

TUTORIAL

# SUCURI STUDIO

Instalação, configuração inicial e primeiros passos

---

# APRESENTAÇÃO

## O que é este material

Este tutorial foi organizado para orientar a instalação do Sucuri Studio e apresentar as configurações básicas necessárias para começar a usar o software com segurança.

O Sucuri Studio é um ambiente de programação visual baseado em fluxogramas. Em vez de escrever linhas de código, o usuário constrói a lógica do programa com blocos, ligações e decisões visuais. Isso facilita o primeiro contato com programação, permite acompanhar o raciocínio passo a passo e ajuda a identificar erros com mais clareza.

Algumas etapas de conexão podem variar conforme a placa utilizada, a versão do sistema operacional e a versão do Sucuri Studio. Neste tutorial, as imagens da conexão física utilizam a Placa Smart, reconhecida no software pela interface Modelix 3.6. As orientações gerais de criação de fluxogramas, blocos e execução também podem ser utilizadas com Arduino e outras interfaces compatíveis.

## O que você aprenderá

- baixar e instalar o Sucuri Studio pelo site da Robokit;
- criar uma conta e fazer login;
- alterar o idioma do programa para Português (Brasil);
- entender as quatro opções da tela inicial;
- reconhecer as principais áreas da interface;
- usar os blocos básicos, os controles de execução e o Painel de Estados;
- organizar e salvar o primeiro fluxograma.
- escolher entre o simulador, uma placa Arduino, a Placa Smart/Modelix 3.6 ou outra interface compatível;
- identificar a porta de comunicação correspondente à placa conectada;
- compreender quais instruções são gerais e quais são específicas da Placa Smart/Modelix 3.6.

## Como instalar o Sucuri Studio

Antes de iniciar a programação, é necessário instalar o software utilizado neste material: o Sucuri Studio. Ele é o ambiente onde os fluxogramas são criados, simulados e, quando necessário, enviados para uma placa ou interface compatível, como Arduino ou Placa Smart/Modelix 3.6.

Siga o passo a passo abaixo para realizar a instalação:

### 1. Acesse o site da Robokit

Primeiro, entre no site da Robokit e, na página inicial, localize a barra de navegação superior. Em seguida, clique na aba “Softwares”.

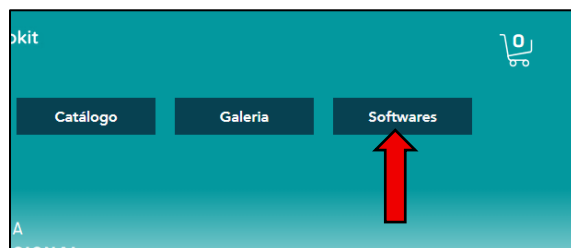


Imagem 2.1 — Botão “Softwares” na tela inicial da Robokit.

### 2. Baixando o Sucuri Studio

Dentro da página de softwares, role a tela para baixo até encontrar as opções de instalação disponíveis. Entre os softwares apresentados, localize **Sucuri Studio** e clique no botão **Acesse agora**.

Após clicar, o download do instalador será iniciado no seu computador.

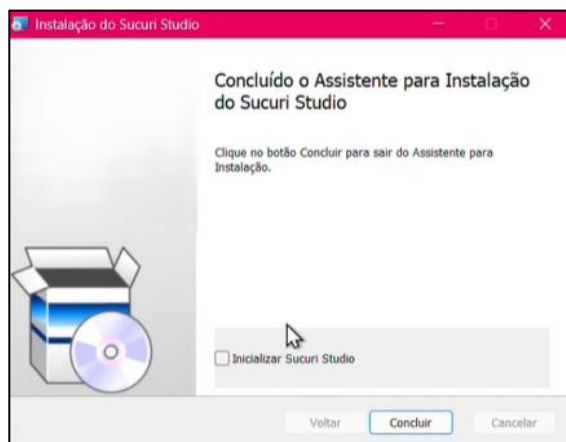
Caso o sistema operacional exiba uma notificação de segurança ou solicite uma confirmação, verifique se o instalador foi baixado diretamente do site oficial da Robokit. Depois dessa verificação, você poderá clicar em opções como “Sim”, “Permitir”, “Executar” ou “Executar assim mesmo” para continuar com a instalação.



Imagem 2.2 — Botão “Acesse agora” na aba Softwares.

### 3. Instalando o programa

Quando a janela de instalação abrir, siga as instruções exibidas na tela. Na maioria das etapas, basta clicar em Avançar, ler e aceitar os termos de licença quando forem apresentados e continuar até concluir a instalação. Ao final do processo, mantenha marcada a opção “Iniciar Sucuri Studio”, caso ela seja apresentada, e clique em “Concluir”.

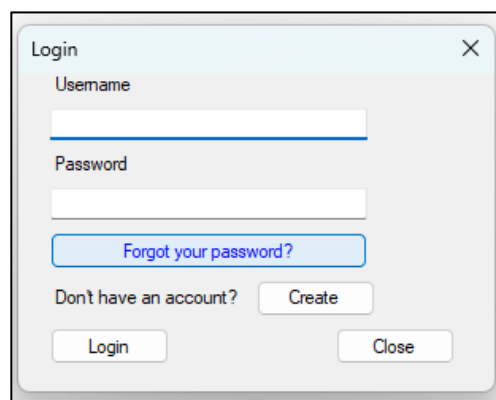


**Imagem 2.3** — Janela de instalação.

### 4. Criando uma conta

Ao iniciar o Sucuri Studio pela primeira vez, será exibida a tela de login. Na tela de cadastro, informe um endereço de e-mail válido, escolha um nome de usuário e crie uma senha. Depois de preencher os campos solicitados, clique novamente em “Create” para concluir a criação da conta.

Em seguida, volte à tela de login, informe o nome de usuário e a senha cadastrados e clique em “Login”. Guarde esses dados em um local seguro, pois eles serão utilizados nos próximos acessos.



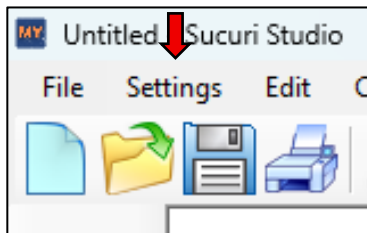
**Imagem 2.4** — Tela de login do Sucuri Studio.

Depois de criar sua conta e fazer login, você terá acesso ao ambiente principal do Sucuri Studio.

## Mudar o idioma do Sucuri Studio

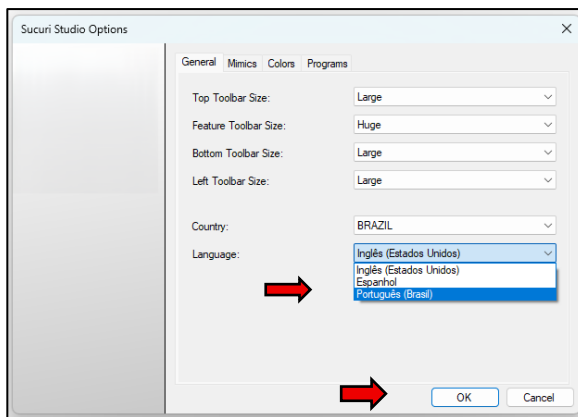
Depois de entrar na sua conta, o software pode estar em inglês. Para alterar o idioma para português, siga os passos abaixo.

Na tela principal do **Sucuri Studio**, observe o canto superior esquerdo do programa. Na barra de menus, localize a opção **Settings**, que fica entre **File** e **Edit**, e clique nela, depois clique em **Options**.



**Imagem 2.5** — Botão “Settings” na tela principal do Sucuri Studio.

Dentro da janela de configurações, acesse a aba **General**. Role até encontrar a opção **Language** e selecione **Português (Brasil)** e clique em **OK**.



**Imagem 2.6** — Aba Options dentro de Settings.

O idioma não será alterado imediatamente. Para aplicar a mudança, feche o Sucuri Studio e abra o programa novamente.

Com o Sucuri Studio instalado e configurado em português, o ambiente de programação está pronto para uso. A partir deste ponto, você poderá escolher entre utilizar o simulador, conectar uma placa Arduino, conectar a Placa Smart/Modelix 3.6, utilizar outra interface compatível ou abrir um fluxograma salvo anteriormente.

## Conectar uma placa física ao Sucuri Studio

As instruções a seguir utilizam a Placa Smart/Modelix 3.6 como exemplo. No caso de uma placa Arduino ou outra interface compatível, os nomes das opções e das portas poderão ser diferentes, mas o processo geral permanece semelhante: conectar a placa ao computador, selecionar a interface correta, escolher a porta de comunicação e confirmar a conexão.

Antes de começar qualquer programação, vale garantir um ponto básico: o Sucuri precisa reconhecer a Placa Smart corretamente. Quando essa conexão está funcionando, a placa aparece no Painel de Estados como Unidade de Controle, e isso muda completamente a aula: o professor consegue testar, depurar e ensinar com mais segurança.

Esse cuidado evita um dos problemas mais comuns em sala: o fluxograma estar certo, mas nada responder porque a conexão com a placa não foi feita da forma correta.

O procedimento mais seguro é simples: primeiro selecionar a porta certa, depois conectar, e só então testar.

Depois de abrir o Sucuri Studio e fazer login, a primeira tela já coloca você no “caminho certo” para a aula: escolha conectar na interface da placa.

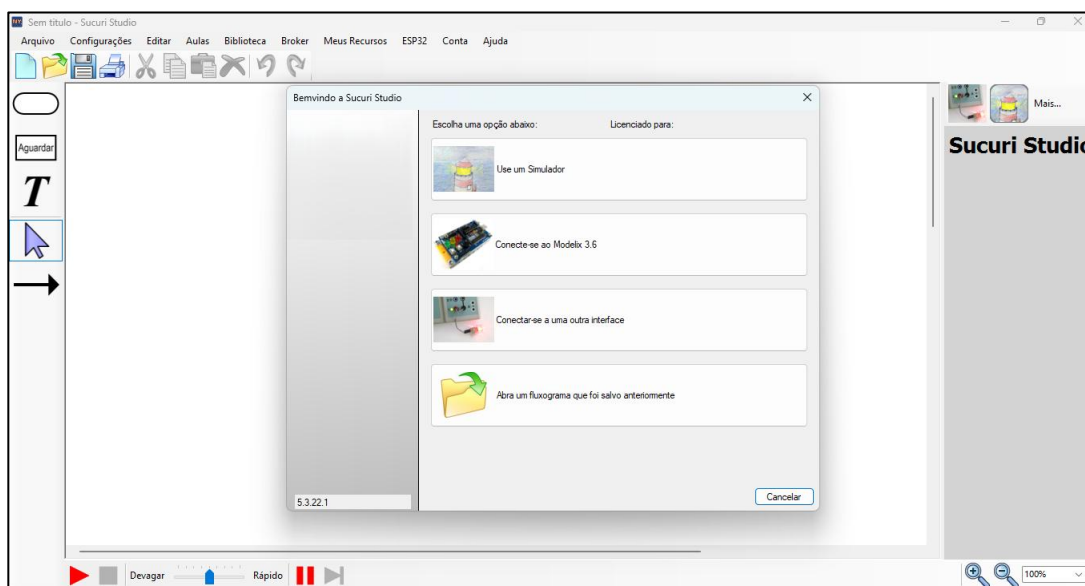
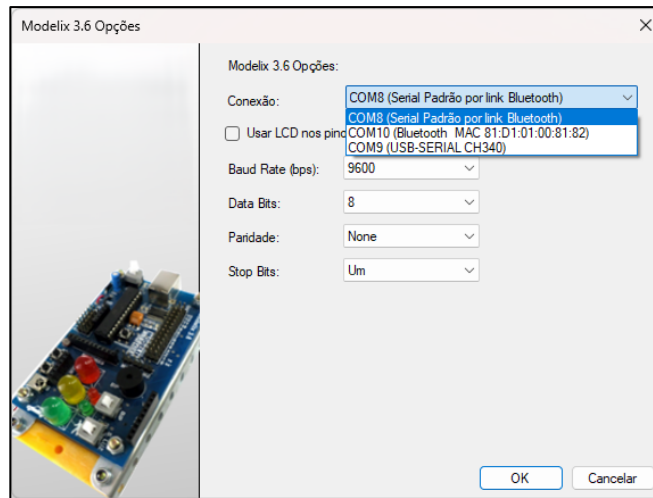


Imagem 2.7 — Tela inicial do Sucuri Studio.

Nessa tela, selecione **“Conecte-se ao Modelix 3.6”**. Evite **“Use um Simulador”** nesta etapa: para aula prática, o objetivo é a placa responder de verdade.

Ao escolher o Modelix 3.6, abre-se a janela de opções de conexão. É aqui que a maioria das aulas trava quando alguém escolhe a COM errada.



**Imagem 2.8 — Seleção da porta COM.**

Quando você conecta a Placa Smart ao computador pelo cabo USB, o sistema operacional cria uma porta de comunicação para ela. Essa porta aparece como COM (por exemplo, COM3, COM9 ou COM10).

Pense na porta COM como o canal por onde o Sucuri Studio conversa com a placa. Se a porta escolhida estiver errada, o software não consegue se comunicar corretamente com o dispositivo, e a aula trava logo no início.

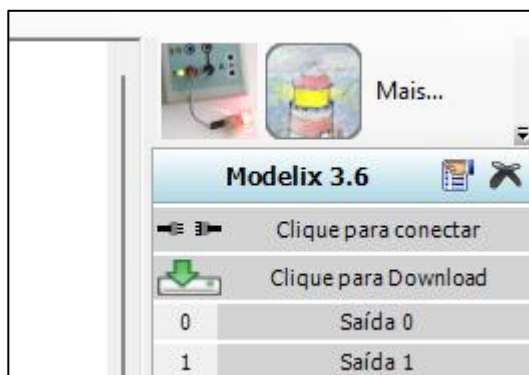
### O que você precisa saber (o essencial):

- COM é a porta de comunicação criada pelo sistema para dispositivos seriais, como placas e adaptadores USB.
- Quando a Placa Smart utilizada neste tutorial está conectada por USB, ela costuma aparecer identificada como USB-SERIAL (CH340). Em placas Arduino ou outras interfaces, a identificação poderá ser diferente, dependendo do modelo da placa e do driver instalado.
- Nesta aula, não será usado Bluetooth. Portanto, qualquer opção relacionada a Bluetooth deve ser ignorada.
- É normal o computador mostrar várias portas COM, porque ele pode ter outros dispositivos instalados ou portas virtuais criadas anteriormente. Por isso, nesta etapa, o mais importante não é o número da COM, mas a identificação do dispositivo ao lado.
- Para os Projetos 1 e 2, não é necessário alterar outras configurações de comunicação. Deixe os parâmetros padrão do programa e selecione apenas a porta correta antes de seguir.

Depois de selecionar a porta correspondente à placa conectada — USB-SERIAL (CH340), no equipamento utilizado neste tutorial — clique em “OK” e siga para realizar a conexão.

O número da porta COM, como COM3, COM8 ou COM10, pode variar de um computador para outro. Por isso, observe principalmente o nome do dispositivo exibido ao lado da porta.

Ao entrar na tela principal do Sucuri, você vai ver o Painel de Estados com o módulo Modelix 3.6 disponível. É exatamente ali que a conexão “vira real” — não basta escolher a COM certa; é preciso mandar o Sucuri conectar.



**Imagem 2.9** — Painel para conexão no Painel de Estados.

Nesse painel, clique em **“Clique para conectar”**. Esse comando inicia a comunicação entre o computador e a Placa Smart usando a porta COM selecionada.

Quando a conexão dá certo, o painel deixa de estar apenas estático e passa a indicar que a unidade está ativa e pronta para receber comandos. Esse é o momento em que a conexão realmente foi estabelecida.

Logo abaixo também aparece a opção **“Clique para Download”**. Aqui, “Download” significa enviar o fluxograma para a Placa Smart, ou seja, gravar nela o programa que foi montado no Sucuri.

Em termos práticos: Conectar abre o canal de comunicação; Download envia o programa para a placa.

Se nada acontecer ao clicar em Conectar, não avance para a programação. Primeiro revise o básico: cabo USB mal encaixado, porta COM errada ou outro programa usando a mesma porta.

### Painel de Estados: o professor “enxerga” o projeto funcionando

Depois de Conectar a placa, o Painel de Estados vira o seu “painel de instrumentos” da aula. É aqui que o professor confirma, sem adivinhação, se o problema está na montagem, na ligação de fios, no sensor, ou na lógica do fluxograma.

Em termos simples: o painel mostra entradas (sensores) e saídas (atuadores).

- **Entradas** são o que o projeto *recebe* do mundo (botão pressionado, luz no LDR, obstáculo no IR).
- **Saídas** são o que o projeto *faz* (acender LED, tocar buzzer, ligar motor).



O ponto mais importante para não travar a aula é entender que existem dois tipos de entrada e elas “aparecem” de formas diferentes:

|    |             |
|----|-------------|
| 11 | Saída 11    |
| 12 | Saída 12    |
| 13 | Saída 13    |
| 0  | Val 0 9,2 % |
| 1  | Val 1 9,4 % |

**Imagem 2.10** — Entrada digital e analógica no painel de estados.

- **Entrada digital (0/1):** é um *sim/não*, *ligado/desligado*.  
Ex.: botão pressionado = 1, solto = 0; sensor IR detectou = 1, não detectou = 0.
- **Entrada analógica (valor contínuo):** é um valor que *varia* (geralmente aparece como número ou percentual).  
Ex.: LDR muda conforme a luz: não é só “claro/escuro”; ele sobe e desce gradualmente.

Com essa base, o professor já tem o essencial para conduzir os Projetos 1 e 2 com mais segurança: selecionar a COM correta, conectar a placa e usar o Painel de Estados para verificar o que está acontecendo de verdade nas entradas e saídas. Isso reduz adivinhação, melhora a depuração e dá mais controle à aula.

## Como “ler” a tela do Sucuri (sem complicar)

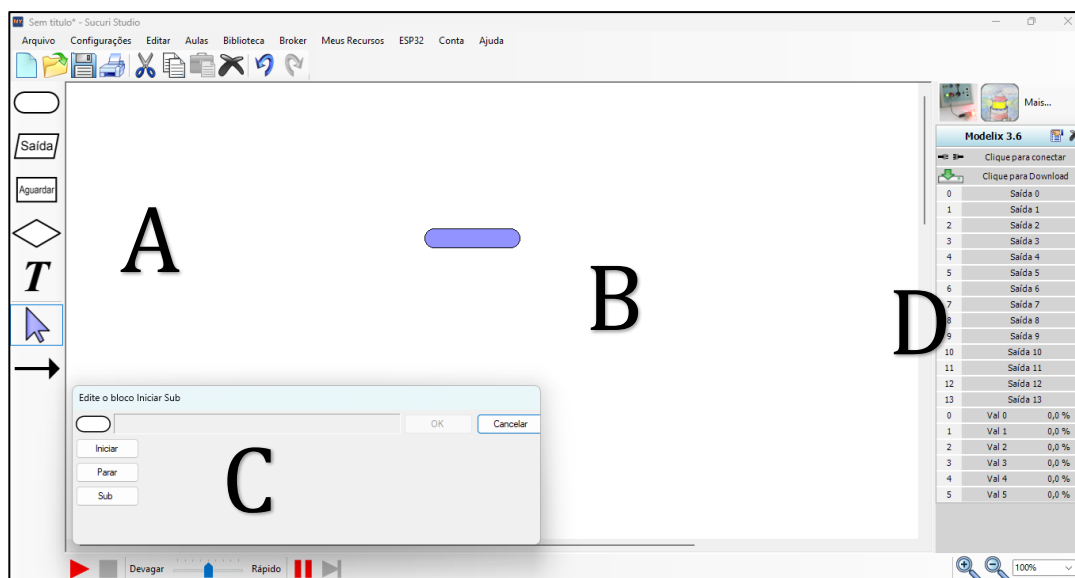


Imagem 2.11 — Visão geral do Sucuri Studio.

O Sucuri Studio funciona por fluxograma. Em vez de escrever linhas de código, o professor e a turma montam a lógica arrastando blocos para a área de trabalho e configurando cada ação na caixa de comandos. Isso torna a programação mais visual e facilita a leitura do raciocínio em sala.

Na prática, pense em 4 áreas (as únicas que você realmente precisa dominar):

1. **Barra de blocos (A):** de onde você pega os blocos principais.
2. **Área de trabalho (B):** onde o fluxograma “nasce”.
3. **Caixa de comandos (C):** onde você configura cada bloco e confirma.
4. **Painel de Estados (D):** onde aparecem entradas (sensores/botões) e saídas (LED, buzzer, motor), com seus valores/estados.

A organização geral da tela é semelhante no simulador e no uso com placas físicas. O que muda é o conteúdo apresentado no Painel de Estados e as opções de entrada e saída disponíveis para a interface selecionada.

Observação útil para aula: o Sucuri permite simular entradas — dá para clicar em entradas digitais e ajustar valores analógicos arrastando o mouse; isso ajuda a demonstrar lógica mesmo antes de plugar tudo.

## Configuração de portas e nomes

Antes de montar o fluxograma, vale ajustar duas coisas que deixam o trabalho mais limpo e evitam erros clássicos em sala: (1) o tipo de cada porta (entrada/saída, servo, motor etc.) e (2) o nome das entradas e saídas. Pense nisso como preparar a “tabela de instrumentos” do projeto: quando cada porta está configurada corretamente e com um nome claro, o aluno entende mais rápido o que está acontecendo e o professor depura com muito mais segurança.

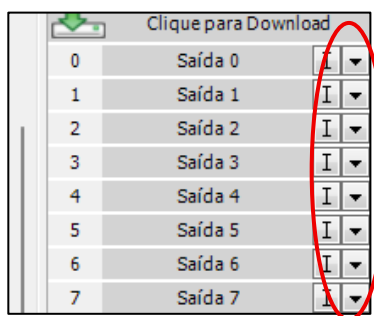


Imagem 2.12 — Botões para renomear a porta e alterar o tipo.

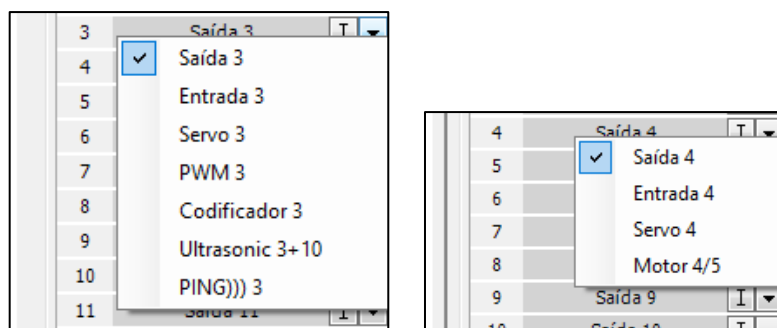


Imagem 2.13 — Menu “Tipo de porta”: escolhendo a função correta para o componente conectado.

No Sucuri, uma mesma porta pode assumir funções diferentes dependendo do que você conectou. Por isso, sempre que a turma trocar um componente (ou quando você estiver iniciando um projeto novo), revise a configuração:

- **Entrada (sensor):** usada quando a porta vai **ler** um estado/valor (ex.: botão, sensor de luz, sensor IR).
- **Saída (atuador):** usada quando a porta vai **comandar** algo (ex.: LED, buzzer).

- **Motor:** usada quando a porta está destinada a **motor DC**. No Sucuri, o motor é tratado como um recurso único, mas tecnicamente ele utiliza duas saídas trabalhando em conjunto para controlar o sentido de giro. Por isso, para usar o motor M1, configure a porta 4 como **Motor 4/5 (M1)**. Se houver um segundo motor, configure a porta 7 como **Motor 6/7 (M2)**. Assim, o software passa a apresentar cada motor de forma mais simples e mais adequada para a aula.
- **Servo:** usada quando você quer controlar um **servo motor**, com posição/ângulo (quando o kit/placa suportar esse uso).
- **Outros modos (quando disponíveis):** o menu pode trazer configurações específicas para determinados componentes. Esses modos ajustam automaticamente o comportamento da porta e liberam os blocos mais adequados no Sucuri. Exemplos comuns são PWM, Codificador e Ultrasonic 3+10. A regra prática é simples: selecione sempre o modo compatível com o componente conectado.

|   |         |   |   |
|---|---------|---|---|
| 3 | Saída 3 | I | ▼ |
| 4 | Saída 4 | I | ▼ |
| 5 | Saída 5 | I | ▼ |

|   |               |   |   |
|---|---------------|---|---|
| 3 | Saída 3       | I | ▼ |
| 4 | LED (lâmpada) | I | ▼ |
| 5 | Saída 5       | I | ▼ |

**Imagem 2.14** — Renomeando a “Saída 4” para “LED (lâmpada)”.

**Renomear** é um recurso simples e muito profissional. Em vez de ficar olhando “A1”, “A2”, “D3”, “Saída 2”, você transforma as portas em algo que faz sentido no projeto.

Exemplos para os Projetos 1 e 2:

- **Entradas:** “Botão”, “LDR (Luz)”, “IR”, “Fim de curso” (se houver).
- **Saídas:** “LED (Lâmpada)”, “Buzzer (Alarme)”, “Motor (Ventilador)”, “Motor Grua”.

O impacto disso é enorme: o aluno passa a ler o fluxograma como uma história (“SE LDR < limiar ENTÃO liga LED (Lâmpada)”), e o professor identifica rapidamente onde está o problema (“o IR não está mudando”, “a saída Buzzer não acionou”).

## Os blocos essenciais (o “kit mínimo” de programação)

Para os Projetos 1 (Casa Inteligente) e 2 (Grua), o professor não precisa “decorar o software inteiro”. Na prática, quase toda a lógica desses dois projetos nasce de cinco blocos-base, que se repetem como uma receita: começar, testar uma condição, acionar uma resposta, controlar o tempo e encerrar quando necessário. Quando a turma entende bem esse conjunto, o resto vira detalhe — e você consegue ensinar programação com clareza, sem transformar a aula em exploração confusa de menus.

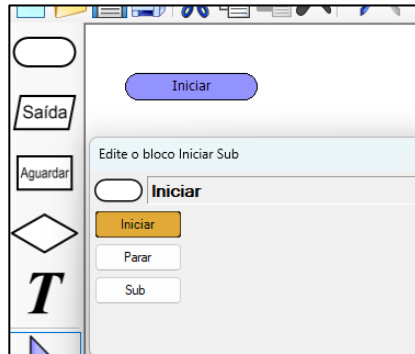


Imagem 2.15 — Iniciar e “rotinas em paralelo”.

O bloco **Iniciar** marca o começo do programa. No Projeto 1, é possível usar múltiplos blocos Iniciar para separar rotinas que funcionam em paralelo, o que ajuda a organizar a Casa Inteligente sem misturar todas as regras no mesmo fluxo. Didaticamente, porém, vale testar uma rotina por vez antes de juntar tudo no funcionamento completo.

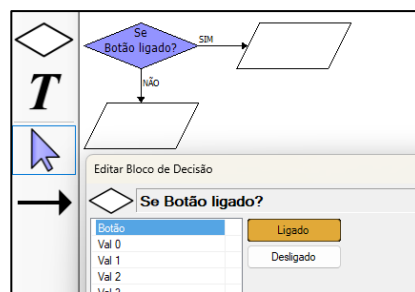
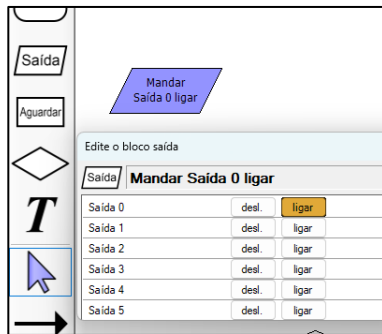


Imagem 2.16 — Decisão (SE / SENÃO).

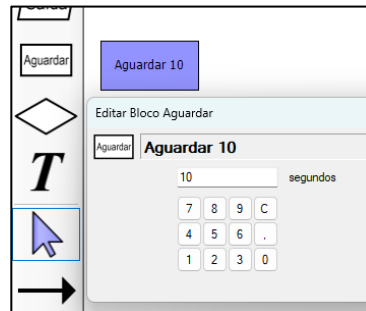
O bloco **Decisão** é o coração da automação: ele testa uma condição (chave/sensor/valor) e segue por SIM ou NÃO.

Regra de ouro: toda Decisão precisa ter os dois caminhos (SIM e NÃO) ligados — isso evita “pontas soltas” no fluxo.



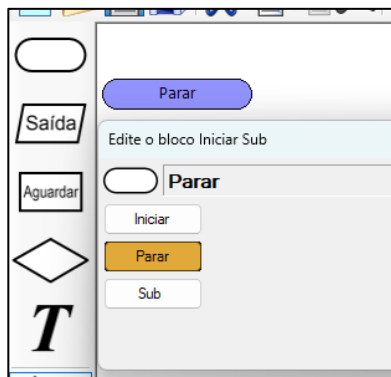
**Imagem 2.17 — Saída (atuadores).**

O bloco **Saída** liga/desliga uma saída digital ou manda o motor girar em um sentido ou outro (conforme a interface/controle). É o bloco que “faz acontecer” (LED, buzzer, motor).



**Imagem 2.18 — Aguardar (tempo).**

O bloco **Aguardar** pausa o fluxo por um tempo definido. É simples, mas muito útil para evitar comandos “rápidos demais” na demonstração e para dar previsibilidade ao teste em sala.



**Imagem 2.19 — Parar.**

Use **Parar** para finalizar um fluxo quando fizer sentido (em geral, nos nossos dois projetos, a lógica principal roda continuamente; então o Parar aparece mais como recurso de organização/controle).

**Para este livro:** não é necessário entrar em variáveis e sub-rotinas como conteúdo principal. O próprio Sucuri “esconde” recursos avançados para manter a barra limpa e só mostra o que é relevante no momento.

### Executar e ensinar: controle de ritmo (para a turma “ver” a lógica)

O professor, no Sucuri, não deve apenas “apertar Executar e torcer para funcionar”. O mais valioso em sala é transformar a execução em demonstração: a turma precisa ver o caminho do raciocínio (onde o fluxograma está, qual decisão foi tomada e qual saída foi acionada). Por isso, os controles de execução não são “detalhe do software” — eles são uma ferramenta didática para ensinar lógica com segurança.

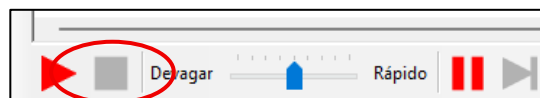


Imagem 2.20 — Botões Executar e Parar.

O básico é simples: **Executar** inicia o programa e **Parar** interrompe a execução quando você quiser reiniciar um teste ou corrigir algo.

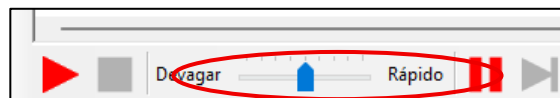


Imagem 2.21 — Controle Devagar/Rápido.

O controle de velocidade muda totalmente a experiência de aprendizagem. Em Devagar, os alunos conseguem acompanhar “o caminho do raciocínio” com clareza, observando cada decisão do programa. Na velocidade do meio, a execução fica no ritmo normal do sistema — é o padrão mais próximo do que se espera durante a maior parte da aula, equilibrando resposta e leitura do que está acontecendo.

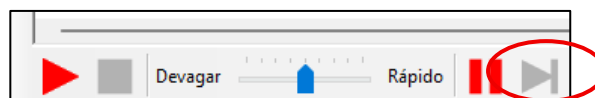


Imagem 2.22 — Botões Pausar e Próximo Bloco.

Esse é o modo perfeito para uma micro demonstração passo a passo. Você pausa o programa e avança um bloco por vez, narrando o que está acontecendo. Isso ajuda o aluno a perceber que o fluxograma não é “mágica”: ele segue uma sequência lógica.

Com esses controles, a aula vira um laboratório de lógica: você testa uma parte, observa o que o programa decidiu, ajusta e só depois acelera para validar o funcionamento completo.

## Preparando o primeiro fluxograma

O objetivo desta etapa é organizar um primeiro fluxo simples no simulador. O comportamento exato das entradas e saídas depende do cenário escolhido, mas a sequência de trabalho é a mesma.

1. Faça login e escolha “Use um Simulador”.
2. Selecione um cenário simples para abrir a área de trabalho.
3. Arraste um bloco inicial para a área de trabalho e configure-o como “Iniciar”.
4. Adicione um bloco “Saída” e escolha uma ação disponível no simulador.
5. Adicione um bloco “Aguardar” e defina um tempo curto para criar uma pausa perceptível.
6. Use a seta de ligação para conectar os blocos na ordem correta.
7. Clique em cada bloco e confira suas opções na caixa de comandos. Confirme as alterações em “OK”.
8. Salve o arquivo antes de executar.
9. Clique em Executar e mova a velocidade para “Devagar”. Observe o caminho do programa.
10. Use Pausar e Próximo Bloco para analisar a lógica passo a passo.
11. Acompanhe o Painel de Estados e confirme se a saída reage conforme o comando.
12. Clique em Parar, faça os ajustes necessários e execute novamente.

Esse primeiro exemplo utiliza o simulador e não exige uma placa conectada. Para utilizar o mesmo princípio com uma placa física, escolha a interface correspondente na tela inicial, configure as entradas e saídas e confirme a conexão antes de executar ou enviar o programa.

**Regra prática:** monte, conecte, configure, salve, execute devagar e acompanhe o Painel de Estados. Essa ordem torna os testes mais claros e evita alterações aleatórias.

## Abrir um projeto depois

Na próxima vez que iniciar o software, escolha “Abra um fluxograma que foi salvo anteriormente” ou use o menu “Arquivo > Abrir”. Localize o arquivo e continue o trabalho a partir da última versão salva.



## FECHAMENTO – Checklist de primeiro uso

Antes de iniciar uma atividade no Sucuri Studio, faça esta conferência rápida:

- ☐ O programa foi instalado a partir do site oficial da Robokit.
- ☐ A conta foi criada e o login foi realizado.
- ☐ O idioma está configurado como Português (Brasil).
- ☐ O software foi fechado e aberto novamente depois da mudança de idioma.
- ☐ A opção inicial escolhida corresponde ao objetivo da atividade.
- ☐ O arquivo foi salvo em uma pasta fácil de localizar.
- ☐ Todos os blocos estão ligados na ordem correta.
- ☐ Cada bloco foi configurado e confirmado em “OK”.
- ☐ A execução foi iniciada em velocidade lenta para facilitar a observação.
- ☐ O Painel de Estados foi usado para acompanhar entradas e saídas.
- ☐ O programa foi interrompido antes de fazer correções no fluxograma.
- ☐ Caso uma placa física seja utilizada, ela está conectada ao computador pelo cabo USB.
- ☐ A porta de comunicação correta foi escolhida.
- ☐ A placa aparece como conectada ou ativa no Painel de Estados.
- ☐ As configurações específicas da interface foram verificadas antes da execução.

### Soluções rápidas para os problemas mais comuns

| Situação                      | O que verificar                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------|
| O idioma não mudou            | Feche completamente o Sucuri Studio e abra-o novamente.                                             |
| O fluxo não inicia            | Confirme se existe um bloco Iniciar, se os blocos estão ligados e se o botão Executar foi acionado. |
| A execução está rápida demais | Mova o controle de velocidade para Devagar ou use Pausar e Próximo Bloco.                           |
| Nada muda no simulador        | Verifique a configuração do bloco de saída e acompanhe o Painel de Estados durante a execução.      |
| O projeto não foi encontrado  | Use Arquivo > Abrir e procure na pasta em que o fluxograma foi salvo.                               |

Com essas etapas concluídas, o Sucuri Studio estará instalado, configurado e pronto para a criação dos primeiros fluxogramas. O usuário poderá iniciar pelo simulador e, posteriormente, aplicar a mesma lógica em uma placa Arduino, na Placa Smart/Modelix 3.6 ou em outra interface compatível, observando sempre as configurações específicas de cada equipamento.