Aula 08 – Técnicas de programação: Sketch com abas

Módulo 3

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Cleiton Rosa

Darice Alessandra Deckmann Zanardini

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Kellen Pricila dos Santos Cochinski

Projeto Gráfico e Diagramação

Edna do Rocio Becker

2024

Introdução

A organização de códigos de programação possibilita, ao desenvolvedor e comunidade, melhor visualização dos comandos e funções aplicados, o que auxilia também na reorganização do código conforme aprimoramento dos projetos de Robótica. Para o Arduino, a formatação visual não interfere na compilação do programa, mas sim a sintaxe aplicada para uma correta compilação do código.

Para os humanos, além da sintaxe adequada – o que nos mostra que o código foi escrito corretamente para ser compilado pelo Arduino – um código bem organizado possibilita a percepção lógica sobre as intenções do projeto e facilita eventuais ajustes e compartilhamentos.

Objetivos desta aula

- Aprimorar a organização de sketches e a criação de funções com a utilização do recurso de abas no Arduino Editor.

Lista de materiais

- Computador e/ou notebook.

- **Roteiro da aula**

Contextualização

Como vimos na **Aula 07 - Recursos das bibliotecas**, as bibliotecas para Arduino, além de otimizarem a programação do Arduino e seus componentes, possuem recursos valiosos para que possamos compreender suas funcionalidades, aprender sua sintaxe e explorar suas possibilidades de aplicação em projetos autorais.

Quando instalamos uma biblioteca pelo software Arduino IDE ou a favoritamos pelo Arduino IDE Online, os desenvolvedores podem disponibilizar exemplos de programação e uma maneira eficaz de compreender uma biblioteca Arduino é analisar tanto estes exemplos quanto seus arquivos **.h** (principalmente) e **.cpp** (para avançar um pouco mais no “interior” das funções) das bibliotecas.

Explorar esses arquivos oferece insights valiosos sobre a estrutura interna da biblioteca e suas funcionalidades, nos capacitando para a utilização de bibliotecas de forma mais eficaz e a adaptando as programações às necessidades específicas de cada projeto que podemos desenvolver e aprimorar.

Avançando na programação

Como vimos na última aula, o arquivo **.h d**efine a interface pública da biblioteca e inclui declarações de classes, estruturas, variáveis e funções que podem ser acessadas por outros arquivos; o arquivo **.cpp** contém a implementação das funções declaradas no **.h** e encontramos a lógica por trás de cada função, permitindo uma compreensão mais detalhada de como a biblioteca funciona.

A oportunidade de explorar os arquivos das bibliotecas nos mostrou também outras técnicas e recursos, como o funcionamento das **abas** no Arduino IDE

No Arduino IDE, as abas aparecem na parte superior do editor de código e representam arquivos individuais que compõem o projeto. Cada aba contém um conjunto específico de instruções ou funções, permitindo sua **decomposição**, ou seja, uma divisão lógica do código em partes menores e mais gerenciáveis, mais fáceis de entender.

Figura 01 – Abas na programação do exemplo GFXcanvas

Fonte: Arduino IDE Online

As abas no Arduino IDE facilitam a navegação entre diferentes partes do código, tornando mais fácil localizar e modificar seções específicas. Elas também contribuem para uma organização mais clara do projeto, ajudando os desenvolvedores a manterem o código estruturado e legível tanto para si mesmo, ao revisitar a programação, quanto para a comunidade, quando o código for compartilhado.

Vantagens	
A separação do código em seções lógicas facilita a organização, manutenção e busca por partes específicas do projeto, como no caso de uma aba para controle de motores, uma para leitura de sensores, uma para constituição de matrizes etc, deixando o programa mais gerenciável.	Organização do código e legibilidade
Com o desenvolvimento de projetos em equipes de Robótica, cada estudante pode focar em uma parte específica do código do projeto, alterando-a sem interferir nas demais partes porque cada um trabalha em aba separada.	Facilidade de colaboração
Ao se criar uma função no projeto, sua seção pode ser facilmente copiada para uma nova aba de qualquer outro projeto, possibilitando replicação ou mesmo outros usos e aplicações entre diferentes projetos.	Reutilização do código
Ao se separar o código em abas, podemos nos concentrar em aprender e entender uma parte do código de cada vez, auxiliando na depuração do projeto e no desenvolvimento da aprendizagem.	Foco no aprendizado
Conforme o projeto, dos mais simples aos mais complexos, as abas permitem modificações do código de forma independente e uma programação modular para projetos mais complexos, com cada função definida em suas abas.	Aplicação e usos

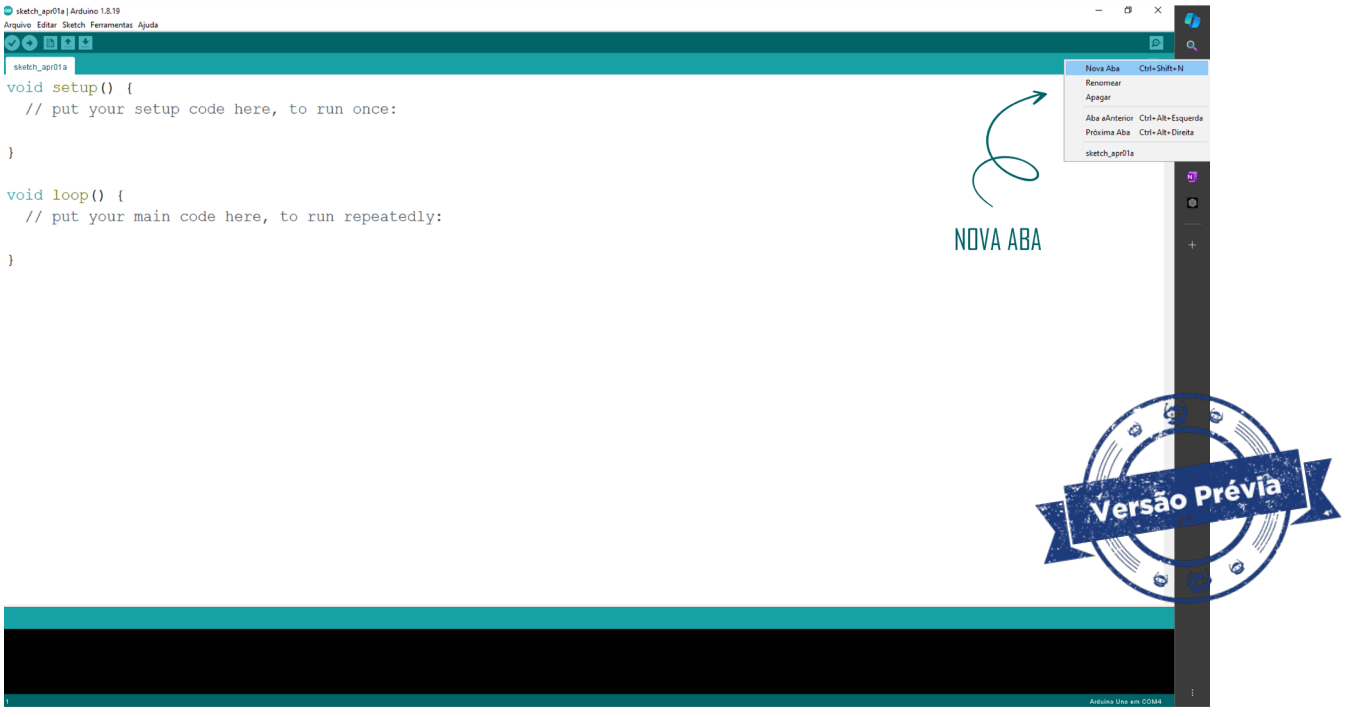
Você recorda, na **Aula 06 - Instax OLED**, em que trabalhamos com matrizes de bytes (arrays) para a impressão de fotografias e imagens no display? Pense em uma animação, com a exibição de várias matrizes. Se todas declaradas na mesma aba do sketch, o código ficaria super longo, não? Se cada matriz for declarada em abas individuais, no mesmo projeto como você pode conferir no sketch Instax OLED_B1N0 animado, o código principal fica mais “limpo”, chamando apenas a matriz presente em cada aba. Além disso, a técnica facilita também a inserção de outras matrizes sem riscos de bagunçarmos o código ou nos atrapalharmos com alguma linha.

O mesmo ocorre, por exemplo, com a programação de um robô rádio controlado, por exemplo. Podemos criar, no próprio sketch da programação, funções especificas de controle do robô, como **void frente(), void re()**, **void pare()**; **void esquerda()**, **void direita()**... A criação de cada uma destas funções pode ficar em abas específicas, otimizando também o código e nosso trabalho, caso queiramos fazer algum ajuste na função criada.

Como podemos adicionar, renomear e fechar abas para gerenciar nossos projetos de Robótica de forma eficiente?

A adição de abas é feita no próprio sketch da programação! Para criar uma aba no Arduino IDE Software, clique na seta localizada no canto superior direito da área de programação. Em seguida, no menu que aparece, selecione “Nova Aba” (New Tab). Digite um nome para a nova aba, seguido pela extensão do arquivo, e clique em OK. Para abas relacionadas ao preâmbulo da programação, insira o nome com a extensão **.h** e para abas com trechos do código, como funções, insira o nome com a extensão **.ino**.

Figura 02 - Criação de nova aba no Arduino IDE Software



Fonte: Software Arduino IDE

Importante!

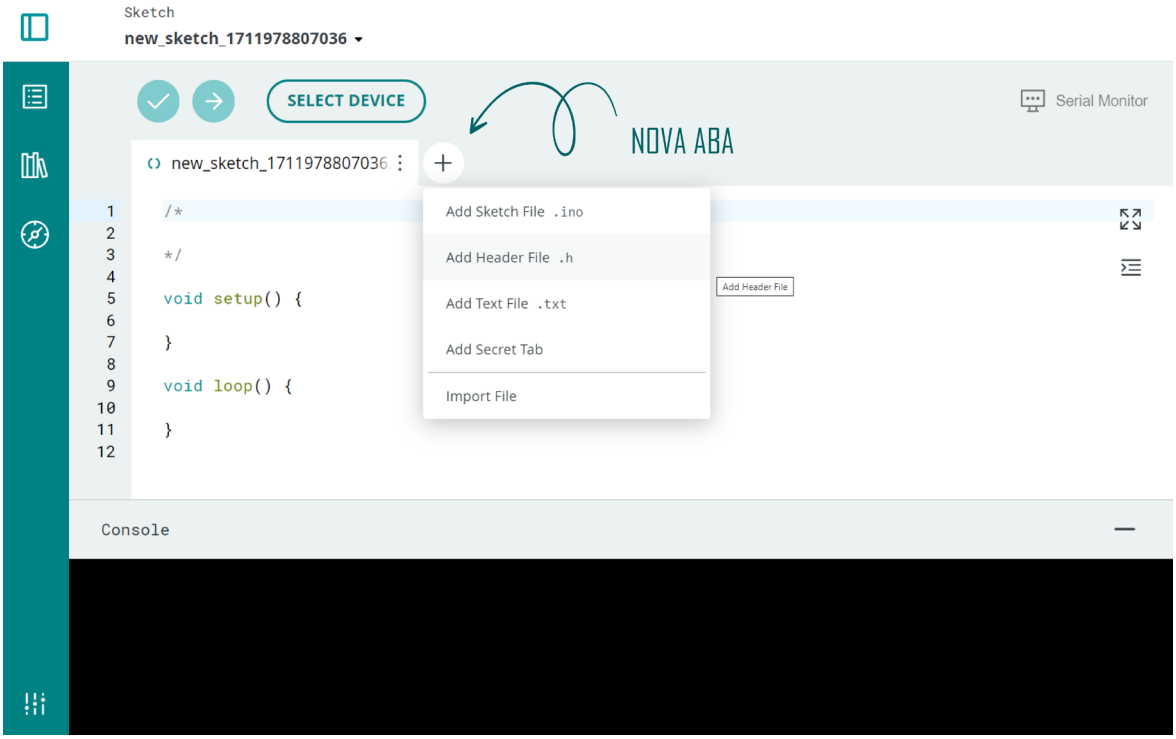
Use nomes consistentes e significativos, relacionados ao conteúdo ou função do código constante na aba, para evitar confusão ou erros de referência.

A nova aba será criada e você poderá começar a adicionar o código a ela. Lembre-se de salvar o seu projeto após adicionar novas abas para garantir que todas as mudanças sejam mantidas.

No caso da versão online do Arduino IDE, clique no sinal de “ **+** “, localizado ao lado da aba do sketch principal, e selecione o tipo de aba que deseja criar. Como vantagem, o Arduino IDE Online abre uma caixa de seleção para você indicar o tipo de aba desejada:

- *Sketch File .ino*, destinada às abas para funções;
- *Header File .h*, destinada às abas de cabeçalho;
- *Text File .txt*, destinada às abas de texto;
- *Secret tab*, destinada à criação de uma aba secreta, não visível no compartilhamento público do sketch.

Figura 03 - Criação de nova aba no Arduino IDE online



Fonte: Arduino IDE Online

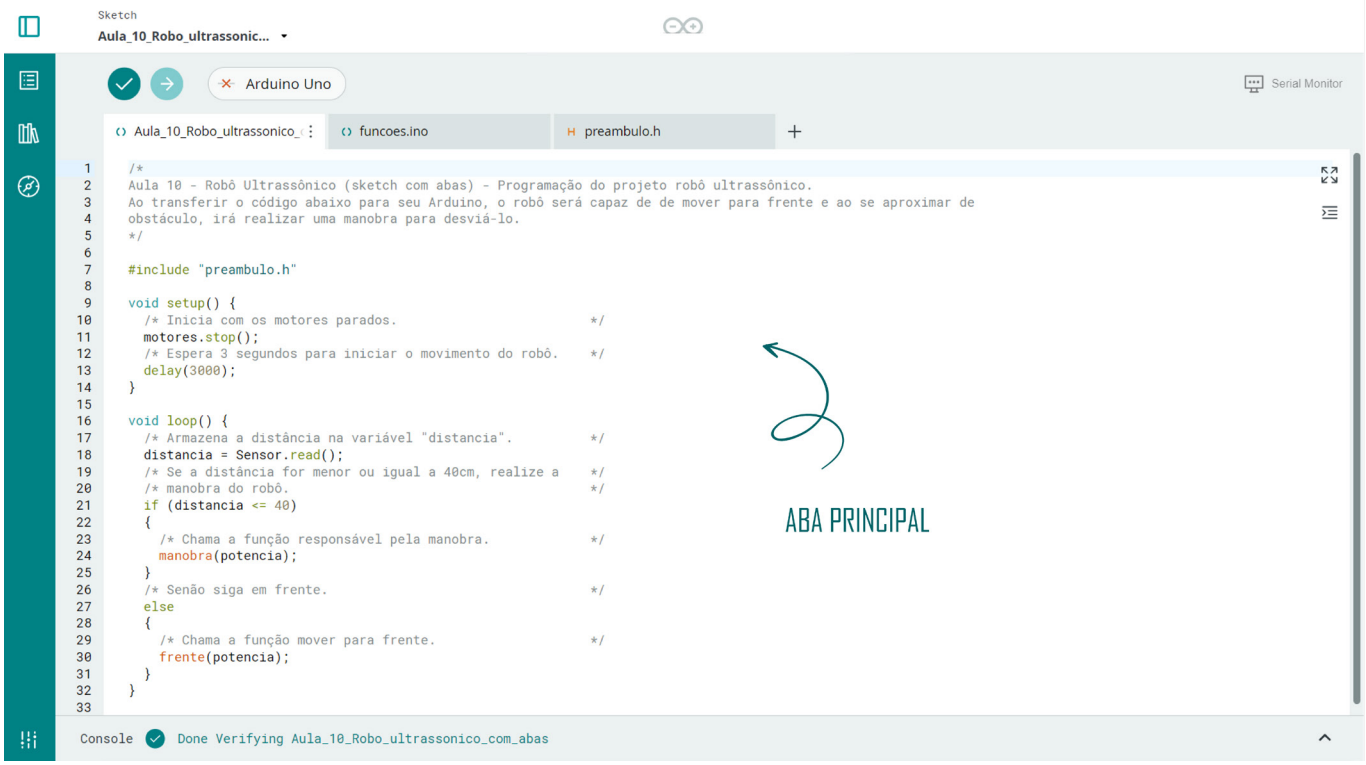
Como nos demais projetos contemplados durante nossa jornada pela Robótica, enfatizamos sempre a importância de comentar o código dentro de cada aba, explicando o que cada parte faz. Isso não só ajuda na compreensão e na manutenção do código por você ou seus colegas, mas também é uma boa prática de programação. Lembre-se de que cada aba representa um arquivo diferente dentro do seu projeto. Alternar entre estas abas permite que você trabalhe em diferentes partes do código de forma organizada e eficiente. E se o seu projeto estiver com muitas abas e elas não couberem na largura da janela do IDE, basta usar as setas de navegação que aparecem ao lado das abas para deslizar entre todas, acessando aquelas que estão fora de vista.

Além das dicas que vimos até agora, considere adotar outras boas práticas para a programação com abas:

- Relacionar os códigos, mantendo as funcionalidades específicas.
- Evitar abas muito extensas, dividindo o código em outras abas, se possível, para manter a legibilidade e otimizar a navegação.
- Testar o código individualmente, antes de integrá-lo ao código principal, para identificar e corrigir com mais precisão eventuais erros.
- Elaborar a documentação do projeto, criando um arquivo de texto, como um *read me* (leia-me), para explicar o projeto como um todo.

Vamos ver um exemplo de programação, disponível no Arduino IDE Online, otimizado por abas?

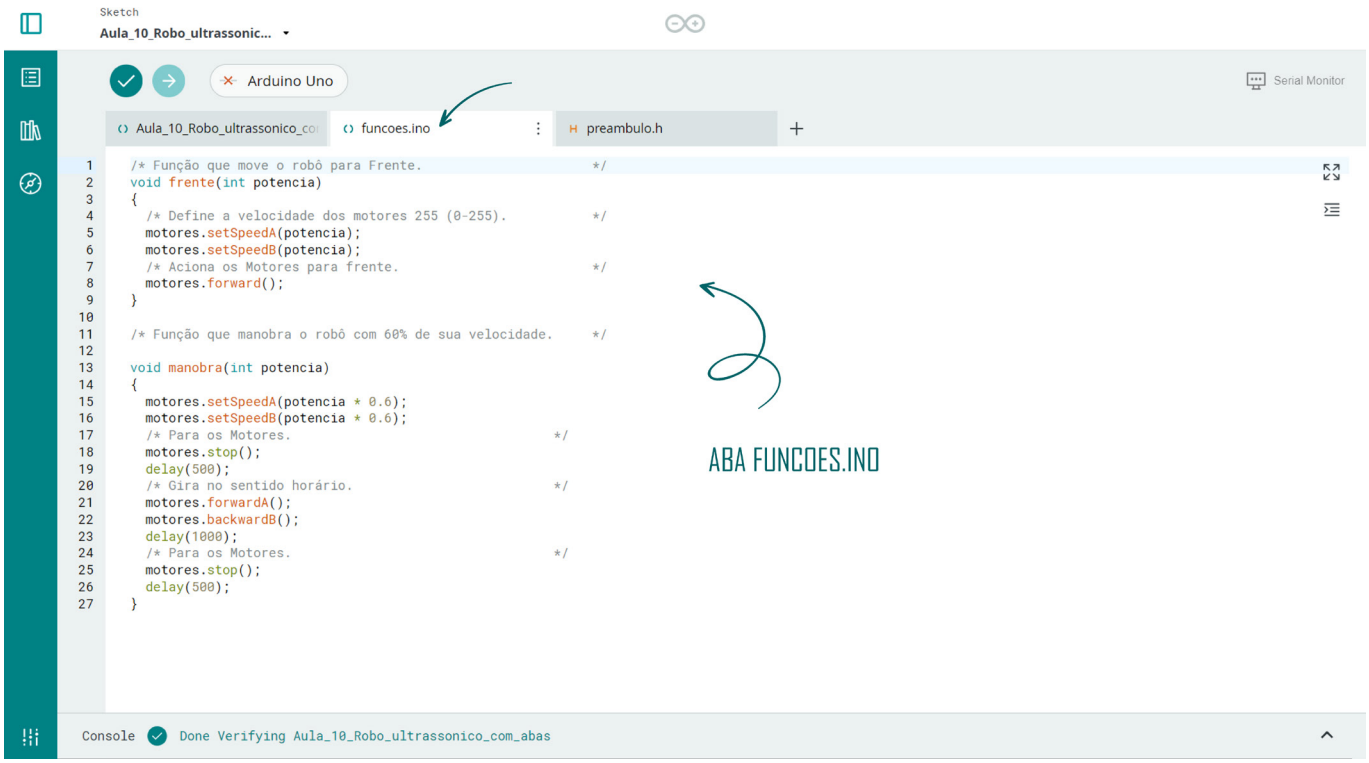
Figura 04 - Sketch com abas - Robô ultrassônico



Fonte: Arduino Online

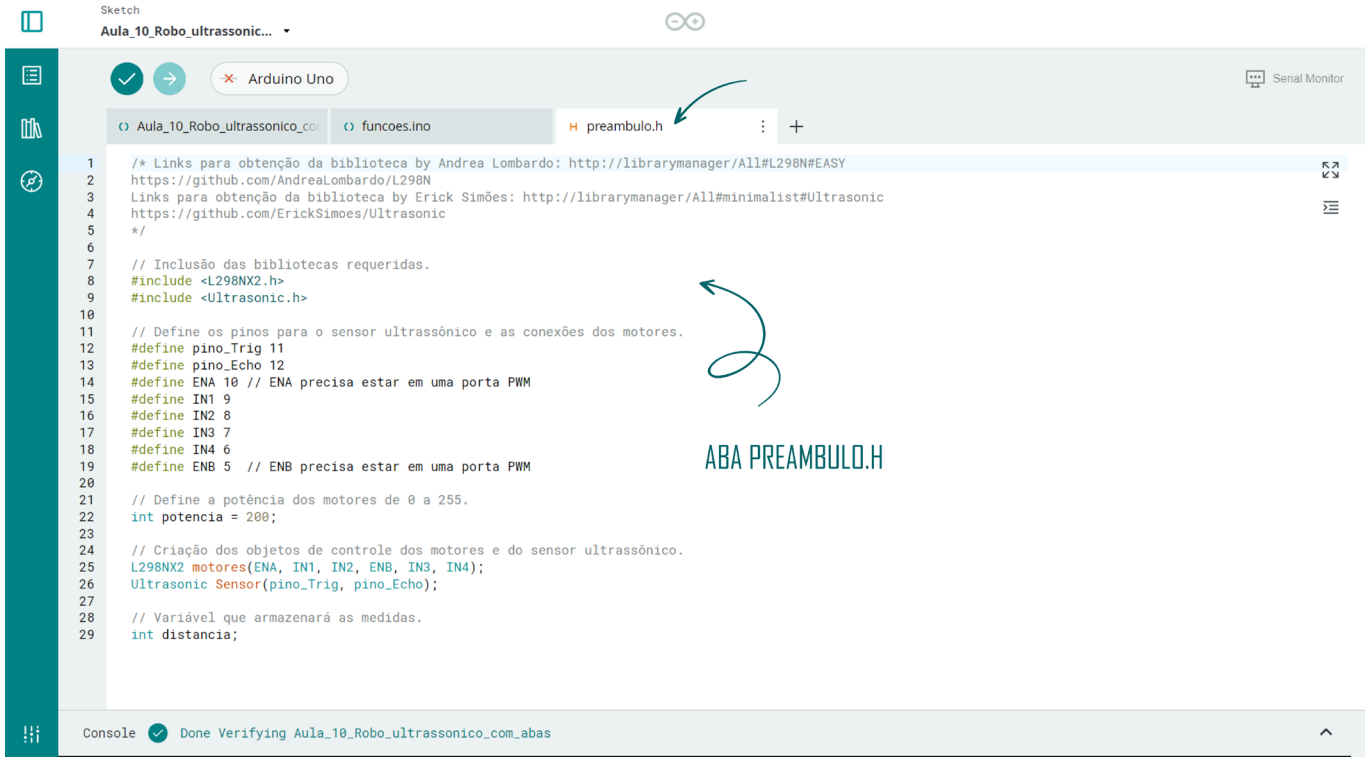
Nesta programação, todas as funções do robô estão concentradas em uma aba específica (funcoes.ino), com as definições para os componentes do projeto – motores e sensor ultrassônico – presentes também em aba própria (preambulo.h) Além desta forma de programar, é possível ampliar as seções com foco em uma aba **.ino** para cada função. Além disso, você pode criar também uma aba específica para deixar organizado todo o preâmbulo da programação, com as inclusões e definições necessárias em seu projeto.

Figura 05 - Aba funcoes.ino



Fonte: Arduino Online

Figura 06 - Aba preambulo.h



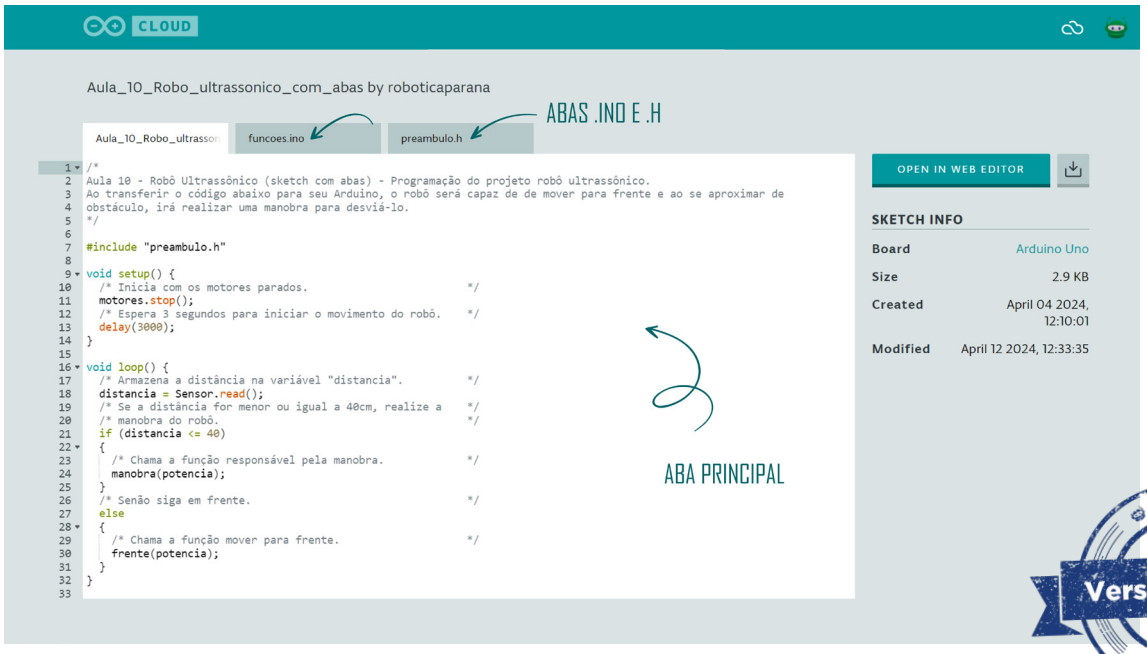
Fonte: Arduino Online

Abas para funções precisarão ser salvas com extensão **.ino**, a mesma do sketch, para que o Arduino olhe como uma sequência do olhar principal. Já a aba com o preâmbulo da nossa programação precisará ser salva com a extensão **.h** por se tratar do cabeçalho do sketch. Ao contrário das abas complementares **.ino**, que serão vistas pelo Arduino de modo direto, participando da compilação do projeto, a aba **.h** deve ser indicada no sketch da programação. Direcionamos este olhar no sketch principal com a inclusão da diretiva **#include** e a sintaxe específica para a aba **.h** que criamos difere da sintaxe para inclusão de bibliotecas.

Inclusão da aba **.h** presente no sketch da programação.	**#include "*nome_da_aba*.h"**
Inclusão de bibliotecas no sketch da programação.	**#include <*nome_da_biblioteca*.h>**

Após a criação de todas as abas, quando você salvar seu sketch no Arduino IDE versão online, as abas serão visualizadas de modo automático ao abrir novamente o projeto e, quando compartilhadas por link ou download, estarão presentes também!

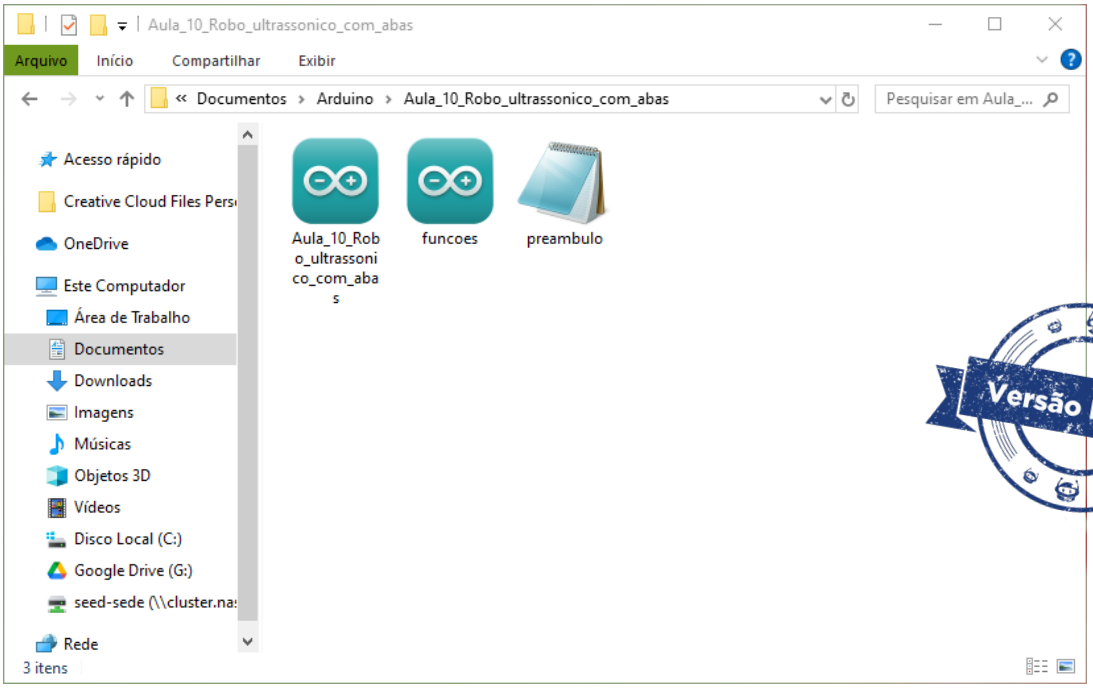
Figura 07 - Sketch compartilhado com abas



Fonte: Arduino Online

Caso você salve seu projeto no software Arduino IDE, observe a pasta do sketch, salva por padrão na pasta **Arduino**, que contará com o número de arquivos correspondente ao número de abas criadas para o projeto.

Figura 08 - Pasta de sketch com abas



Fonte: Windows

Caso você compartilhe seu projeto, precisará compartilhar a pasta completa para que o Arduino IDE mantenha a referência a todas as abas e não apresente erros de compilação.

A partir do exemplo que trouxemos de reorganização de sketch e das dicas contempladas no decorrer desta aula, que tal se aventurar e explorar novas formas de trabalhar com os códigos de programação do Arduino? Bons projetos!

Desafios:

Escolha um dos códigos estudados anteriormente ou um projeto pessoal que você desenvolveu e gostaria de aprimorar. Que tal revisitar este código e identificar áreas que possam ser mais organizadas ou separadas em partes diferentes?

Utilizando abas, reescreva o código para dividi-lo em seções lógicas e torná-lo mais limpo e legível.

E se...

O recurso das abas não funcionar? Observe se os nomes declarados para os arquivos e abas correspondem aos declarados no sketch da programação. Em abas adicionais, certifique-se de declarar todas as funções antes de usá-las no arquivo principal e evite definir a mesma função ou variável em múltiplas abas para não causar erros de compilação. Se você estiver usando arquivos **.h**, não se esqueça de incluí-los entre aspas no seu arquivo principal usando a diretiva #include.

Teste o código em cada aba separadamente antes de integrá-lo ao projeto principal para identificar e corrigir erros mais facilmente

Verifique o local em que os arquivos relacionados ao seu projeto estão salvos.

Feedback e finalização

A organização é primordial em todos os setores da nossa vida. Apesar de termos nossos momentos de desapego e ímpetos, manter ordem e bons hábitos acaba por nos trazer tranquilidade.

Este princípio contribui também para o desenvolvimento de projetos de Robótica, como vimos. Compartilhe com seus colegas uma análise sobre como a organização de um código de programação em abas pode torná-lo mais fácil para ser entendido e modificado no futuro.

Compartilhem experiências sobre a utilização de outros códigos e programações anteriores que vocês tenham realizado. Analisem como os bons hábitos, também aplicados à programação, ampliam a autonomia no desenvolvimento de projetos.

REFERÊNCIAS

ARDUINO. **Built-in Examples**. Disponível em: https://docs.arduino.cc/built-in-examples/. Acesso em: 18 mar de 2024.

ARDUINO. **Documentação de Referência da Linguagem do Arduino**. Disponível em: https://www.arduino.cc/reference/pt/. Acesso em: 18 mar de 2024.

ARDUINO DESDE CERO. **Incluir Arduino.h: Cómo utilizarlo en tus proyectos**. Disponível em: https://arduinodesdecero.com/cursos/incluir-arduino-h-como-utilizarlo-en-tus-proyectos/. Acesso em: 22 mar de 2024.

ELETROGATE. **Criando uma Biblioteca para Arduino**. Disponível em: https://blog.eletrogate.com/criando-uma-biblioteca-para-arduino/. Acesso em: 22 mar de 2024.

EMBARCADOS. **Criando suas próprias bibliotecas para Arduino**. Disponível em: https://embarcados.com.br/criando-bibliotecas-para-arduino/. Acesso em: 22 mar de 2024.

MEDIUM. **Como fazer uma biblioteca para Arduino?** Disponível em: https://medium.com/roboino/como-fazer-uma-biblioteca-para-arduino-d5da8bbb65d8. Acesso em: 22 mar de 2024.

DIRETORIA DE TECNOLOGIA E INOVAÇÃO EDUCACIONAL (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

Ailton Lopes

Andrea da Silva Castagini Padilha

Cleiton Rosa

Darice Alessandra Deckmann Zanardini

Edgar Cavalli Junior

Edna do Rocio Becker

José Feuser Meurer

Kellen Pricila dos Santos Cochinski

Marcelo Gasparin

Michele Serpe Fernandes

Michelle dos Santos

Roberto Carlos Rodrigues

Os materiais, aulas e projetos da “Robótica Paraná” foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica.

Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.