvimento de circuitos? Entenderam por que são erros e como podem ser corrigidos?

Aula 9: Conhecendo o servo motor

Objetivos da aula sugerida

- Orientar os alunos a procurarem informações sobre um componente e compilar os aprendizados de forma a não depender exclusivamente das falas do professor.
- Assim como na aula 8, deixar os alunos desenvolverem a construção do circuito proposto de forma mais independente.

Recursos para a aula

- Dispositivos com acesso à internet
- Projetor (opcional)
- Arduino Uno
- Cabo USB 2.0 A B
- Jumpers
- Protoboard
- Servo motor sg90
- Potenciômetro 10k ohm
- Capacitor 100 μ F e fonte externa (se criar circuito com +1 servo motor)

Descrição das atividades

- Pesquisando sobre o servo motor

- Duração: 25 min
- Objetivo: discutir e listar com os alunos quais são as informações básicas a serem pesquisadas para aprender sobre um componente; onde e como obter essas informações.
- A dinâmica sugerida da atividade é promover um momento de interação da turma, em que os alunos com o apoio do professor citem o que é importante saber sobre um componente. O ideal é que se relembre das aulas anteriores o que foi ensinado e o que foi aplicado nos projetos, como “Qual é a principal função? ”, “Como funciona? ”, “Quais as conexões? ”, “É necessário ter mais um componente associado? ”, “Quais são as aplicações? ”. Em relação aos recursos, os alunos poderiam pesquisar em apostilas, livros e professores da área, mas provavelmente o mais acessível seria na internet. Assim, é importante orientar os alunos a buscarem em sites de maior confiança e acessarem mais de um site. Feito isso, os alunos podem pesquisar individualmente ou em duplas, tendo orientação do professor. Por fim, a classe pode discutir sobre as informações coletadas e comentar como foi a pesquisa (dificuldades, facilidades, etc.)
- Informações importantes sobre o micro servo motor 9g sg90
 - É um servo motor (um recurso que transforma energia elétrica em mecânica e consegue alcançar uma posição com precisão e mantê-la). Normalmente é capaz de girar até 180°, mas alguns podem girar até 1 volta completa.
 - A precisão do servo motor deve-se ao mecanismo que verifica a posição: se estiver correta, mantém-se; se não, gira até a posição desejada.
 - O servo que iremos utilizar possui 3 conexões: o de alimentação (geralmente vermelho), que deve ser conectado ao pino 5V do Arduino; o de ground (marrom ou preto), que deve ser conectado ao GND do Arduino e o de sinal (laranja ou amarelo), que pode ser conectado a um pino de saída.
 - No Arduino, é importante verificar a voltagem de operação do servo motor. Pode ser que a alimentação pelo cabo USB não seja suficiente para o servo, então se opta por uma bateria ou uma fonte externa. Além disso, pode-se utilizar um capacitor associado ao servo, já que o capacitor reduz a variação de tensão provocada pelo mecanismo do

servo. Ao iniciar o movimento, o servo puxa muito mais corrente do que enquanto está em movimento, gerando essa variação na tensão.

- No Arduino, permite realizar diferentes projetos com movimentos rotativos de até meia ou 1 volta. Na indústria, é utilizado na robótica, no automobilismo, etc.

- Acionando o servo motor

- Duração: 15 min
- Objetivo: criar um circuito em que o servo motor faz uma meivolta e depois retorna à posição inicial. Manter a autonomia dos alunos em desenvolver o código e circuito virtual e físico, mas sempre acompanhado do auxílio do professor.
- Parte 1: Montagem virtual do circuito
 - A dinâmica sugerida é similar às aulas anteriores, em que os alunos seguem os passos macro para simulação. No caso de utilizar apenas 1 servo de 4.8 ~ 6V de tensão, não é necessário incluir o capacitor e a fonte externa.
- Parte 2: Programação
 - O servo também possui um bloco específico, que inclui a sua biblioteca própria (<Servo.h>). Nele, é possível indicar a rotação desejada servo.
 - Trata-se de um código curto, porém é importante se atentar aos blocos de espera (controle) após a rotação. Além disso, seguir a boa prática de comentar o código

- Parte 3: Montagem física/protótipo do circuito

- Idem às aulas anteriores.

- Desafio: controlando o motor

- Duração: 25 min
- Objetivo: criar um circuito em que o servo motor é controlado por um potenciômetro. Destacar a possibilidade de incremento de um circuito.
 - Parte 1: montagem virtual do circuito
 - Nessa parte, os alunos terão apenas que acrescentar o potenciômetro e conectá-lo ao Arduino corretamente.

- Parte 2: Programação
 - O grau de dificuldade dessa parte é maior; por isso, é importante orientar os alunos de forma geral para que eles consigam desenvolver o código. As posições do potenciômetro, que é controlada manualmente, devem estar associadas aos graus que o servo alcança em seu intervalo. Assim, após definir a leitura do potenciômetro, pode-se definir a variável de ângulo do servo, que será mapeada para abranger a meia volta (0 a 180°). Posteriormente, os blocos são parecidos ao programa anterior.
 - Recomenda-se que os alunos insiram a opção de impressão no monitor a fim de visualizar o ângulo daquele instante.
- Parte 3: Montagem física/protótipo do circuito
 - Atentar-se à necessidade ou não de acrescentar capacitor para que haja menor oscilação de tensão entre o início e o servo operando. O servo pode começar a trepidar muito. Atentar-se para conectar as pernas do capacitor de forma correta (catodo negativo com GND e anodo positivo com 5V do Arduino ou a fonte externa.)
 - As sugestões de projetos para essa aula encontram-se na atividade *Projeto 5: acionar e controlar um servo motor*. Utilizar como apoio apenas ao professor. Link para acessar a sala de aula: <https://www.tinkercad.com/joinclass/U1AY9S3732FC>, utilizando o apelido *teste4960*.

□ Avaliação da aula

- Os alunos citaram as principais questões para se pesquisar quando nos deparamos com um novo componente?
- Os alunos conseguiram achar as principais informações, ter senso crítico de ler, refletir e criticar as informações?
- Os alunos conseguiram fazer os dois circuitos propostos? Tiraram suas dúvidas durante as atividades?

Aula 10: Revisão

□ Objetivos da aula sugerida

- Retomar alguns conceitos importantes abordados nas aulas até agora
- Reforçar o pensamento crítico, a capacidade de resolver problemas e o trabalho em equipe

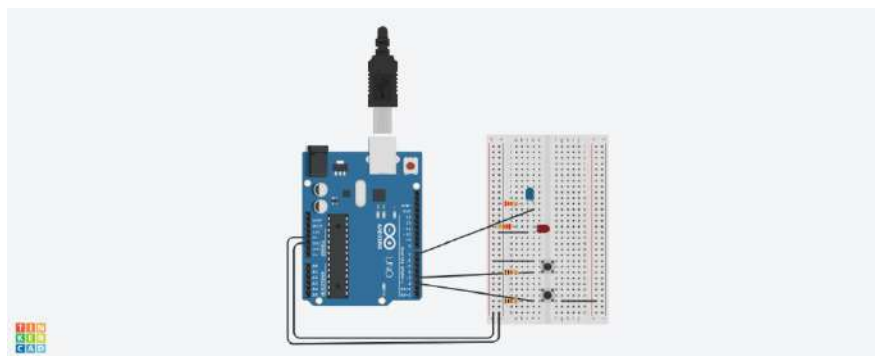
□ Recursos para a aula

- Dispositivos com acesso à internet
- Projetor (opcional)
- Arduino Uno
- Cabo USB 2.0 A B
- LED
- Jumpers
- Protoboard
- Resistor 220 ohms

□ Descrição das atividades

- Exercício 1: Por que o programa não está funcionando corretamente?
 - Objetivo: desenvolver a capacidade dos alunos de investigar, propor soluções e testar. Revisar tópicos relevantes que foram trabalhados durante as aulas anteriores.
 - Duração: 20 min.
 - Nessa atividade, o professor pode criar um projeto tendo em vista suas percepções durante as aulas (necessidade de reforço de certo tema, por exemplo) ou pode adaptar do projeto sugerido tinkercad.com/classrooms/km3QrLj1xPq/activities/4r5MfGCVIHt. Nesse projeto, são recapitulados a parte de programação e sobretudo a montagem virtual do circuito, como sinal analógico vs. digital, orientação das conexões da protoboard, ligações dos pinos de GND e 5V do Arduino na protoboard, a conexão dos pinos do LED e do botão, entre outros. Além disso, têm

possibilidades de ajustes não obrigatórias, mas que auxiliam na clareza da montagem (sistema de cores dos jumpers, evitar que elementos se sobreponham, entre outros). Tratam-se de tópicos básicos que foram abordados no início do curso, mas que são muito importantes para que os alunos fiquem atentos a esses detalhes quando forem criar um circuito e saibam verificá-los de forma mais ágil. Mas vale reforçar que o ideal é o professor entender qual exercício seria o mais adequado para os alunos nesse momento.



Fonte: autoral.

- Exercício 2: Desenvolvendo uma solução diante de uma situação
 - Objetivo: trazer uma situação em que os alunos reflitam e discutam sobre como podem utilizar projetos eletrônicos para resolver um problema ou melhorar algo. Focar nas discussões, busca por informações e pensamento lógico da solução, não necessariamente na execução do projeto por conta da limitação do tempo.
 - Duração: 20 min.
 - Assim, como na atividade anterior, o professor pode criar uma situação ou adaptar a sugerida a seguir. É importante que os alunos reflitam se é possível de fato aplicar um projeto eletrônico, qual seria a estrutura base da montagem (componentes de entrada e saída) e da programação (lógica, sequência). A dinâmica dessa atividade é bastante flexível, podendo ser desenvolvido pela turma em conjunto, pequenos grupos ou individual. Pode-se utilizar lousa, papéis e/ou computadores para anotar e estruturar as ideias.
 - Situação sugerida: Seu amigo João costuma fazer iogurte natural em sua casa. Quando ele comentou como ele faz esse processo, você ficou pensando se teria alguma forma de deixar isso mais fácil a partir do que você está aprendendo no curso de Robótica Legal. Reflita e discuta com seus colegas sobre essa situação.

- Obs: Mesmo que os alunos não saibam como é feita a receita proposta, incentiva-se a busca pelas informações necessárias na internet. Além disso, é importante acompanhá-los, orientá-los e questioná-los durante a atividade, buscando entender como seria a ideia proposta da solução. Nesse processo, é importante incentivá-los a buscar os componentes que poderiam utilizar, já que não atuamos com todos os componentes no Tinkercad.
 - Direcionamento sugerido: no momento em que é necessário esperar para que o leite esfrie, poderia ter um sistema que avisasse quando o leite chegasse na temperatura certa. Assim, pode-se utilizar o sensor de temperatura, que serviria como entrada, e a saída poderia ser um LED que acende, um buzzer que dispara um som, etc. Em relação à programação, poderia ser implementado uma condicional: se a temperatura for x °C, piscar o LED 3 vezes para avisar a pessoa.
 - Posteriormente, pode-se fazer mais questionamentos aos alunos: quais poderiam ser outras aplicações com essa lógica?
- Percepção do curso
 - Objetivo: entender e fomentar as opiniões dos alunos em relação à dinâmica das aulas e conteúdos abordados no curso para que o professor saiba do panorama geral da aula e tenha material para aprimorar as aulas seguintes e também dos próximos cursos.
 - Duração: 10 min.
 - O formulário de Avaliação de Diagnóstico sugerido é mostrado abaixo.

O que você está achando do curso até agora?

Já passamos por várias aulas e aprendemos um pouco sobre programação em blocos e projetos com Arduino, não é mesmo? Por isso, é muito importante a sua opinião em relação ao andamento do curso. É rapidinho!

Perguntas

Respostas

| | |
|--|---|
| <p>Qual afirmação você mais se identifica em relação ao curso até agora?</p> | <p><input type="checkbox"/> Estou gostando do formato das aulas. Acho interessante o conteúdo abordado.</p> <p><input type="checkbox"/> Estou gostando do formato das aulas, mas não acho interessante o conteúdo abordado.</p> <p><input type="checkbox"/> Não estou gostando do formato das aulas, mas acho interessante o conteúdo abordado.</p> <p><input type="checkbox"/> Não estou gostando do formato das aulas e não acho interessante o conteúdo abordado</p> |
| <p>Se quiser, conte-nos por que escolheu a afirmação acima.</p> <p>O objetivo dessas perguntas é contribuir para que o curso de Robótica Legal esteja mais próximo do contexto dos alunos. Fique à vontade para expressar suas percepções (se preferir, pode falar diretamente com o professor).</p> | <p>Aberta.</p> |

☐ Avaliação da aula

- Os alunos conseguiram identificar os erros do exercício 1? Conseguiram fazer o projeto funcionar adequadamente? Ajustaram as boas práticas de montagem e programação?
- Os alunos conseguiram desenvolver um pensamento lógico para o exercício 2? Conseguiram responder sobre quais seriam a entrada e saída da solução proposta?

Aula 11: Projeto - idealização

□ Objetivos da aula sugerida

- Contextualizar e discutir a situação-problema para que a turma entenda a motivação do projeto
- Estimular os grupos a buscarem por uma solução à situação-problema fornecida

□ Recursos para a aula

- Dispositivos com acesso à internet
- Projetor (opcional)
- Papéis, lápis, caneta

□ Descrição das atividades

- Introdução ao projeto
 - Objetivo: apresentar brevemente para a turma o planejamento das últimas aulas do curso, ressaltando as principais etapas que desenvolverão para o projeto.
 - Duração: 5 min
 - É sugerido que o professor forneça uma lista das principais atividades do projeto como material de apoio das próximas aulas para cada grupo. Assim, eles podem ter uma visão mais geral do projeto e autonomia maior no andamento das atividades. Um exemplo sugerido:
- Situação-problema
 - [] Sabemos qual é a situação-problema? Quem, quando, onde, de que forma impacta?
- Nossa solução
 - [] O que é? Quem, quando, onde, de que forma impacta?
 - [] Como funciona?
 - [] Quais são os materiais necessários para criar essa solução? Algum deles ainda não sabemos como funciona?

- Montagem virtual
 - Quais são os componentes necessários para o circuito eletrônico? O que é entrada e saída ?
 - Sabemos como utilizar cada componente? Quais são as conexões para que funcione? Se necessitam de resistores?
 - Revisamos o circuito para ver se não há nenhuma conexão solta? A lógica do circuito é coerente?

- Programação
 - Qual é a sequência de ações (algoritmo) para que o circuito que fizemos anteriormente funcione?
 - Quais tipos de blocos utilizaremos? Quais blocos? Faz sentido o uso de variável neste código?
 - Fizemos os comentários ao longo do código?
 - Testamos o código? É necessário fazer ajustes?
 - A simulação virtual está funcionando adequadamente?

- Montagem física/prototipagem do circuito
 - Fizemos e revisamos a montagem do circuito físico? Não esquecer de checar se os jumpers foram conectados corretamente, se os pinos utilizados no circuito físico correspondem ao do código
 - Foi possível compilar e rodar o código na IDE do Arduino?
 - Testamos o código no circuito físico? É necessário fazer ajustes?
 - A simulação real está funcionando adequadamente?

- Montagem física/prototipagem do projeto inteiro
 - É necessário unir o circuito a outros materiais para que o projeto funcione no modo esperado? Se sim, o que é necessário fazer e como? Pedir ajuda para o professor
 - Testamos o código com o projeto inteiro? É necessário fazer ajustes?
 - A simulação real inteira está funcionando adequadamente?

- Apresentação do projeto e feedback

- [] Planejar o que cada um vai apresentar: falar sobre a situação-problema, qual foi o desenho da solução, apresentar o projeto em funcionamento e os aprendizados
 - [] Dar e receber feedback dos alunos e do professor
- Contextualização da situação-problema-
- Objetivo: apresentar para a turma a situação-problema e incentivar a discussão de tópicos relacionados a esse tema
 - Duração: 20 min
 - Assim como nas aulas anteriores, é apresentada uma sugestão de atividade. Nessa última etapa do curso, que engloba 4 aulas, será abordado a questão de lixeiras, mas cabe ao professor entender e adaptar o conteúdo da aula. A situação problema sugerida é a seguinte:
 - No Brasil, a questão da importância de coletar lixo e descartá-lo de forma adequada vem sendo debatida ao longo dos anos e muitas iniciativas públicas e privadas surgiram. Uma das primeiras etapas que é importante se atentar é a escolha da lixeira nos locais. São utilizados muitos tipos de lixeiras, com formato, tamanhos e tampas diferentes. A partir do que vocês vêm em escolas, hospitais, parques, restaurantes, quais são os modelos encontrados? Esses modelos seriam os ideais para serem utilizados, considerando as questões de meio-ambiente, higiene e acessibilidade?
 - Em relação à dinâmica, a atividade pode ser aplicada com um grupo geral da sala ou já separado pelos grupos do projeto (4-5 pessoas). É importante que a turma discuta quais os modelos que vem no cotidiano e quais são as vantagens e desvantagens de cada um. Os conteúdos na internet podem auxiliar nesse processo de embasamento das questões levantadas. Em relação ao meio-ambiente, é importante discutir sobre materiais recicláveis e não recicláveis, coleta de lixo em diferentes regiões do Brasil; em relação à higiene, é importante pensar na menor exposição do lixo aberto e no menor contato ao jogar um lixo; por fim, em relação à acessibilidade, é importante pensar na inclusão das pessoas que fazem uso.

- **Busca pela solução**

- Objetivo: cada grupo discutir ideias de soluções voltadas ao conhecimento adquirido durante o curso e escolher aquela que mais apropriada. Aprofundar os detalhes da solução proposta.
- Duração: 30 min.
- É preferível que os grupos façam essa etapa sem uso de internet para estimular a criatividade coletiva. Pode-se utilizar cartolinas, post-its que auxiliem os alunos a discutir e encontrar uma solução à situação-problema proposta. Os alunos poderiam fazer um desenho no final da concepção inicial do projeto deles. Aqui é importante eles refletirem: a partir do que aprendemos no curso, o que podemos propor como solução? A quem atenderia em quais situações? Quais são os pontos positivos e negativos? Conseguimos contornar algum ponto negativo apresentado? Como seria o funcionamento da solução? Quais os materiais necessários para fazer essa solução?

- **Fechamento da aula**

- Objetivo: retomar o que foi feito nessa aula e mencionar quais serão os próximos passos
- Duração: 5 min.
- Utilizar a checklist para que os grupos deem check nas atividades feitas durante a aula e visualizarem quais são as próximas atividades

- **Avaliação da aula**

Os alunos interagiram no momento de discussão da situação-problema? Foram levantadas as questões relacionadas ao meio-ambiente, higiene e acessibilidade?

Os alunos desenvolveram ideia (s) de solução? Construíram um material que mostrasse a concepção inicial do grupo?

Aula 12: Projeto - desenvolvimento no ambiente virtual

□ Objetivos da aula sugerida

- Desenvolver a ideia do projeto na prática, construindo a montagem virtual do circuito e a programação
- Retomar os processos que foram abordados nas aulas anteriores de como fazer essas atividades propostas sem a necessidade de instruções passo-a-passo de como fazer o projeto rodar.

□ Recursos para a aula

- Dispositivos com acesso à internet
- Projetor (opcional)

□ Descrição das atividades

- Visualização da etapa atual em relação ao projeto e montagem virtual do circuito e programação
 - Objetivo: situar os alunos de quais etapas serão desenvolvidas nessa aula, que são a construção da montagem virtual e programação do projeto
 - Duração: 55 min.
 - Como os projetos de cada grupo podem ser diferentes, é preferível que as atividades não sejam divididas por intervalo de tempo proposto. Os grupos terão a checklist para se orientarem e saber quais pontos são essenciais de serem definidos para prosseguir no projeto. O papel do professor é acompanhar os grupos de como eles estão estruturando o circuito e tirar possíveis dúvidas.
 - A ideia do projeto foi baseada nesse material: Criando a sua própria lixeira inteligente <https://inovaedu.tech/lixeira-inteligente/>
 - As sugestões de projetos para essa aula encontram-se na atividade *Projeto final: lixeiras*. Utilizar como apoio apenas ao professor. Link para acessar a sala de

aula: <https://www.tinkercad.com/joinclass/U1AY9S3732FC>, utilizando o apelido *teste4960*.

- Fechamento da aula
 - Objetivo: retomar o que foi feito nessa aula e mencionar quais serão os próximos passos
 - Duração: 5 min.
 - Utilizar a checklist para que os grupos deem check nas atividades feitas durante a aula e visualizarem quais são as próximas atividades

□ Avaliação da aula

- Os alunos conseguiram transformar a concepção criada na aula anterior em um projeto no tinkercad? Conseguiram fazer a montagem virtual do circuito e a programação?
- Os alunos fizeram a simulação virtual? Conseguiram encontrar possíveis erros e ajustá-los?

Aula 13: Projeto - prototipagem

□ Objetivos da aula sugerida

- Construir o protótipo do circuito simulado na aula anterior, além de fixá-lo ao objeto associado.
- Testar e fazer ajustes até que o projeto funcione adequadamente

□ Recursos para a aula

- Dispositivos com acesso à internet
- Projetor (opcional)
- Arduino Uno
- Cabo USB 2.0 A B
- Jumpers
- Protoboard

- Componentes específicos eletrônicos (Resistor, LED, sensor ultrassônico, servo motor com as hélices , buzzer, etc.), lixeira, arame, conector de fio, furadeira, régua, caneta.

□ **Descrição das atividades**

- Visualização da etapa atual em relação ao projeto e prototipagem
 - Objetivo: construir o protótipo do circuito baseado no projeto simulado no tinkercad e do projeto como um todo a partir do desenho feito na 1ª aula do projeto
 - Duração: 55 min
 - É importante repassar as orientações gerais de montagem física de circuitos para evitar os erros recorrentes e a queima de algum componente. Para fazer possíveis furos e assim fixar componentes na lixeira, é essencial que o professor faça essa tarefa. Por isso, recomenda-se que o professor passe de grupo em grupo para ver o andamento e aproveitar para já fazer as tarefas necessárias. Caso tenha grupo (s) que finaliza (m) essa etapa antes do (s) outro (s), pedir para que eles auxiliem esses grupos.
 - Lembrar os alunos de escreverem o que eles fizeram nessa aula, pontos positivos e negativos e se tem algum ponto para retornar dessa aula na aula seguinte
- Fechamento da aula
 - Objetivo: retomar o que foi feito nesta aula e mencionar quais serão os próximos passos
 - Duração: 5 min.
 - Utilizar a checklist para que os grupos deem check nas atividades feitas durante a aula e visualizarem quais são as próximas atividades

□ **Avaliação da aula**

- Os alunos conseguiram fazer o circuito eletrônico funcionar adequadamente? Conseguiram testar e fazer os ajustes necessários?
- Os alunos conseguiram testar o projeto todo? Fizeram possíveis ajustes?

Aula 14: Projeto - apresentação final e conclusão do curso

□ Objetivos da aula sugerida

- Cada grupo apresentar seu projeto, destacando o início (problema e concepção da solução), meio (desenvolvimento) e fim (entrega do projeto) e os aprendizados gerais.
- Mostrar a importância do feedback e incentivar o feedback entre os alunos
- Obter um feedback final da percepção do projeto e do curso no geral

□ Recursos para a aula

- Idem à aula 13

□ Descrição das atividades

- Visualização da etapa atual em relação ao projeto e finalização para apresentação
 - Objetivo: mostrar que essa é a etapa final e dar um tempo para possíveis testes finais e acabamentos. Os grupos devem-se organizar para definir o que cada um abordará na apresentação. Recomenda-se a todos falarem.
 - Duração: 15 min.
- Apresentação dos grupos
 - Objetivo: entender a percepção dos alunos de como foi trabalhar no projeto como um todo. Praticar a capacidade de se expressar e expor um projeto diante de um público.
 - Duração: 30 min.
 - Além da demonstração da solução funcionando, é importante analisar o que os grupos apresentam e se eles compreenderam como é feito um projeto eletrônico de maneira geral, mesmo que o produto final não saiu como esperado. O professor pode fazer algumas perguntas, auxiliando na apresentação do projeto e fazendo questionamentos para o grupo. É válido incentivar que os grupos também façam perguntas e comentários para praticarem e entenderem a importância de ouvir os outros grupos e fornecer feedback.
- Feedback final e fechamento do curso

- Objetivo: professor entender individualmente qual foi a percepção dos alunos nessas últimas aulas de projeto e do curso em si.
- Duração: 15 min.
- Formulário de Avaliação de Diagnóstico sugerido é mostrado no final desta aula.

| Queremos ouvir sua opinião final do curso! | |
|--|---|
| <i>O curso está chegando ao fim e queremos te ouvir uma última vez. Parabéns por ter chegado até aqui, sabemos que esse curso não é fácil, mas estamos buscando cada vez mais desenvolver os alunos para o contexto atual.</i> | |
| Perguntas | Respostas |
| <p>O que você achou do projeto final? Escolha a(s) opção(es) que você mais se identifica.</p> | <p><input type="checkbox"/> Desinteressante <input type="checkbox"/> Interessante <input type="checkbox"/> Desafiador <input type="checkbox"/> Muito difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Fácil <input type="checkbox"/> Muito fácil <input type="checkbox"/> Outro: _____</p> |
| <p>O que você achou do curso no geral?</p> | <p><input type="checkbox"/> Estou gostando do formato das aulas. Acho interessante o conteúdo abordado. <input type="checkbox"/> Estou gostando do formato das aulas, mas não acho interessante o conteúdo abordado. <input type="checkbox"/> Não estou gostando do formato das aulas, mas acho interessante o conteúdo abordado. <input type="checkbox"/> Não estou gostando do formato das aulas e não acho interessante o conteúdo abordado</p> |

| | |
|--|----------------|
| <p>Se quiser, conte-nos um pouco mais sobre a sua percepção do projeto final e do curso no geral. O objetivo dessas perguntas é contribuir para que o curso de Robótica Legal esteja mais próximo do contexto dos alunos. Fique à vontade para expressar suas percepções, o que gostou e o que não gostou.</p> | <p>Aberta.</p> |
|--|----------------|

Avaliação da aula

- Os alunos conseguiram finalizar o projeto?
- Os alunos fizeram a apresentação e falaram sobre as principais etapas e aprendizados?
- Qual foi a percepção dos alunos coletada pelo formulário?