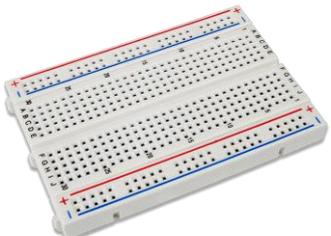
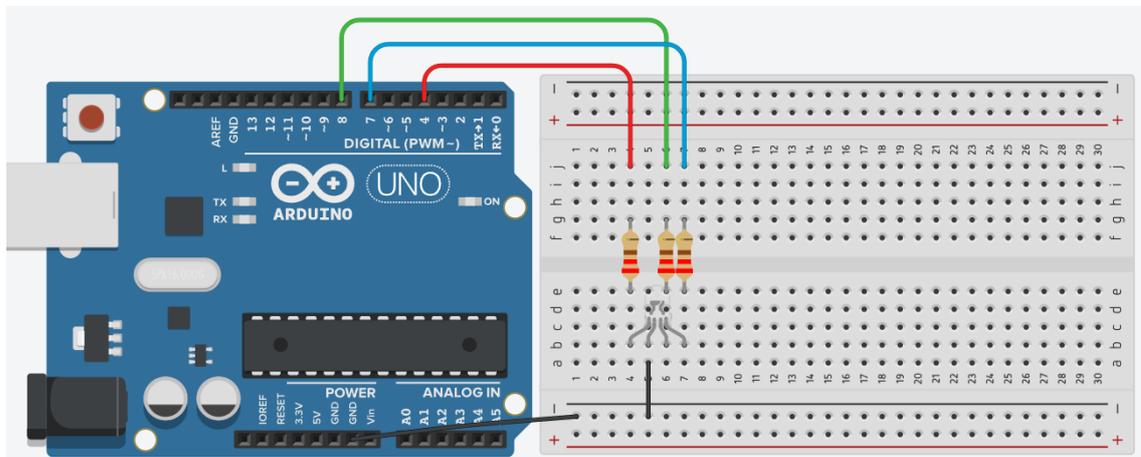




 Pretende-se que **continues** na **prática a atividade** do **LED RGB** realizada em laboratório **anteriormente**. Para isso irás **necessitar** do seguinte **material**:

<p>Arduino UNO</p> 	<p>Resistências 220Ω</p> 
<p>LED RGB cátodo comum</p> 	<p>Breadboard</p> 
<p>Cabos tipo Dupont</p> 	<p>Multímetro</p> 

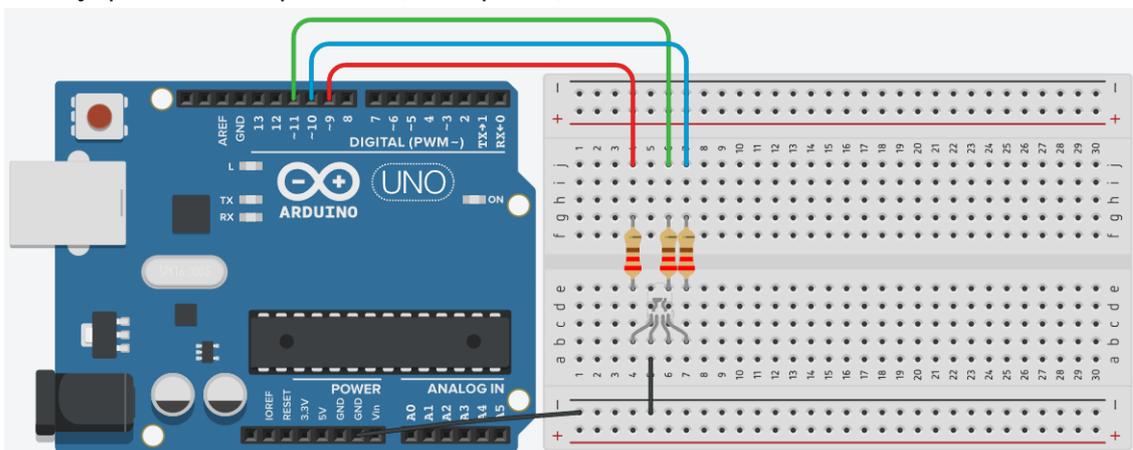
 No teu **diário de bordo**, cria um novo **título** no **final** do **documento** de nome **Arduino Laboratório 4.2 – LED RGB**. Deves ter o teu **esquema**, tal e qual montaste no laboratório anterior.





ARDUINO Mas qual a **razão** de **voltarmos** a trabalhar com o **LED RGB**? Bem, se reparaste apenas conseguias **algumas cores** com o **LED RGB**. E se eu te disser que é possível **16 milhões de cores** com este mesmo **LED**? Pois, mas par isso temos de **alterar um bocadinho** o nosso **esquema elétrico** e **também** o nosso **código**.

ARDUINO Começa por **alterar** os pinos de 4, 7 e 8 para **9, 10 e 11**.



Nota: **podes utilizar** qualquer **outro pino** desde que sejam **PWM** (Pulse Width Modulation)

ARDUINO Qual a **razão** desta **alteração**? Como já falaste nas aulas **teóricas** estes pinos com o **til ~** são **PWM** e são **sensíveis** a **mais valores** do que apenas 0 (**LOW**) ou 1 (**HIGH**). Desta forma é **possível** dizer que **quantidade** de **cor** queremos **misturar** para obter outra cor. Acaba por ser como as **cores** de um **monitor RGB** (0-255,0-255,0-255).

ARDUINO **Abre** o IDE do **Arduino** e **abre** o **ficheiro** do **laboratório anterior 4RGB1**.

```
Sketch_0020a | Arduino IDE 2.2.1
Arquivo  Editar  Recurso  Ferramentas  Ajuda
Selecionar Placa
Sketch_0020a.ino
1 void setup() {
2   // put your setup code here, to run once:
3 }
4
5 void loop() {
6   // put your main code here, to run repeatedly:
7 }
8
9
10
```

ARDUINO **Começa** por **guardar** o **ficheiro** com um **novo nome** para **não alterares** o **ficheiro original**.
Dá-lhe o **nome** de **4RGB2**.

ARDUINO **Procede** às seguintes **alterações** ao **código**:

Corrige o **pinout**.

Pino 9 para o **vermelho**;

Pino 10 para o **azul**;

Pino 11 para o **verde**.

ARDUINO Na função **setup()**, **mantém** o **código** do **anterior**. Nada muda.

```
void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(greenPin, OUTPUT);
}
```



 Finalmente o programa em si. Na função `loop()` é que vão **haver** as **diferenças significativas**.

Como queremos escrever valores de **0 a 255** nos **pinos PWM**, já **não** podemos utilizar a função `digitalWrite()`, mas **sim** a `analogWrite(0 a 255)`.

Nota: lembra-te das aulas teóricas onde viste que os **pinos** marcados com **til** ~tanto podem ser **digitais (entrada ou saída)** ou serem **analógicos (apenas saída)**

 Altera o teu **código** para **veres a intensidade de verde subir dos 0 aos 255**.

```
void loop() {  
  for(i=0;i<=255;i++){  
    analogWrite(greenPin, i); // i vai de 0 a 255  
    delay(10);  
  }  
}
```

Relembrar: o ciclo **FOR** permite realizar um ciclo entre o valor **inicial (0** neste caso) e **final (255** neste caso) com **incrementos** de (**1** neste caso). Esse valor será **injetado** no `analogWrite()` o que irá permitir **por ciclo** o **greenPin** ter **valores diferentes** (entre 0 e 255).

Atenção: *Tens de declarar a variável i junto dos pinos.*

```
int i;
```

 Envia o **programa** para o **Arduino**.



E verifica o **resultado**.

 Troca no **código** de **greenPin** para **bluePin** e **redPin** para **veres as diferenças**.

 Retira o ciclo **FOR** e **procede** a uma **mistura de cor**:

```
void loop() {  
  analogWrite(greenPin, 20); // 20 de verde  
  analogWrite(bluePin, 150); // 150 de azul  
  analogWrite(redPin, 40); // 40 de vermelho  
  delay(1000);  
}
```

Repara que há uma **mistura de cor** onde **prevalece** o **azul**. **Resultará** numa cor perto do **violeta**.

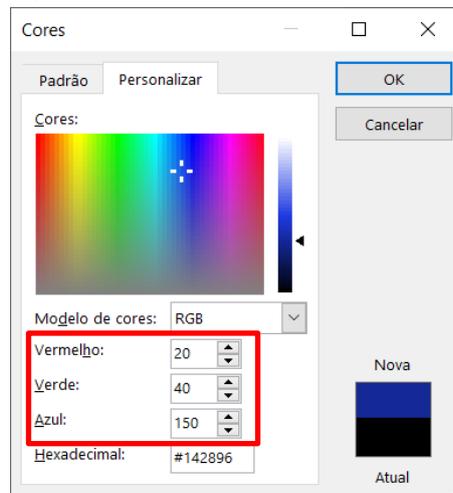
 Sabendo que:

Cor	RGB
Branco	(255,255,255)
Vermelho	(255,0,0)
Verde	(0,255,0)
Azul	(0,0,255)
Amarelo	(255,255,0)
Rosa	(255,0,255)
Violeta	(40,20,150)

Acrescenta outras cores e faz uma **sequência** ao teu gosto.



Podes recorrer a uma **paleta de cores** (word, paint, online) e **verifica** quais os **valores de RGB** que necessita **misturar** para **obter** uma **determinada cor**.



 Faz um pequeno **vídeo** com o **telemóvel** do **circuito + Arduino** com o **LED** a **funcionar** e **envia** para o teu **Drive** para a pasta **Arduino - Vídeos**. **Altera** o **nome** do ficheiro de vídeo para **4.2 – RGB**.

Tira também uma **foto** e **cola** no teu **diário de bordo**.

 **Guarda** o **programa**, no **IDE** do **Arduino** com o nome **4RGB2**. **Envia** este **ficheiro** para o teu **drive** para a pasta **Arduino - Programas**

 **Chama** o teu **professor** para **avaliar**.