

RELATÓRIO SEMANAL - AULA 02 - PORTAS LÓGICAS

Disciplina: Sistemas em Tempo Real e
Embarcados

Nome do Estudante: João Pedro Issmael Vieira

E-mail Autor / Estudante:
joao.pedrov@sempreceub.com

Matrícula / RA: 22252263

Curso: Ciências da Computação

Brasília – DF



Msc. Wendell Cruzeiro

wendell.cruzeiro@ceub.edu.br

Data: 25/02/2026

Resumo: A atividade se baseia em construir as portas lógicas de AND, OR e NOT de forma física, digital e elétrica. Utilizamos o Falstad para analisar o comportamento do fluxo da corrente, no Tinkercad montamos o protótipo do circuito, simulamos a lógica por um código em python e, por fim, montamos o circuito usando um Arduino e um Breadboard.

Palavras Chaves: Arduino, Circuitos Digitais, Portas Lógicas, Tinkercad, Falstad, Eletrônica Prática

Abstract: The activity is based on building AND, OR, and NOT logic gates in physical, digital, and electrical forms. We used Falstad to analyze the behavior of the current flow, in Tinkercad we assembled the circuit prototype and, finally, we assembled the circuit using an Arduino and a Breadboard.

Keywords: Arduino, Digital Circuits, Logic Gates, Tinkercad, Falstad, Practical Electronics.

1 INTRODUÇÃO

A eletrônica digital se baseia em sinais binários (0 e 1) e exploramos como esses sinais binários se comportam por meio das portas lógicas **AND**, **OR** e **NOT**. A proposta da atividade visa justamente realizar a teoria elétrica pura(análise de fluxo no Falstad), depois fazer a simulação dos componentes (prototipagem visual no Tinkercad), a lógica de programação (simulação em Python) físicos e depois realizar a montagem física (Arduino e Breadboard).

As portas lógicas que foram estudadas foram:

- **Porta AND:** a saída só é ativada se todas as entradas tiverem valor Verdadeiro.
- **Porta OR:** a saída é ativa se pelo menos uma das entradas tiver valor verdadeiro.
- **Porta NOT:** atua como um inversor, o que é originalmente Verdadeiro vira Falso e vice-versa.

2 METODOLOGIA

2.1 SIMULAÇÃO ELÉTRICA

Nessa etapa é feito o diagrama esquemático no Falstad, permitindo visualizar o sentido da corrente. O circuito de cada porta lógica foi montada utilizando componentes passivos como switches, sources de energia com duplo terminal, resistores e LED, para garantir que a montagem da lógica booleana estava correta.

2.2 PROTOTIPAGEM DIGITAL

No Tinkercad, a técnica utilizada foi a simulação de hardware realista, diferente do Falstad, aqui foi preciso lidar com a fiação e posicionamento realista dos componentes. A fiação foi organizada no Breadboard disponibilizado, conectado a um Arduino com o lado positivo em 5V e o negativo no GND, e o resistor foi conectado de forma que o LED estivesse protegido.

2.3 DESENVOLVIMENTO EM PYTHON

No Google Colab foram escritos scripts que simulavam os comportamentos das portas. O usuário dava a entrada 1 ou 0 via teclado e o programa processava esses dados simulando as portas lógicas respectivas exibindo o resultado em binário no console

2.4 MONTAGEM DE HARDWARE

Execução física do projeto utilizando o Arduíno para fornecer energia para o sistema no ProtoBoard.

3 MATERIAIS

Softwares: Navegador para Falstad/Tinkercad e Interpretador Python no Google Colab.

Arduíno: Utilizado como fonte de alimentação de 5V e referência de terra (GND).

Breadboard (Protoboard): Placa de ensaio para a conexão dos componentes.

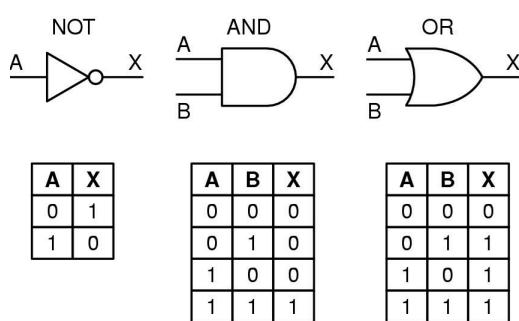
Botões (Push-buttons): Utilizados para simular as entradas binárias (pressionado = 1, solto = 0).

Fios Macho-Macho (Jumpers): Essenciais para interconectar o Arduíno e os botões na breadboard.

LED e Resistor: O LED que serviu como indicador visual da saída, enquanto o resistor que evitou o LED de queimar.

4 RESULTADOS E DISCUSSÃO

Os resultados demonstraram que a simulação das portas lógicas funcionaram conforme o previsto pelas suas respectivas tabela verdade:



Na porta AND o LED de saída só foi ativado quando ambas as entradas estavam com valor Verdadeiro. Na porta OR o LED ligou com qualquer uma das entradas em valor Verdadeiro. Na porta NOT, o LED já estava ligado, porém ao acionar o botão causamos um curto circuito, desligando o LED e efetivamente invertendo seu valor.

5 CONCLUSÕES

O trabalho foi concluído com sucesso, atingindo o objetivo de demonstrar o funcionamento das portas lógicas **AND**, **OR** e **NOT** em diferentes camadas: elétrica, digital e física. Essa abordagem garantiu uma compreensão profunda de como a lógica booleana opera em diversos contextos.

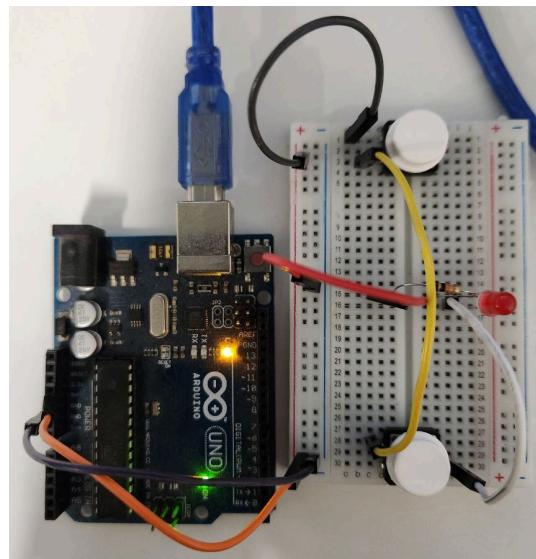
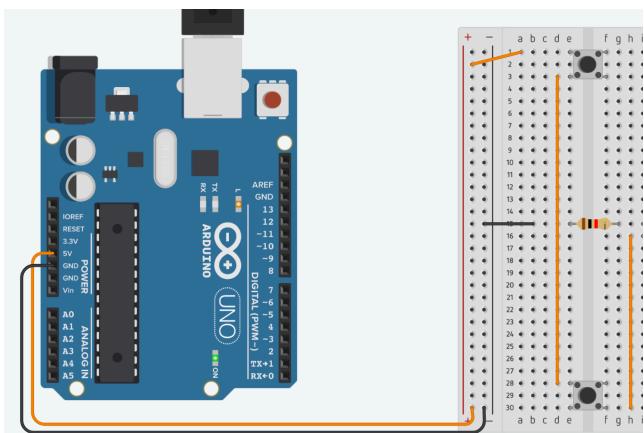
Os principais desafios foram o medo de danificar os componentes físicos por erro de conexão e a complexidade de organizar a lógica e o posicionamento de cada peça no Tinkercad. No entanto, a prática permitiu superar essas dificuldades, validando que a teoria e a prática caminham juntas no desenvolvimento de circuitos digitais.

6 REFERÊNCIAS

- DEITEL, H. M.; DEITEL, P. J.; CHOHNES, D. R. Sistemas operacionais. 3. ed. São Paulo: Pearson, 2005.
- DENARDIN, G. W.; BARRIQUELLO, C. H. Sistemas operacionais de tempo real e sua aplicação em sistemas embarcados. São Paulo: Blucher, 2019.
- ESCOTTÁ, A. T.; BECCARO, W. Controle automático de volume em tempo real utilizando inferência fuzzy em um sistema embarcado. Trends in Computational and Applied Mathematics, v. 22, p. 41–60, 2021. Disponível em: <https://www.scielo.br/j/tcam/a/wjqHkwWq>
- W KW46gsCghRtgSg/ SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. Fundamentos de sistemas operacionais. 9. ed. Rio de Janeiro: LTC, 2015.

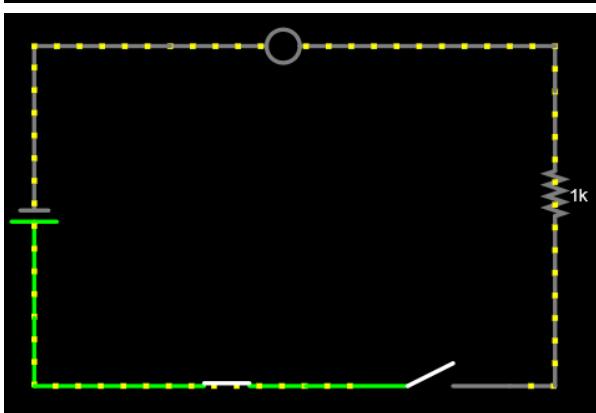
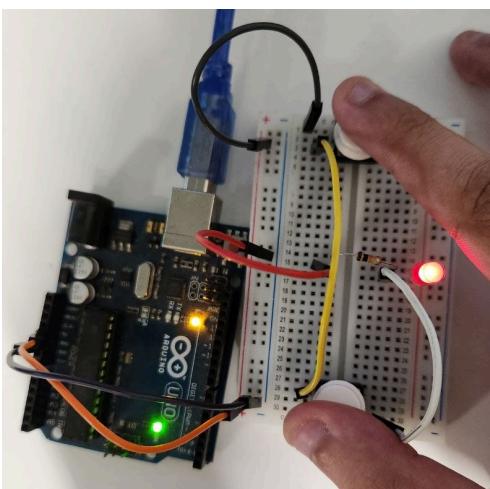
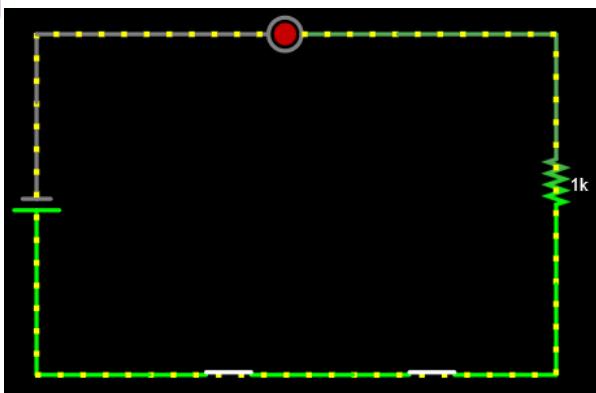
APÊNDICE

PORTA AND

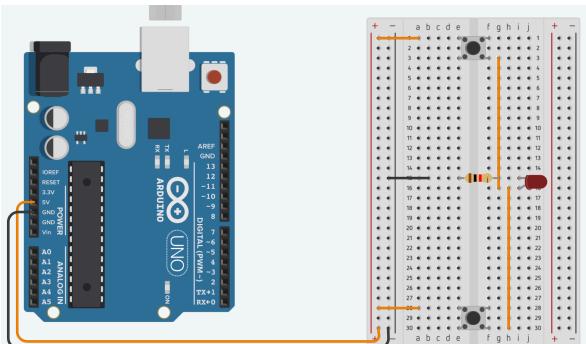


```
num1 = int(input("Escolha entre 0 e 1"))
num2 = int(input("Escolha entre 0 e 1"))

if num1 == 0 and num2 == 0:
    print("0")
if num1 == 0 and num2 == 1:
    print("0")
if num1 == 1 and num2 == 0:
    print("0")
if num1 == 1 and num2 == 1:
    print("1")
```

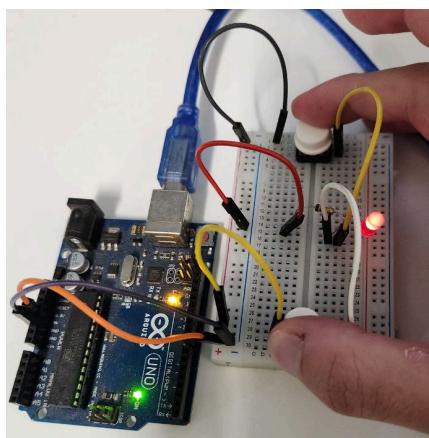
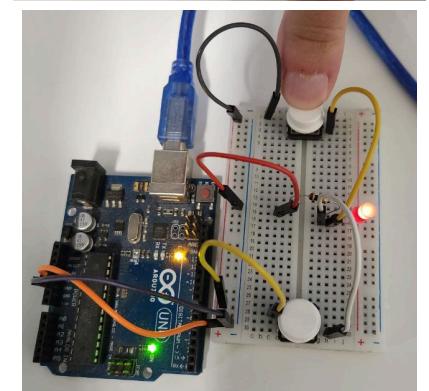
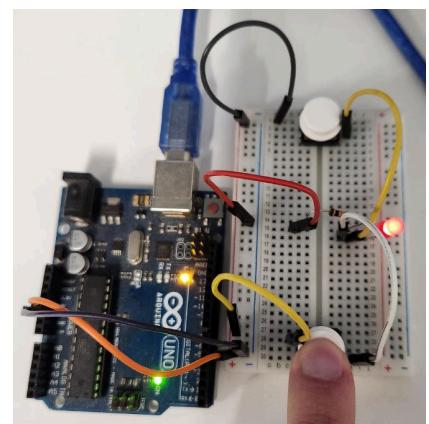
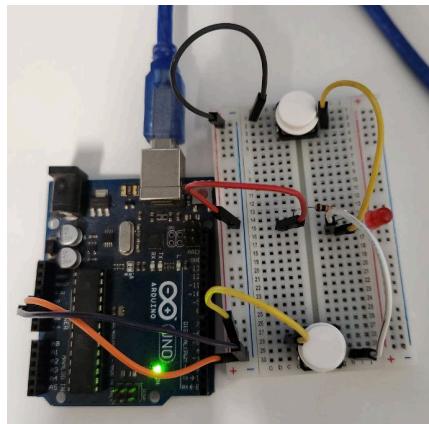
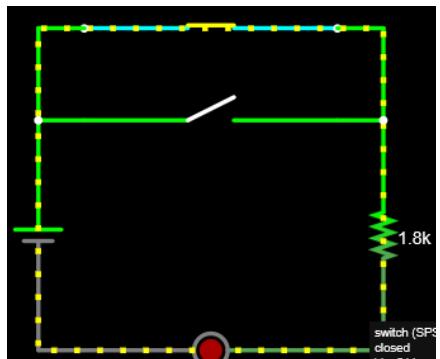
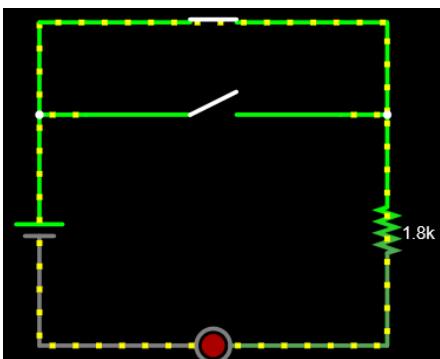


POR TA OR

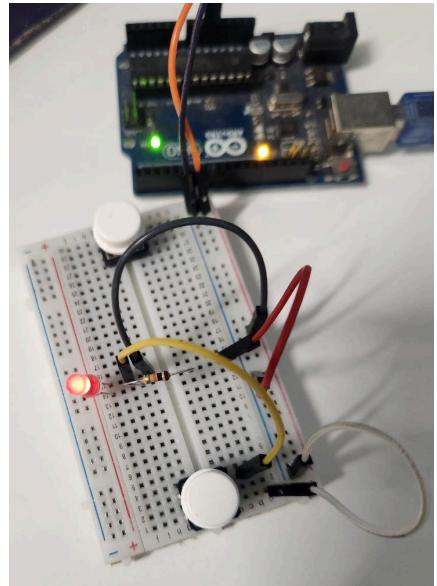
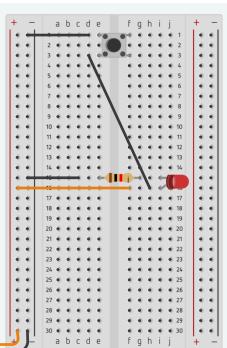
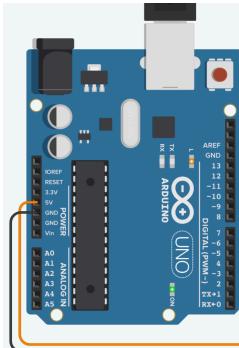


```
num1 = int(input("Escolha entre 0 e 1"))
num2 = int(input("Escolha entre 0 e 1"))

if num1 == 1 or num2 == 1:
    print("1")
else:
    print("0")
```



PORTA NOT



```
num1 = int(input("Escolha entre 0 e 1"))

if num1 == 1:
    print("0")
else:
    print("1")
```

