



**UNIVERSIDADE SALGADO DE OLIVEIRA**  
**TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ÉVENY JANIELY OLIVEIRA DE MELO**

**PROTOTIPAGEM COM ARDUINO**

Relatório de aula e pesquisa

RECIFE

2025

ÉVENY JANIELY OLIVEIRA DE MELO

**PROTOTIPAGEM COM ARDUINO**

Relatório de aula e pesquisa

Relatório técnico-científico  
apresentado como requisito para  
obtenção de nota na disciplina de  
Laboratório de Computação Aplicada  
ministrada pelo professor:

RECIFE  
2025

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>4</b>
<b>1.1 Objetivos Gerais e Específicos .....</b>	<b>4</b>
1.1.1 Objetivos gerais .....	4
1.1.2 Objetivos específicos .....	4
<b>2 DESENVOLVIMENTO .....</b>	<b>5</b>
<b>2.1 A plataforma Arduino e a prototipagem .....</b>	<b>5</b>
2.1.1 Importância e usos .....	5
<b>2.2 Registro técnico de aula .....</b>	<b>6</b>
2.2.1 Sensor LM35 .....	7
2.2.2 Explicação sobre a IDE, usos e teste de código .....	9
<b>3 CONCLUSÃO .....</b>	<b>13</b>
<b>REFERÊNCIAS .....</b>	<b>14</b>

## 1 INTRODUÇÃO

Criada na Itália em 2005, a famosa plataforma de prototipagem denominada Arduino tem se tornado cada vez mais popular globalmente, principalmente em ambientes acadêmicos. Sua arquitetura, constituída tanto de hardware quanto de software, possibilita aos usuários uma base sólida para desenvolver inúmeras ideias e soluções inovadoras de forma simples e veloz.

O presente relatório técnico-científico engloba informações sobre a plataforma de prototipagem Arduino originadas de pesquisas e da aula ministrada sobre tal, visando explorar os fundamentos teóricos e práticos do Arduino, explorando seu conceito, arquitetura, e o processo de prototipagem.

### 1.1 Objetivos Gerais e Específicos

#### 1.1.1 Objetivos gerais

Registrar e descrever de forma clara as aplicações de prototipagem com o Arduino e o conhecimento teórico e prático adquirido sobre a plataforma, estruturando um documento que demonstre sua funcionalidade e relevância na criação de soluções tecnológicas.

#### 1.1.2 Objetivos específicos

- Contextualizar a importância da plataforma no cenário tecnológico atual e na prototipagem.
- Detalhar seus componentes fundamentais e usos.
- Explorar e documentar o conhecimento adquirido sobre o Arduino em sala de aula e em pesquisas.
- Registrar usos experimentais da IDE do Arduino, feitos a partir de prévia orientação do professor.

## 2 DESENVOLVIMENTO

Nesta seção será apresentado o detalhamento dos fundamentos teóricos e a dissertação sobre a aplicação prática da plataforma Arduino, conforme citado nos objetivos gerais e específicos do relatório. Inicialmente, será abordado o que é e a importância do Arduino junto com seus principais usos no cenário acadêmico e tecnológico. Logo em seguida, serão exploradas informações sobre a plataforma, originadas na aula ministrada pelo professor e em pesquisas pessoais, finalizando com a documentação sobre o uso do ambiente de desenvolvimento e o teste de código passado em sala.

### 2.1 A plataforma Arduino e a prototipagem

No ano de 2005, em terras italianas, um grupo de pesquisadores criou o Arduino, uma plataforma de prototipagem eletrônica, com o objetivo inicial de auxiliar estudantes sem prévio conhecimento aprofundado em programação e eletrônica, no desenvolvimento de protótipos. Este dispositivo permitiu que qualquer pessoa pudesse explorar o desenvolvimento de projetos por meio de um preço acessível.

O Arduino é amplamente reconhecido por sua natureza open source, que possibilita que os usuários e desenvolvedores compartilhem soluções e contribuam para o aprimoramento da plataforma. Sua arquitetura o permite interagir com qualquer ambiente por meio de hardware e software, podendo ser conectado a um computador para transmitir e receber dados.

Funcionalmente, o Arduino é um dispositivo de fácil utilização e dotado de várias portas, digitais e analógicas, essenciais para conexões com módulos e sensores. Essa característica o torna um dispositivo fundamental na criação de projetos interativos de complexidade variada, que envolvem a leitura de dados ambientais simples até a automação residencial e a robótica avançada. Resumidamente, a vantagem do Arduino vem do poder que ele tem de simplificar a ponte entre a leitura de dados físicos e o controle.

#### 2.1.1 Importância e usos

O Arduino é amplamente utilizado em diversos setores por sua flexibilidade e facilidade de uso. Ele se tornou a escolha ideal para facilitar o aprendizado e a experimentação em áreas como a robótica, IoT (internet das coisas), automação residencial e

em projetos educacionais e acadêmicos. No entanto, seu uso principal e foco deste documento reside no seu uso para prototipagem.

A prototipagem configura-se como um processo essencial no desenvolvimento de produtos e sistemas, consistindo na criação de versões iniciais (protótipos) para testar e validar conceitos, funcionalidades e o design antes de investir na produção em larga escala. Neste contexto, o Arduino destaca-se como uma ferramenta poderosa para a prototipagem, oferecendo baixo custo e alta velocidade na validação de ideias.

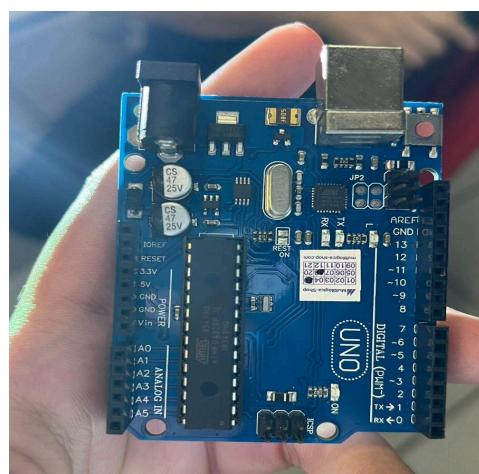
Além disso, sua importância no cenário educacional e acadêmico é crescente. Ele proporciona uma abordagem prática e envolvente para o ensino de eletrônica, programação e engenharia, ao ser utilizado em salas de aulas para desde ensinar conceitos básicos de eletrônica até criação de projetos, protótipos e realização de experimentos.

## 2.2 Registro técnico de aula: o Arduino e o LM35

Com o conceito e a importância do Arduino tendo sido estabelecidos, esta seção concentra-se no registro detalhado das informações e instruções fornecidas em sala de aula, essenciais para a compreensão do dispositivo.

Conforme a aula ministrada, a plataforma Arduino é um dispositivo monocore, caracterizado por seu microcontrolador, que possui uma capacidade de processamento de aproximadamente 20 MIPS (milhões de instruções por segundo). Em sala, a família de hardware exposta foi a Arduino Uno R3.

**Figura 1 - Registro da peça estudada em aula.**



Fonte: desconhecida.

É fundamental, contudo, diferenciar o microcontrolador do microprocessador:

- Microprocessador: unidade de processamento compacta, tipicamente integrada em uma CPU, que é responsável por executar funções aritméticas e lógicas em sinais digitais.
- Já o microcontrolador, é a unidade básica de computação em dispositivos eletrônicos inteligentes, como máquinas de lavar. Ele se diferencia por ser um computador muito pequeno, com seus próprios sistemas de memória e entrada e saída incorporados em um único chip. É aqui que entra o microcontrolador do Arduino Uno R3, que sendo o componente principal de sua estrutura, é o responsável por gerenciar múltiplas conexões através de suas várias portas analógicas e digitais.

O sucesso da prototipagem com o Arduino se deve a sua arquitetura que permite a interação direta com o mundo físico: se baseia na utilização de sensores para receber dados do ambiente. Para gerenciar essa interação, a programação é realizada em uma linguagem baseada em C/C++, que garante o controle de baixo nível necessário sobre o hardware. Foi citado em aula, por exemplo, o sensor de temperatura LM35, que é um dispositivo capaz de traduzir graus Celsius para milivolts, enviando o dado analógico para o Arduino processar.

Outra questão importante é saber diferenciar os circuitos analógicos e digitais:

- Circuitos Analógicos: caracterizados pelos componentes e as ligações entre eles (trilhas). (ex.: fontes/carregadores são analógicos em sua maioria)
- Circuitos Digitais: Todo circuito lógico que também seja integrado por instruções/software. (ex.: o Arduino). Possui no mínimo uma memória, um processador e uma unidade de armazenamento.

Como ponto de comparação sobre capacidade de processamento, foi citado que o Raspberry Pi é um exemplo de dispositivo que possui uma capacidade de processamento superior à do Arduino.

### 2.2.1 Sensor LM35

A série de sensores de temperatura LM35 é composta por circuitos integrados de

precisão desenvolvidos para medir a temperatura de forma simples, sendo perfeito para conectar em microcontroladores (como o do Arduino) ou outros circuitos de controle. Esse pequeno dispositivo tem como principal característica fornecer uma saída diretamente em graus Celsius.

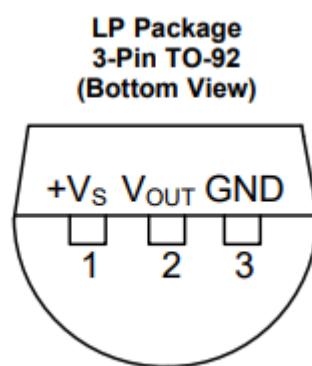
Essa característica oferece a vantagem de, diferente dos sensores calibrados em Kelvin, eliminar a necessidade do usuário precisar subtrair uma grande tensão constante para converter a leitura para a escala de Celsius, tornando o recebimento de dados mais eficiente.

Além disso, o LM35 é ideal para ser interligado a circuitos de leitura ou controle, por possuir:

1. Uma saída linear, na qual a temperatura e tensão são lineares, simplificando a programação e o processamento de sinal.
2. Um sinal de tensão gerado estável e menos suscetível a ruídos ou perdas ao ser transmitido para o circuito de leitura.
3. Um sensor já calibrado desde a produção, o que dispensa ajustes externos e garante alta precisão desde o primeiro uso.

Tais características listadas permitem que o dispositivo opere tanto com fontes de alimentação simples (ex.: Fontes USB), quanto com fontes positivas e negativas (ex.: Amplificadores operacionais), conferindo grande flexibilidade ao projeto.

**Figura 2 - Encapsulamento da série LM35 exposto em sala.**



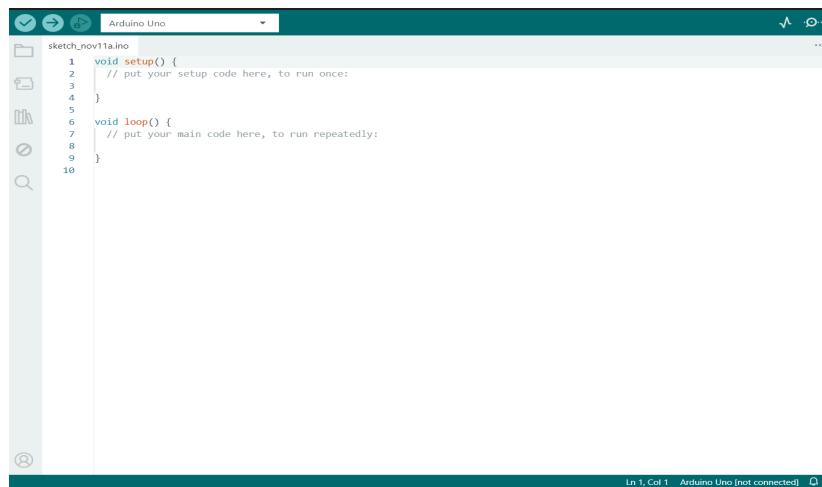
Fonte: ALLDATASHEET. .LM35 Datasheet (PDF) - Texas Instruments. Disponível em:

<<https://www.alldatasheet.com/datasheet-pdf/pdf/517588/TI1/LM35.html>>. Acesso em: 11 nov. 2025.

## 2.2.2 Explicação sobre a IDE, usos e teste de código

Com a arquitetura do hardware bem definida, esta subseção se concentra na documentação do teste da IDE do Arduino, seguindo instruções passadas pelo professor. O procedimento inicial envolve o download e a configuração da IDE e identificação do hardware, sendo necessário primeiramente selecionar o modelo Arduino Uno na plataforma para garantir o processo de compilação e comunicação.

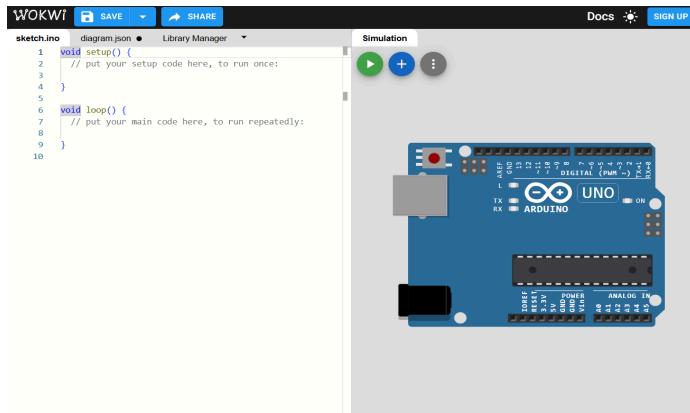
**Figura 3 - IDE da plataforma Arduino**



Fonte: autor.

A partir da instalação e configuração do programa o próximo passo seria conectar a placa ao computador. O Arduino, por sua natureza de comunicação, requer que o sistema operacional atribua a ele uma porta serial (Port) para que o código compilado possa ser enviado. No entanto, pela falta de um Arduino físico em mãos, foi preferido utilizar um simulador online Wokwi. Essa ferramenta cumpre a função essencial de testar o código e conectar componentes virtuais, permitindo a validação do projeto sem precisar de hardware físico.

**Figura 4 - Ambiente Virtual Wokwi.**



Fonte: autor.

Em seguida, para validação lógica, foi analisado e adicionado o código de leitura do sensor de temperatura LM35, estruturado na sintaxe C/C++, no ambiente virtual Wokwi.

**Figura 5 - Código passado em aula para leitura e conversão do sensor LM35**

```

#define pin_lm35 A0 //Definição de qual pino está conectado ao sensor

void setup() {

Serial.begin(9600); // estabelece o canal de comunicação.

pinMode(pin_lm35, INPUT); // Configura o pino para receber dados do sensor.

}

void loop() {

float valor_analog_lm35 = float(analogRead(pin_lm35)); // Captura o sinal
analógico do sensor que varia de 0 a 1023.

float tensao = (valor_analog_lm35 * 5) / 1023; // Traduz o valor bruto para
para tensão elétrica (volts).

float temperatura = tensao / 0.010; // dividimos a tensão por 0.010V, que
representa os 10 mV (fator de escala do sensor), obtendo a temperatura em
Celsius.

Serial.print("A temperatura é:");

```

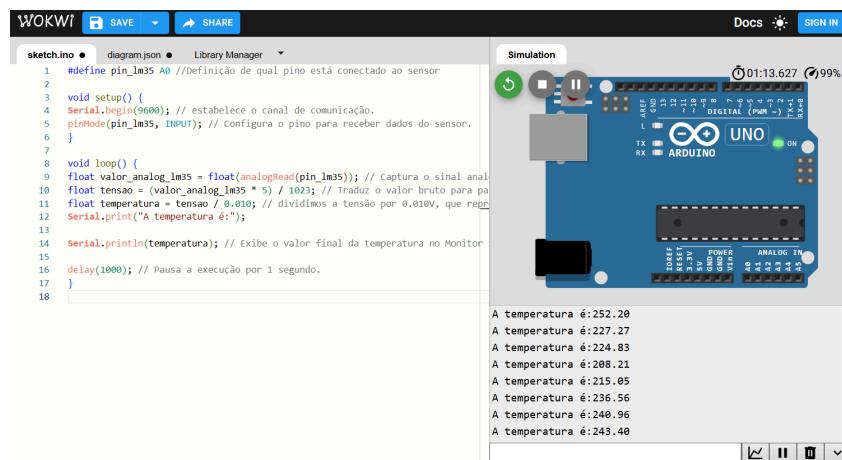
```
Serial.println(temperatura); // Exibe o valor final da temperatura no Monitor Serial.
```

```
delay(1000); // Pausa a execução por 1 segundo.
```

```
}
```

Fonte: Sidney Cunha.

**Figura 6 - Interface do Wokwi com a saída de dados do sensor LM35 no Monitor Serial.**



Fonte: autor.

O código analisado possui como principal objetivo traduzir o sinal elétrico do sensor LM35 em graus Celsius, demonstrando a capacidade do Arduino de realizar a conversão e processamento matemático. Inicialmente o código utiliza `#define` para definir qual pino está conectado ao sensor (o do Arduino) e logo após isso, na parte `void setup()` (a configuração) é estabelecido o canal de comunicação entre o arduino e o Monitor Serial (computador) e o pino é configurado para receber dados do sensor.

Na próxima parte do código, o `void loop()`, é onde ocorre todo o processamento:

1. O Arduino lê o valor do sinal analógico que vem do sensor.
2. O código traduz esse valor de energia, primeiro para volts e depois aplica a fórmula matemática para encontrar a temperatura em graus Celsius.
3. Por fim, o código envia o resultado (“A temperatura é:”) e o valor calculado para ser

exibido na tela do computador.

O comando `delay(1000)` faz com que o sistema espere exatamente 1 segundos para repetir o ciclo, garantindo que a medição seja contínua. Sendo assim, o resultado final do processamento do código é o Arduino lendo o ambiente e mostrando o dado traduzido para o usuário.

Finalizando a documentação técnica do código passado, é fundamental registrar o conceito de três elementos essenciais na programação e comunicação do Arduino:

- Função `void setup()`: Representa o bloco onde ocorre a configuração e a parametrização do que se quer que funcione. Sua principal função é preparar o ambiente, sendo executado apenas uma vez ao ligar ou reiniciar a placa.
- Função `void loop()`: Sendo executado após o `void setup()`, essa função contém a parte de processamento de um código, a qual é executada infinitamente. É útil para gerir tarefas, monitorar sensores e comunicar com outros periféricos.
- Função `baud rate()`: Também conhecida como taxa de comunicação ou transmissão, determina a velocidade (em bits por segundo) em que um dado é transmitido ou recebido. Sua configuração precisa é vital para garantir que a comunicação seja sincronizada e que a leitura dos dados ocorra sem erros.

### 3 CONCLUSÃO

Em um mundo onde a prototipagem pode ser uma maneira crucial para validar um projeto, este relatório cumpriu o objetivo principal ao registrar os dados obtidos em aula e descrever de forma clara as aplicações de prototipagem com o Arduino, bem como o conhecimento teórico e prático adquirido sobre a plataforma.

O registro das informações apresentadas e a execução dos testes de código no simulador Wokwi foram fundamentais para entender o fluxo de trabalho e comprovar a funcionalidade do código de leitura do sensor LM35. Esta abordagem, que combina teoria com prática, solidificou a compreensão sobre a interface entre hardware e software e destacou a importância da fase de prototipagem digital para a antecipação de falhas, a redução de erros e otimização do tempo no ciclo de desenvolvimento.

Conclui-se, portanto, que o Arduino constitui uma base sólida e indispensável para o desenvolvimento de futuros projetos, sendo uma ferramenta estratégica para inovação e desenvolvimento em áreas como a Internet das Coisas (IoT), robótica e a automação de sistemas.

## REFERÊNCIAS

**ARDUINO. Software.** Arduino. Acesso em: 10 nov. 2025. Disponível em:  
<https://www.arduino.cc/en/software>.

**AWS. Qual é a diferença entre microprocessadores e microcontroladores?** Amazon Web Services. Acesso em: 9 nov. 2025. Disponível em:  
<https://aws.amazon.com/pt/compare/the-difference-between-microprocessors-microcontrollers>.

**CIRCUITÁR. Setup e Loop: Entendendo a Estrutura Básica de um Sketch.** Circuitár. Acesso em: 11 nov. 2025. Disponível em:  
<https://www.circuitar.com.br/projetos/setup-loop/index.html>.

**DATASHEET. LM35.** Texas Instruments. Acesso em: 11 nov. 2025. Disponível em:  
<https://www.alldatasheet.com/datasheet-pdf/pdf/517588/TI1/LM35.html>.

**KALATEC. Arduino: o que é?** Blog Kalatec. Acesso em: 9 nov. 2025. Disponível em:  
<https://blog.kalatec.com.br/arduino-o-que-e>.

**MAKER HERO. O que é Arduino?** Blog Maker Hero. Acesso em: 9 nov. 2025. Disponível em:  
[https://www.makerhero.com/blog/o-que-e-arduino/?srsltid=AfmBOor9Y4IyjiGQRCAKvIyaETiNc0cALTd0B7GVEmgNxo\\_1yCHdQgoT](https://www.makerhero.com/blog/o-que-e-arduino/?srsltid=AfmBOor9Y4IyjiGQRCAKvIyaETiNc0cALTd0B7GVEmgNxo_1yCHdQgoT).

**MAKER HERO. Wokwi: Simulador de Arduino para Testar seus Códigos.** Blog Maker Hero. Acesso em: 10 nov. 2025. Disponível em:  
[https://www.makerhero.com/blog/wokwi-simulador-de-arduino/?srsltid=AfmBOooCR0F8iZ\\_XK3JGZ](https://www.makerhero.com/blog/wokwi-simulador-de-arduino/?srsltid=AfmBOooCR0F8iZ_XK3JGZ).

**VICTOR VISION. Baud Rate: o que é e como funciona essa taxa de comunicação serial.** Blog Victor Vision. Acesso em: 11 nov. 2025. Disponível em:  
<https://victorvision.com.br/blog/baud-rate>.

VICTOR VISION. **O que é Arduino.** Blog Victor Vision. Acesso em: 9 nov. 2025. Disponível em: <https://victorvision.com.br/blog/o-que-e-arduino>.

WOKWI. **New Project: Arduino Uno.** Wokwi. Acesso em: 10 nov. 2025. Disponível em: <https://wokwi.com/projects/new/arduino-uno>.

XAVIER, S. **Linguagem Arduino: void loop.** Arduino Factory, 2022. Acesso em: 11 nov. 2025. Disponível em: <https://arduinofactory.fr/pt/linguagem-arduino-void-loop>.