

Abertura: Apresente o grupo (Francisco, Renato, Arthur, Matos e Carlos Eduardo) e a disciplina.

Explique rapidamente o domínio da aplicação: **"Nosso projeto resolve um problema prático de engenharia e educação: criamos um simulador computacional da placa didática Apple Juice, usada no laboratório de Física."**

Mencione que o sistema utiliza POO em C++ para simular circuitos digitais discretos como o NE555 e CD4017.

Arquitetura e Modelagem

Encapsulamento: Mostre como os atributos (como `R1`, `R2`, `C` do Chip555) são `private` para garantir que o estado interno do chip seja sempre válido

Herança e Polimorfismo: Explique a hierarquia criada.

Gerenciamento de Recursos (RAII): Mencione rapidamente que não usamos `new` ou `delete` soltos, optando pela alocação segura na *stack* ou usando `std::vector`/ponteiros inteligentes.

Robustez, I/O e Testes

Tratamento de Exceções: Mostre rapidamente um trecho de código com o bloco `try-catch`.

Exemplo prático no código: Explique que se alguém instanciar um `Chip555` com resistência negativa, o construtor lança um `std::invalid_argument`.

Entrada e Saída (I/O) / Testes: O professor pede o uso de fluxos (`cin` / `cout`) e testes automatizados (unit tests). Como o nosso é gráfico, você pode mostrar rapidamente no terminal (rodando em paralelo) um teste unitário simples validando se o 555 calcula o tempo certo, usando `cout` para imprimir os resultados no console.

A Demonstração (Demo)

Essa é a parte principal, vamos mostrar o sistema funcionando.

Método Render/Visualizer: Explique que a visualização é feita no método da `BoardAppleJuice`, cumprindo o requisito de imprimir o estado dos objetos.

Execução: Rode o programa. Mostre a interface gráfica da Raylib.

Interatividade: Demonstre as funcionalidades ao vivo:

- Aperte a tecla `ENTER` para ligar a simulação.
- Mostre o LED do Clock (555) piscando e os LEDs do Contador (4017) avançando em sincronia.
- Aperte a tecla `R` para mostrar o reset do circuito em tempo real.

(Bônus): Mencione que a simulação elétrica roda em uma thread separada da interface gráfica, protegida por um `mutex`, o que torna o simulador altamente responsivo.

Encerramento: Agradeça a atenção e deixe claro que o código-fonte está disponível no repositório exigido (GitHub) e (talvez) convide o professor para a apresentação presencial onde vocês farão a demonstração ao vivo.