

FIAP

Disciplina: Arquiteturas Disruptivas e Big Data

Curso: Tecnologia em Análise e Desenvolvimento de Sistemas

NAC 10

Instruções:

- **LEIA ATENTAMENTE ESTAS INSTRUÇÕES E OS ENUNCIADOS DOS EXERCÍCIOS!**
- A prova deverá ser realizada em grupos de cinco alunos, porém cada exercício da prova deverá ser apresentada e enviada pelo computador do aluno responsável
- Cada grupo receberá um número de identificação. Guarde-o.
- Cada aluno deverá indicar os nomes dos integrantes do grupo como comentário do programa que está entregando
- A montagem dos componentes do Arduino ficará a cargo dos próprios alunos, que não poderão pedir auxílio aos demais grupos. A cada intervenção do professor será descontado 0,5 ponto da nota da NAC, a não ser que os componentes apresentem problemas
- O arquivo resultante de exercícios realizados deve ser compactados no formato GrupoNN.ZIP e enviado para o sistema de entregas, onde NN é o número do grupo
 - **ATENÇÃO: entregas que não seguirem o padrão sofrerão desconto de 0,5 ponto em cada exercício em que o erro for cometido**

Exercício 1 (1.5 pontos) – Montagem dos componentes do Arduino. Ligue os componentes como mostrado abaixo

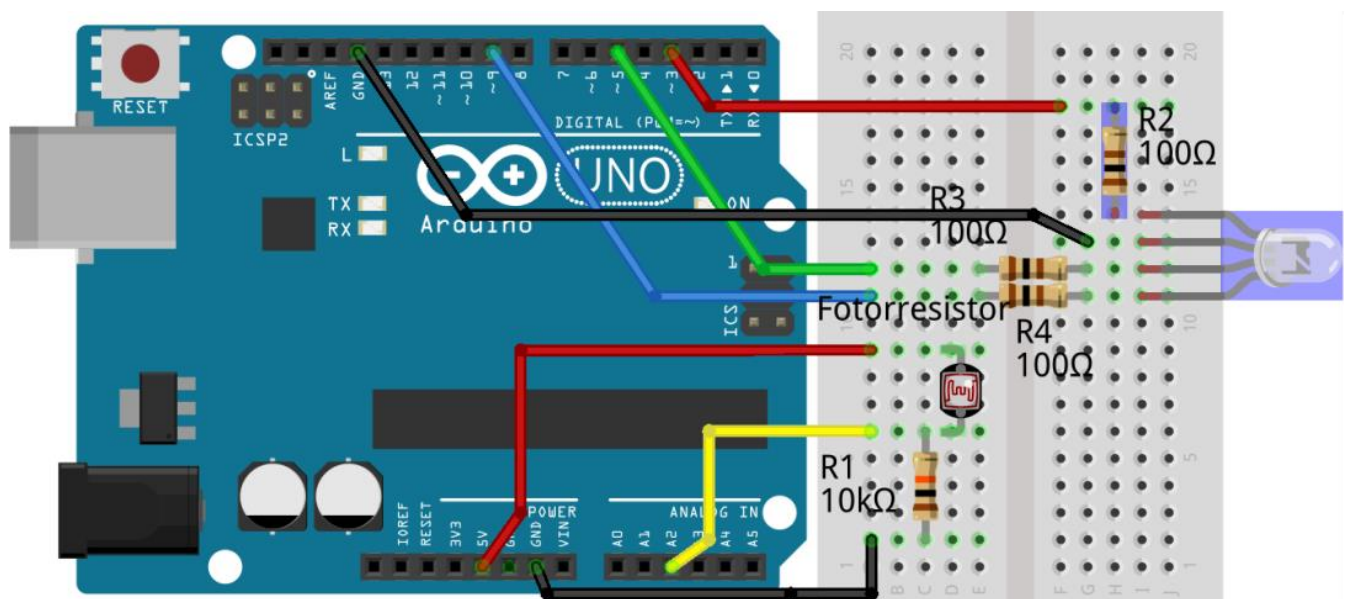


Figura 1: esquema de montagem dos componentes

O LED apresentado acima é um LED tricolor com quatro terminais, sendo um para acionar a junção dopada com material para produzir luz azul (porta 9), outro para acionar a junção dopada com material para gerar a luz verde (porta 5), outro para acionar a junção dopada com material para gerar a luz vermelha (porta 3), e o terceiro (catodo, o polo negativo) comum às três junções.

Embora normalmente seja tratado por LED tricolor (RGB), esse tipo de LED é na realidade um "multicolor", já que além das três cores independentes, cada qual gerada em uma junção, essas três junções podem ser simultaneamente polarizadas, resultando na emissão de luz amarela, alaranjada, ciano, etc.



Figura2: LED tricolor

Este exercício não possui entrega e será verificado pelo professor em sala.

Exercício 2 (1.5 pontos) – Programação do Arduino.

Use o Arduino IDE para programar o Arduino para que receba um comando externo via porta serial para o acendimento do LED tricolor, aceitando um JSON que pode conter pelo menos um dos seguintes campos, onde `nnn` é um número inteiro de 0 a 255:

- “red”: `nnn` -> liga o anodo da luz vermelha com a potencia `nnn`
- “green”: `nnn` -> liga o anodo da luz verde com a potencia `nnn`
- “blue”: `nnn` -> liga o anodo da luz azul com a potencia `nnn`

Além disso, no mesmo programa, faça o Arduino enviar a informação lida no sensor de luz (porta A2) no formato JSON a cada 2 segundos, usando o campo “luz”.

Teste o funcionamento do programa através do Serial Monitor e mostre o funcionamento para o professor analisar. Salve o arquivo como `Exercicio2.ino` e compacte com o nome `Nome.ZIP`.

Exercício 3 (3 pontos) – Programação do Gateway usando o Node-RED

Faça a programação do Gateway que conecta as informações do Arduino a um servidor MQTT, que estará rodando na máquina local através do comando `mosquitto`.

Os dados de saída do Arduino devem ser encapsulados em um JSON com um único campo `"d"`, e submetidos a um tópico MQTT formado de acordo com o padrão:

```
fiap/lab/<lab-ID>/grupo/<grupo-ID>/maquina/<NN>/sensores
```

onde `<NN>` é o número da máquina do Gateway, com dois dígitos, `<lab-ID>` é o identificador do laboratório e `<grupo-ID>` é o identificador do grupo. Por exemplo, se o grupo 03 tem seu Arduino conectado na máquina 09 do lab 702, que envia a informação `{"luz":650}`, então o JSON `{"d":{"luz":650}}` deve ser mandado para o seguinte tópico MQTT:

```
fiap/lab/702/grupo/03/maquina/09/sensores
```

De modo análogo, o Gateway deve assinar o seguinte tópico MQTT:

```
fiap/lab/<lab-ID>/grupo/<grupo-ID>/maquina/<NN>/comandos
```

onde `<NN>`, `<lab-ID>` e `<grupo-ID>` são os mesmos identificadores acima mencionados. O conteúdo da mensagem deve vir no formato JSON, dentro do campo `"cmd"`. Por exemplo, na chegada de uma mensagem `{"cmd":{"red":200}}`, o Gateway deve enviar o seguinte comando para o Arduino: `{"red":200}`

Crie também um outro flow do Node-RED para testar o Gateway. Exporte ambos os flows para um arquivo denominado `Exercicio3.json`, e compacte para o arquivo `GrupoNN.ZIP`.

Exercício 4 (2 pontos) – Programação da Aplicação Web usando o Node-RED

Use um flow do Node-RED para programar uma aplicação web que exporte as funcionalidades de leitura dos sensores e execução de comandos do Arduino através de uma API HTTP que trafega dados no formato JSON. A aplicação web deverá tanto receber quando enviar os dados para o Arduino através dos tópicos MQTT criados no exercício anterior. A API deve ter duas entradas:

- Entrada tipo GET, na URL `/luz`, que deverá devolver o último valor lido da luminosidade do Arduino através do campo `"value"`, não se esquecendo de informar o tipo de dado de retorno no cabeçalho HTTP. Exemplo: `{"value":650}`
- Entrada tipo PUT, na url `/led`, que deverá receber um JSON contendo o campo `"color"` e enviar o comando correto ao Arduino que faça com que o LED acenda com a cor desejada, sempre através do tópico MQTT correspondente. Os valores permitidos ao campo `"color"` do JSON são as strings `"BLACK"`, `"WHITE"`, `"RED"`, `"GREEN"`, `"BLUE"`, `"ORANGE"`, `"YELLOW"`, `"CIAN"` ou `"PURPLE"`. Essa entrada deve

responder com o código de resposta 400 caso a requisição tiver sido mal feita. A tabela abaixo contém a combinação de cores RGB necessária:

- BLACK: (0, 0, 0)
- WHITE: (255, 255, 255)
- RED: (255, 0, 0)
- GREEN: (0, 255, 0)
- BLUE: (0, 0, 255)
- ORANGE: (255, 64, 0)
- YELLOW: (255, 255, 0)
- CIAN: (0, 255, 255)
- PURPLE: (255, 0, 255)

Por exemplo, para uma entrada `{"color": "WHITE"}`, deve ser enviado para o MQTT o JSON `{"cmd":{"red":255,"green":255,"blue":255}}`.

Exporte o flow para um arquivo denominado `Exercicio4.json`, e compacte para o arquivo `GrupoNN.ZIP`.

Exercício 5 (2 pontos) – Programação do Cliente HTTP usando o Node-RED

Use um flow do Node-RED para programar um cliente web que consuma as funcionalidades de leitura dos sensores e execução de comandos do Arduino criadas no exercício anterior. O cliente deve usar um node tipo “inject” para criar as entradas a serem enviadas, enquanto as respostas devem ser mostradas na aba de depuração.

Exporte o flow para um arquivo denominado `Exercicio5.json`, e compacte para o arquivo `GrupoNN.ZIP`.