Agrupamento de Escolas de Águas Santas

Curso Profissional de Técnico de Gestão e Programação de Sistemas Informáticos

Ano letivo 2021/2022

Turma 12.ºH

RELATÓRIO

PROVA DE APTIDÃO PROFISSIONAL

Braço Robótico

José Nogueira n.º 12

Nuno Pereira n.º13

Março 2022

Cofinanciado por:

Braço Robótico

2021/2022

José Nogueira

Nuno Pereira

12.º H

Professores orientadores:
António Neves
Diana Almeida
Teotónio Silva

Cofinanciado por:

Agradecimentos

A elaboração do Relatório Final não seria possível sem o apoio de alguns intervenientes. Assim sendo, pretendemos agradecer a todos os que sempre nos apoiaram e contribuíram para a realização e concretização desta nossa Prova de Aptidão Profissional.

Gostaríamos de agradecer aos professores orientadores, António Neves, Diana Almeida e Teotónio Silva, pela sua disponibilidade e compreensão, orientando, por um lado, e guiando o desenrolar do nosso trabalho durante o período da PAP, por outro, manifestando sempre as suas opiniões enriquecedoras para o desenvolvimento deste trabalho e deste relatório final.

À professora Ilda Anjos, pela sua dedicação e pela sua ajuda na correção de textos. Um agradecimento especial aos meus colegas de turma, por respeitarem o espaço de trabalho, cumprindo as regras para uma boa concentração e foco no trabalho a desenvolver ao longo deste período de PAP.

Descrição Sumária

Ao longo deste relatório será apresentado o projeto *Braço Robótico*, realizado no âmbito do nosso Projeto de Aptidão Profissional.

A base do projeto é fazer com que o braço trabalhe, de forma a conseguir pegar e transportar um objeto num eixo de 180º, até onde o utilizador desejar.

Começamos pela montagem do circuito e da programação na aplicação *Tinkercad*. Logo depois da programação e montagem no *Tinkercad* estarem concluídas, iniciamos a montagem do braço. Numa base de madeira MDF hidrófugo, o braço (com a sua base incluída) foi aparafusado na base de madeira, junto com o Arduíno Uno, uma placa de ensaio e um suporte para 4 pilhas de formato AA de 1,5 Volts.

Já com tudo totalmente montado, foram feitas as ligações entre os Servos Motores, Arduíno, Placa de Ensaio e o Suporte de Pilhas e os Joysticks, de acordo com a montagem realizada no *Tinkecad*.

Índice

Índice de figuras

Índice de tabelas

Siglas e Acrónimos

PAP – Projeto de Aptidão Profissional.

MDF – Painel de Fibras de média densidade formado por aglutinação das fibras de madeira com resinas, em processo seco.

FCT - Formação em Contexto de Trabalho.

AC – Arquitetura de Computadores.

PWM – Pulse Width Modulation.

ICSP – In-Circuit Serial Programmer.

1. Introdução

O Projeto de Aptidão Profissional foi desenvolvido na Escola Secundária de Águas Santas, durante um período de seis meses, a começar em Setembro e a terminar em Março.

A elaboração do projeto começou com uma ideia lançada pela professora Susana Santos, quando nos apresentou a possibilidade de podermos montar e programar um braço robótico. Nessa altura decidimos que seria um excelente desafio e que era um projeto interessantíssimo. A base do projeto é conseguir com que o braço faça um eixo de 180 graus, de forma a pegar num objeto e transporta-lo num angulo de 180 graus.

Começamos pela montagem do circuito e da programação na aplicação *Tinkercad*.

Logo depois da programação e da montagem no *Tinkercad* estarem concluídas, iniciamos a montagem do braço. Numa base de madeira MDF hidrófugo, o braço com a sua base incluída foi aparafusado na base de madeira, junto com o Arduíno Uno, uma placa de ensaio e um suporte para 4 pilhas de formato AA de 1,5 Volts.

Já com tudo totalmente montado, foram feitas as ligações entre os Servos Motores, Arduíno, Placa de Ensaio e o Suporte de Pilhas e os Joysticks, de acordo com a montagem realizada no *Tinkercad*.

1.1 Objetivos

Os objetivos da PAP prendem-se com uma pesquisa sobre a programação necessária para o funcionamento do braço robótico que consistiu no seguinte: planificação da montagem do braço robótico; montagem do braço robótico; programação do braço robótico; testagem e melhoramento do braço robótico; pesquisa da programação e programas para a criação da aplicação; elaboração da aplicação e conexão ao braço robótico; a testagem da aplicação e o último objetivo foi a testagem completa do braço robótico com a aplicação.

1.2 Planeamento do projeto

O nosso projeto foi desenvolvido em várias fases, tal como mostra a tabela. Iniciamos este trabalho em setembro e terminamos em março. Ao longo deste período completamos todas as propostas que apresentamos no tempo em que nos comprometemos a fazer.

	setembro	outubro	novembro	dezembro	janeiro	fevereiro	marco
Etapa 1							
Etapa 2							
Etapa 3							
Etapa 4							
Etapa 5							

Tabela 1 – Etapas do Projeto

<u>Na primeira etapa</u>, inicializamos a Proposta de PAP com a:

. Elaboração do Plano PAP;

. Pesquisa/Estudo sobre a montagem e programação do braço robótico;

. Elaboração do circuito e do modelo do projeto na aplicação Online *TinkerCad*;

Na segunda etapa, inicializamos a Montagem e Programação do Braço Robótico com a:

. Pesquisa para a elaboração da aplicação para controlo do braço Robótico;

. Montagem do braço Robótico;

. Programação do braço Robótico;

Na terceira etapa, realizamos o Desenvolvimento da Aplicação;

Na quarta etapa, realizamos o Desenvolvimento do Site e do Relatório;

Na quinta etapa, finalizamos o projeto com as seguintes fazes:

. Testagem do braço robótico e eventuais erros;

. Testagem braço robótico com aplicação e eventuais erros;

. Finalização do braço Robótico;

. Conclusão da elaboração do Site;

. Conclusão da elaboração do Relatório;

2. Descrição Técnica

2.1 Recursos

Para a concretização do projeto, utilizamos a plataforma *Tinkercad*, para o desenvolvimento inicial da programação. Depois dos testes realizados na plataforma, utilizamos o *Software* ARDUINO IDE para a programação do braço robótico.

Para a construção do braço robótico foi fornecido um *Kit Arm Robot* da *Ebotics*, com os seguintes componentes:

- 1 Arduíno UNO R3;
- 2 Módulos de Joysticks;
- 1 Placa-sensor;
- 4 Micros 9g servos;
- 8 Cabos Fêmea – Fêmea;
- 3 Cabos Macho – Fêmea;
- 1 Estrutura para o Braço;
- 1 Base para pousar o braço;
- 1 Cabo USB Tipo A Tipo B;
- 1 Bateria de 9V;
- 52 Parafusos, de 5 tamanhos diferentes;
- 26 Porcas;

O *Hardware* utilizado foi o meu portátil pessoal com as seguintes especificações:

- Rayzen 5 3500U;
- Gráfica Vegas 8 4GB;
- Memória RAM 8GB;
- SSD 256GB;

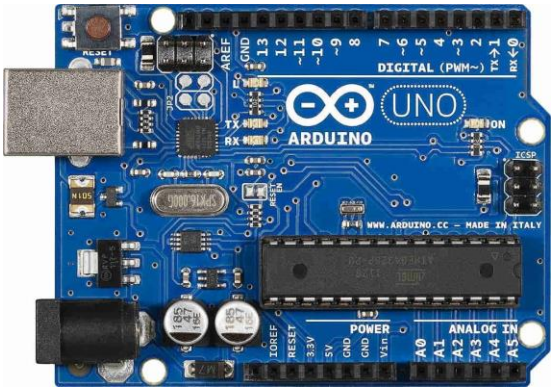
Utilizamos equipamento escolar como: Ferro de Soldar e arame para algumas soldas necessárias, computador para o desenvolvimento *Web* e para algumas pesquisas e trabalhos mais leves.

Item	Componente	Preço Unitário	Quantidade	Subtotal
1	Placa Ensaio 400 Pontos	4,50€	1	4,50€
2	Potenciómetro Linear 1*K* Metálico	2,15€	4	8,60€
3	Cabos *Jumpers* Macho-Macho Coloridos	-	20	1,73€
4	Cabos *Jumpers* Macho-Fêmea Coloridos	-	20	1,73€
5	*Arm Robot Ebotics Robotic Arm Kit*	---	1	---
6	Suporte 4 Pilhas AA	2,45€	1	2,45€
7	Servo Motores MG60S 360º	9,34€	2	18,68€
8	4 Pilhas AA	1,95€	2	3,90€
9	16 Piece Student Tools Kit	1€	1	1€
10	Mangas Térmicas	-	15	0,20€
			Total	*42,79 €*

Tabela 2 – Componentes e respetivos preços

- *Tinkercad* – *Tinkercad* foi utilizado para fazer a simulação do funcionamento dos servos motores;

- Arduíno – Arduíno é utilizado para programar o braço e para enviar os dados para o braço funcionar e estar programado;



- Suporte 4 pilhas AA – O suporte de 4 pilhas AA é utilizado para alimentar a placa de ensaio e sucessivamente alimentar só os servos motores;

- Servos Motores – são utilizados para movimentar o braço na direção que pretendermos;

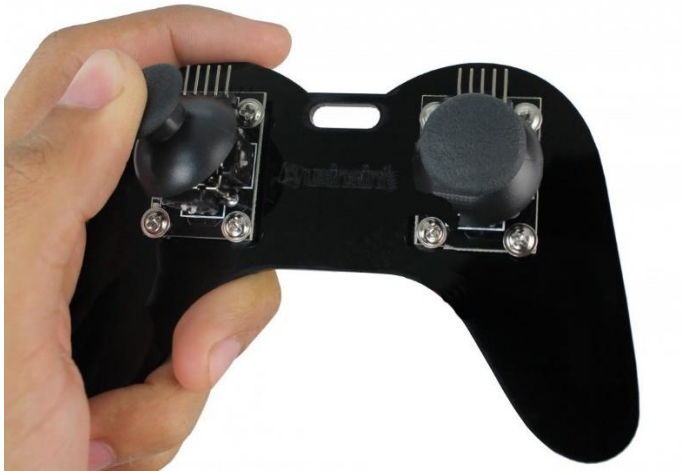
- *JoySticks* - Os dois *JoysSticks* servem para controlar a partir dos eixos (x,y) o braço robótico;

- Base Madeira – A Base de Madeira MDF cinzenta foi concebida para suportar e reforçar a base do braço robótico assim como incluir a placa de ensaio e o suporte de pilhas AA;

- Jumpers – Jumpers Macho – Macho, Jumpers Macho – Fêmea, serviram para fazer as ligações entre Placa de Ensaio – Arduíno – Servos Motores;

- Mangas Térmicas – serviram para esconder as soldas realizadas;

- Comando – Serviu de Base para os Joysticks;



2.2 Desenvolvimento

Este projeto foi desenvolvido com base num *Arm Robot* da *Ebotics*.

Para que com o *Kit* da *Ebotics* pudéssemos montar o braço robótico, necessitávamos de aprender a ajustar os servos motores, todas as funcionalidades da nossa *board* (Arduíno UNO R3), o que cada porta permite executar e que funcionalidade iriamos atribuir. Precisávamos também de conhecer e aprofundar o nosso conhecimento na linguagem de programação do Arduíno.

Arduíno UNO R3

Arduíno UNO R3 é uma placa de desenvolvimento baseada no microcontrolador ATmega328 (*datasheet*). Tem 14 pinos digitais de entrada/saída (dos quais 6 podem ser utilizadas como saídas *PWM*), 6 entradas analógicas, um cristal oscilador 16 *MHz*, uma ligação USB, um *jack* alimentação, *ICSP*, e um botão *reset*. Contém tudo o que é necessário para programar o microcontrolador, basta ligá-lo a um computador com um cabo *USB* para enviar a programação ou passar energia, ou liga-lo a um adaptador AC-DC para dar só energia assim como a uma bateria para começar a utiliza-lo.

Planificação do Projeto

O projeto consistiu em elaborar um braço robótico que fosse capaz de transportar um objeto num eixo de 180 graus.

No início tínhamos como ideia criar um braço robótico, programá-lo e mexê-lo, a partir de uma aplicação para *Android*, mas concluímos que seria melhor abandonar esta ideia e criar, a partir dos módulos de *Joystick* a programação necessária para conseguir com que o braço robótico funcionasse.

De seguida, iremos explicar melhor esta introdução básica da planificação do projeto, explicando todas as etapas e como elas foram desenvolvidas.

Controlo do Braço

No início tivemos complicações relativamente ao desenvolvimento do projeto. Após um trabalho em equipa, sobretudo na elaboração de pesquisas, decidimos excluir a ideia inicial de controlar o braço robótico a partir de um módulo *Bluetooth* e usarmos única e exclusivamente os módulos de *Joysticks* para o controlo dos movimentos dos servos motores, e assim conseguir mexer o braço robótico.

Para o projeto ser desenvolvido corretamente dividimo-lo em algumas partes. A primeira consistiu na elaboração de uma simulação no *Tinkercad,* com a primeira programação realizada, para vermos, mais ou menos, como iria ficar a montagem do circuito, a programação dos servos motores e da disposição dos componentes.

A segunda parte procedemos à montagem, a partir das instruções do manual do *Kit* da *Ebotics*. Uma vez concluída a montagem do braço, partimos para a terceira etapa.

A terceira etapa fizemos a planificação da disposição dos componentes necessários na base de madeira.

A quarta etapa resultou na instalação dos componentes na base de madeira, entre eles: o suporte de pilhas, que fornece corrente a uma parte da placa de ensaio e energia aos servos motores; a placa de ensaio que recebe, como já referido, energias das pilhas e faz a transferência para os servos motores. Recebe também energia do Arduíno na outra metade e transfere essa energia aos módulos de *Joysticks*. O Arduíno é o núcleo principal, o que faz com que tudo funcione, receba a informação dos *Joysticks* e transmita a informação para os servos motores. Também armazena o código sendo o responsável por correr a programação.
Já com todos os componentes instalados, passamos para a ligação de todos os servos motores ao Arduíno e a placa de ensaio.

A quinta etapa foi concluir a programação, e enviar para o Arduíno e testar para ver se tudo funcionava corretamente. Com algumas dificuldades e desafios na programação, conseguimos descobrir a melhor programação e deixar tudo a funcionar.

Simulação *Tinkercad*

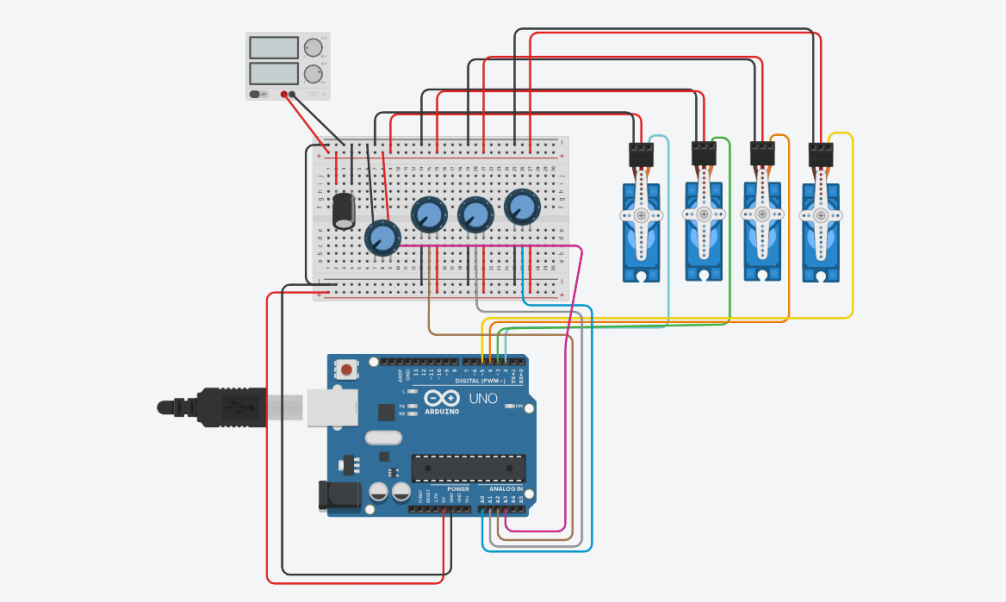


Figura 1 – Simulação Tinkercad

A aplicação *Tinkercad* foi-nos útil para realizarmos a simulação do circuito e a respetiva programação inicial que encontra-se em anexo.

Programação Arduíno

```
#include <Servo.h>

#define servoBase        6  // Porta Digital do Servo da Base
#define servoAlturaBraco 3  // Porta Digital do Servo da Altura do Braço
#define servoAnguloBraco 5  // Porta Digital do Servo do Ângulo do Braço
#define servoGarra       4  // Porta Digital do Servo da Garra

#define potBase         A0  // Porta Analógica do Potenciometro para Controle da Base
#define potAlturaBraco  A1  // Porta Analógica do Potenciometro para Controle da Altura do Braço
#define potAnguloBraco  A2  // Porta Analógica do Potenciometro para Controle do Ângulo do Braço
#define potGarra        A3  // Porta Analógica do Potenciometro para Controle da Garra.

#define botaoCongela     8  // Porta Digital do Botao Congela
#define botaoDescongela  9  // Porta Digital do Botao Descongela
```

Figura 2 – Defenição das portas no arduíno

Definição da biblioteca principal do nosso código e definição de todas as portas digitais e analógicas com um respetivo nome para ser mais fácil posteriormente na programação.

```
//Instanciação dos Objetos de Controle dos Servos
Servo base;
Servo altura;
Servo angulo;
Servo garra;

//Variaveis para cálculo dos angulos
int valorPotBase;
int valorAngBase;
int valorPotAltura;
int valorAngAltura;
int valorPotAngulo;
int valorAngAngulo;
int valorPotGarra;
int valorAngGarra;

//Variaveis para controle dos botões
int estadoBotaoC, estadoBotaoD;
int estadoAntBotaoC = HIGH;
int estadoAntBotaoD = HIGH;
boolean congelado = false;
```

Figura 3 – Variáveis

Nesta parte foi realizada a instalação dos objetos de controlo dos servos motores, a declaração de variáveis para cálculo dos ângulos dos servos motores e as variáveis de controlo dos botões.

```
//Variaveis para controle dos botões
int estadoBotaoC, estadoBotaoD;
int estadoAntBotaoC = HIGH;
int estadoAntBotaoD = HIGH;
boolean congelado = false;

void setup(){
  pinMode(botaoCongela, INPUT_PULLUP);
  pinMode(botaoDescongela, INPUT_PULLUP);

  pinMode(12, OUTPUT);
digitalWrite(12,LOW);

  //Configuração das Portas dos Servos
  base.attach(servoBase);
  altura.attach(servoAlturaBraco);
  angulo.attach(servoAnguloBraco);
  garra.attach(servoGarra);
}
```

Figura 4 – void Setup

Ainda antes do *void setup()*,há a declaração de variáveis para controlar os botões dos módulos de *Joysticks.*

Dentro do *void setup(),* fazemos a declaração dos botões como "*input_pullup*", ou seja, basta pressionar uma vez que o botão ativa. A declaração do *LED* para uma função que será posteriormente explicada, e a configuração das portas analógicas dos servos motores.

```
void loop() {

  if (!congelado) {
    valorPotBase = analogRead(potBase);              //Leitura do Angulo do Potenciometro
    valorAngBase = map(valorPotBase, 0, 1024, 0, 180); //Conversão do valor do potenciometro (de 0 até 1024) para o angulo (de 0 até 180)
    base.write(valorAngBase);                        //Envio do angulo para o Servo

    valorPotAltura = analogRead(potAlturaBraco);          //Leitura do Angulo do Potenciometro
    valorAngAltura = map(valorPotAltura, 0, 1024, 180, 0); //Conversão do valor do potenciometro (de 0 até 1024) para o angulo (de 0 até 180)
    altura.write(valorAngAltura);                         //Envio do angulo para o Servo

    valorPotAngulo = analogRead(potAnguloBraco);          //Leitura do Angulo do Potenciometro
    valorAngAngulo = map(valorPotAngulo, 0, 1024, 180, 0); //Conversão do valor do potenciometro (de 0 até 1024) para o angulo (de 0 até 180)
    angulo.write(valorAngAngulo);                         //Envio do angulo para o Servo

    valorPotGarra = analogRead(potGarra);             //Leitura do Angulo do Potenciometro
    valorAngGarra = map(valorPotGarra, 0, 1024, 0, 90); //Conversão do valor do potenciometro (de 0 até 1024) para o angulo (de 0 até 180)
    garra.write(valorAngGarra);                       //Envio do angulo para o Servo
  }
```

Figura 5 – Void loop

No *void loop()*, está o coração do código, é neste *void* aonde se encontra a programação do controlo dos servos motores.

O *if(!congelado){* - vê se o botãocongela **não** está ativo, e inicia a função.

De seguida:

valorPotBase = analogRead(potBase); - Faz a leitura do ângulo dos *Joysticks* para da base, lê o valor e ser for positivo aciona o servo para um sentido, caso seja negativo aciona o servo para o sentido inverso. Ex: valor = 1, servo vira para a direita, valor = -1, servo vira para a esquerda.

valorAngBase = Map(ValorPotBase, 0 , 1024 , 180 , 0); - Faz a conversão do valor do *Joystick* de 0 a 1024 para o angulo de 0 a 180.

Base.Write(valorAngbase) – Envia o ângulo para o servo e fá-lo mexer.

valorPotAltura = analogRead(potAltura); - Faz a leitura do ângulo dos *Joysticks* para altura, lê o valor e ser for positivo aciona o servo para um sentido, caso seja negativo aciona o servo para o sentido inverso. Ex: valor = 1, servo vira para a direita, valor = -1, servo vira para a esquerda.

valorAngAltura = Map(ValorPotAltura, 0 , 1024 , 0 , 180); - Faz a conversão do valor do *Joystick* de 0 a 1024 para o angulo de 0 a 180.

Altura.Write(valorAngAltura) – Envia o ângulo para o servo e fá-lo mexer.

valorPotAngulo = analogRead(potAngulo); - Faz a leitura do ângulo dos *Joysticks* para o ângulo, lê o valor e ser for positivo aciona o servo para um sentido, caso seja negativo aciona o servo para o sentido inverso. Ex: valor = 1, servo vira para a direita, valor = -1, servo vira para a esquerda.

valorAngAngulo = Map(ValorPotAngulo, 0 , 1024 , 0 , 180); - Faz a conversão do valor do *Joystick* de 0 a 1024 para o angulo de 0 a 180.

Angulo.Write(valorAngAngulo) – Envia o ângulo para o servo e fá-lo mexer.

valorPotGarra = analogRead(potGarra); - Faz a leitura do ângulo dos *Joysticks* para a garra, lê o valor e ser for positivo aciona o servo para um sentido, caso seja negativo aciona o servo para o sentido inverso. Ex: valor = 1, servo vira para a direita, valor = -1, servo vira para a esquerda.

valorAngGarra = Map(ValorPotGarra, 0 , 1024 , 0 , 90); - Faz a conversão do valor do *Joystick* de 0 a 1024 para o angulo de 0 a 90.

Garra.Write(valorAngGarra) – Envia o ângulo para o servo e fá-lo mexer.

```
//Se o botao de congelar foi apertado
estadoBotaoC = digitalRead(botaoCongela);
if (estadoBotaoC != estadoAntBotaoC) {
   congelado = true;
   digitalWrite(12, HIGH);

}
estadoAntBotaoC = estadoBotaoC;

//Se o botao de descongelar foi apertado
estadoBotaoD = digitalRead(botaoDescongela);
if (estadoBotaoD != estadoAntBotaoD) {
   congelado = false;
   digitalWrite(12,LOW);
}
estadoAntBotaoD = estadoBotaoD;
```

Figura 6 – Função dos botões

Nesta última parte do código criamos uma função de segurança para travar o braço.

A função funciona da seguinte forma:
Caso o botão de congelar seja apertado, a função altera o valor do estado do botão e como passa a congelado, no *if(!congelado){*, a verificação vai ser inválida então aciona o *LED* vermelho e caso tentamos mexer nos *Joysticks* não vão ativar a programação dos servos motores.

Caso a função do Botão descongela seja apertado, a função não altera o valor do estado do botão e como passa a descongelado, no *if(!congelado){*, a verificação vai ser válida então não aciona o *LED* vermelho e caso tentamos mexer nos *Joysticks* já vão ativar a programação dos servos motores.

Website

Para a apresentação da nossa PAP decidimos criar um *Website*, de forma intuitiva para uma boa explicação do projeto.

Para a realização do *Website*, foram utilizadas as seguintes linguagens de programação HTML e CSS.

A estrutura do Website está dividida em várias partes, mas todas estão na página *index.html*.

Figura 7 – Printscreen Apresentação PAP

Imagem inicial do *Website* para a introdução da apresentação da PAP.

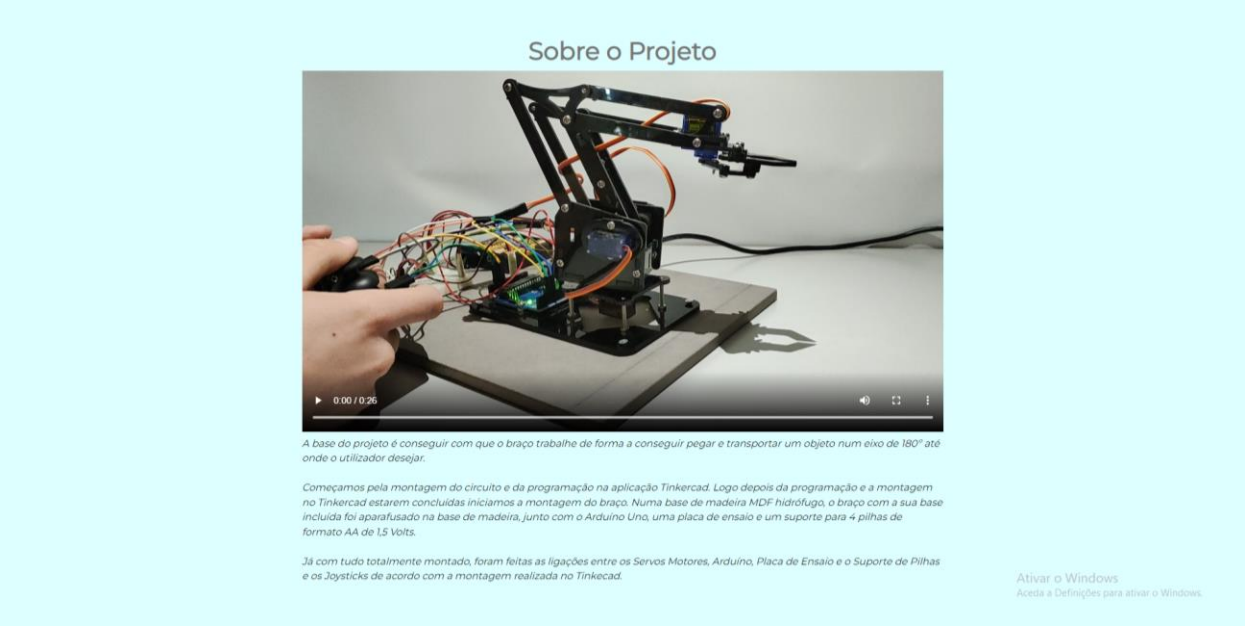


Figura 8 – Printscreen website "Sobre o Projeto"

Neste tópico vamos abordar como foi a preparação das nossas ideias e como foi criar e desempenhar todos os processos até a conclusão do braço.

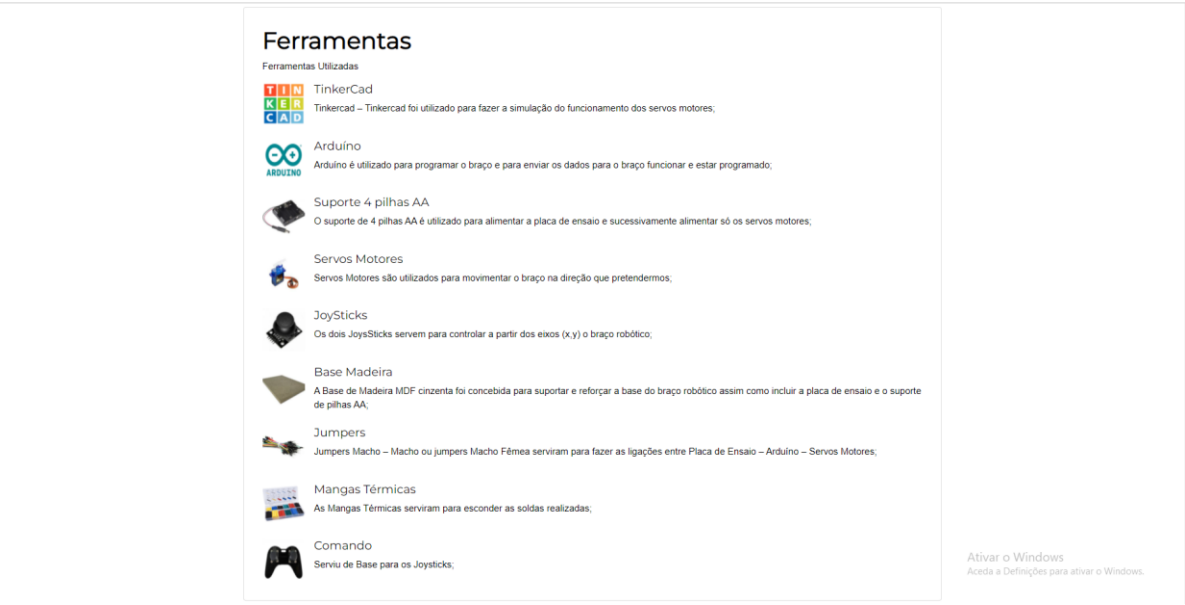


Figura 9 – Printscreen website "Ferramentas"

Neste tópico abordaremos todos os materiais necessários para a elaboração da PAP.

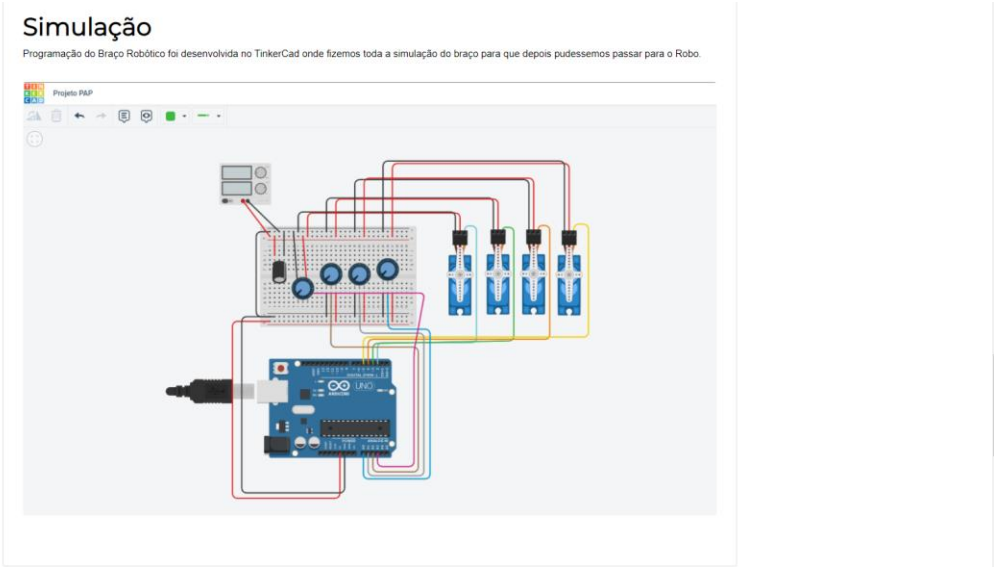


Figura 10 – Printscreen website "Simulação"

Neste tópico abordaremos a simulação inicial realizada no *Tinkercad*.

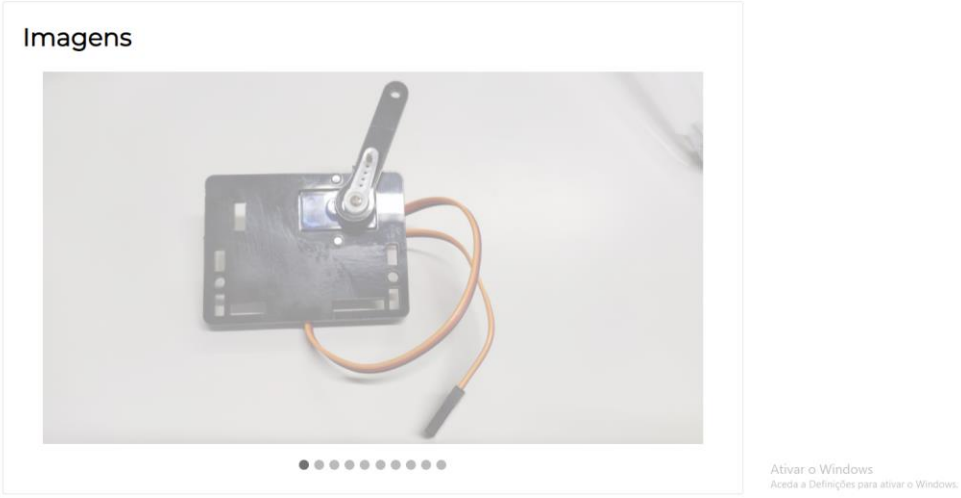


Figura 11 – Printscreen website "Imagens"

Neste tópico mostraremos imagens da construção e montagem do braço robótico.

Figura 12 – Printscreen website "Conclusão"

Este tópico será única e exclusivamente para finalizar a apresentação da PAP.

3. Reflexão crítica

Em termos globais, com este trabalho tivemos a oportunidade de cooperar e trabalhar em dupla, de forma a superarmos todas as dificuldades que pudessem surgir.

Desde sempre gostamos de explorar novos conhecimentos e com a FCT, realizada no 11º ano e com os conhecimentos adquiridos nas disciplinas da formação técnica, da formação sociocultural e da formação científica, ganha-mos bases e interesse nesta área, e então decidimos explorar mais e realizar este projeto.

Para a conclusão e realização deste projeto tivemos de ter uma grande capacidade de autonomia, de persistência e de entreajuda. Ambos concretizamos todos os passos existentes e impostos no nosso trabalho.

Ao longo destes meses de trabalho árduo, enfrentamos várias dificuldades tais como, o tempo de espera de peças para o projeto que atrasou a sua montagem, bem como a sua programação.

Durante a programação houve alguns problemas nos servos motores, e com a pinça do braço robótico, mas todas as dificuldades enfrentadas foram superadas.

Em suma, a prova de aptidão profissional foi um grande desafio e colocou-nos à prova das nossas capacidades, a todos os níveis.

Gostamos muito de trabalhar juntos neste projeto e concretiza-lo de forma exímia.

Para o futuro temos planos como a elaboração de uma aplicação *Android* para controlar a movimentação do braço, acrescentar um led verde, que indique o braço está a realizar uma tarefa, e um led vermelho que indique que o braço está em repouso.

Referências

Servo Motor

- https://docs.arduino.cc/learn/electronics/servo-motors [consultado em novembro 2021]

- https://www.arduinoportugal.pt/controlando-um-servomotor-arduino/ [consultado em novembro 2021]

- https://blog.fazedores.com/como-usar-servo-motor-com-arduino/ [consultado em novembro 2021]

Braço Robótico

- https://www.arduinoportugal.pt/braco-robotico-desenvolvimento-do-projeto/ [consultado em outubro 2021]

- https://www.youtube.com/watch?v=UfFC4gcr5tA [consultado em dezembro 2021]

- https://www.youtube.com/watch?v=vVOyWQZ25M8 [consultado em dezembro 2021]

- https://www.youtube.com/watch?v=Ecw3kCo4AdQ [consultado em dezembro 2021]

Electrofun

- https://www.electrofun.pt [consultado em outubro2021]

Aquario

- https://www.aquario.pt [consultado em outubro 2021]

Ebotics

- https://ebotics.com/product/arm-robot/ [consultado em setembro 2021]

Anexos

Anexo I Programação Tinkercad

```
// inclui bibilioteca do servomotor
#include <Servo.h>

// define pinos dos servos
#define pinServ1 2
#define pinServ2 3
#define pinServ3 4
#define pinServ4 5

// define as portas dos potenciometros
#define pot1 A0
#define pot2 A1
#define pot3 A2
#define pot4 A3

// nomeia os servos
Servo serv1,serv2,serv3,serv4;

// cria as variavies dos angulos de cada motor
int motor1,motor2,motor3,motor4;

// variaveis para serem usadas na funcao que imprime no monitor serial a posicao dos motores
unsigned long mostradorTimer = 1;
const unsigned long intervaloMostrador = 2000;

void setup() {

  //inicia o monitor serial
  Serial.begin(9600);

  // atribui pinos dos servos
  serv1.attach(pinServ1);
  serv2.attach(pinServ2);
  serv3.attach(pinServ3);
  serv4.attach(pinServ4);

}

void loop(){

  // leitura dos potenciometros
  motor1 = map(analogRead(pot1),0,1023,0,180);
  motor2 = map(analogRead(pot2),0,1023,90,0);
  motor3 = map(analogRead(pot3),0,1023,0,80);
  motor4 = map(analogRead(pot4),0,1023,0,43);

  // posicionamento dos potenciometros
  serv1.write(motor1);
  serv2.write(motor2);
  serv3.write(motor3);
  serv4.write(motor4);

   if ((millis() - mostradorTimer) >= intervaloMostrador) {
```

```
  // envio para o monitor serial do posicionamentos dos motores
  Serial.println("************************************************");

  Serial.print("Pot1:");
  Serial.print(analogRead(pot1));
  Serial.print(" Angulo Motor1:");
  Serial.println(motor1);

  Serial.print("Pot2:");
  Serial.print(analogRead(pot2));
  Serial.print(" Angulo Motor2:");
  Serial.println(motor2);

  Serial.print("Pot3:");
  Serial.print(analogRead(pot3));
  Serial.print(" Angulo Motor3:");
  Serial.println(motor3);

  Serial.print("Pot4:");
  Serial.print(analogRead(pot4));
  Serial.print(" Angulo Motor4:");
  Serial.println(motor4);

  mostradorTimer = millis();
  }

  // tempo de espera para recomeçar
  delay(100);

}
```

Anexo II Programação Arduíno

```
#include <Servo.h>

#define servoBase       6
#define servoAlturaBraco 3
#define servoAnguloBraco 5
#define servoGarra      4

#define potBase         A0
#define potAlturaBraco  A1
#define potAnguloBraco  A2
#define potGarra        A3

#define botaoCongela    8
#define botaoDescongela  9

Servo base;
Servo altura;
Servo angulo;
Servo garra;

int valorPotBase;
int valorAngBase;
int valorPotAltura;
int valorAngAltura;
int valorPotAngulo;
int valorAngAngulo;
int valorPotGarra;
int valorAngGarra;

int estadoBotaoC, estadoBotaoD;
int estadoAntBotaoC = HIGH;
int estadoAntBotaoD = HIGH;
boolean congelado = false;

void setup(){
  pinMode(botaoCongela, INPUT_PULLUP);
  pinMode(botaoDescongela, INPUT_PULLUP);

  pinMode(12, OUTPUT);
digitalWrite(12,LOW);

  //Configuração das Portas dos Servos
  base.attach(servoBase);
  altura.attach(servoAlturaBraco);
  angulo.attach(servoAnguloBraco);
  garra.attach(servoGarra);
}


void loop() {

  if (!congelado) {
    valorPotBase = analogRead(potBase);
    valorAngBase = map(valorPotBase, 0, 1024, 0, 180);
    base.write(valorAngBase);

    valorPotAltura = analogRead(potAlturaBraco);
    valorAngAltura = map(valorPotAltura, 0, 1024, 180, 0);
```

```
  altura.write(valorAngAltura);

  valorPotAngulo = analogRead(potAnguloBraco);
  valorAngAngulo = map(valorPotAngulo, 0, 1024, 180, 0
  angulo.write(valorAngAngulo);

  valorPotGarra = analogRead(potGarra);
  valorAngGarra = map(valorPotGarra, 0, 1024, 0, 90);
  garra.write(valorAngGarra);
 }


 estadoBotaoC = digitalRead(botaoCongela);
 if (estadoBotaoC != estadoAntBotaoC) {
   congelado = true;
   digitalWrite(12, HIGH);

 }
 estadoAntBotaoC = estadoBotaoC;

 estadoBotaoD = digitalRead(botaoDescongela);
 if (estadoBotaoD != estadoAntBotaoD) {
   congelado = false;
   digitalWrite(12,LOW);
 }
 estadoAntBotaoD = estadoBotaoD;

}
```

Anexo III Programação Website

```
<!DOCTYPE html>
<html>
<head>
<title>Projeto de Aptidão Profissional</title>
<link rel="shortcut icon" type="img/icon.jpg" href="img/icon.png"/>
</head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Montserrat">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">
<link rel="stylesheet" type="text/css" href="assets/css/style.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body, h1,h2,h3,h4,h5,h6 {font-family: "Montserrat", sans-serif}
.w3-row-padding img {margin-bottom: 12px}
.w3-sidebar {width: 120px;background: #ececec;}
#main {margin-left: 120px}
@media only screen and (max-width: 600px) {#main {margin-left: 0}}

footer {
text-align: center;
padding: 3px;
background-color: black;
color: white;
}

.img {
min-height: 100%;
background-position: center;
background-size: cover;
}

</style>
<body class="w3-white">
<div class="w3-display-container w3-animate-opacity w3-text-white">
<img src="assets/img/img-01n.jpg" alt="boy" style="max-width:101%;height:auto;">
<div class="w3-display-topleft w3-padding-large w3-xlarge">
PAP
</div>
<div class="w3-display-middle">
<h1 class=" w3-animate-top">Projeto de Aptidão Profissional</h1>
<hr class="w3-border-grey" style="margin:auto;width:40%">
<p class="w3-large w3-center">Realizado: por Daniel Nogueira e Nuno Pereira</p>
</div>
<div class="w3-display-bottomleft w3-padding-large">
12ºH
</div>
</div>
<div class="w3-container w3-padding-64 w3-pale-blue " id="us">
<div class="w3-content">
<h1 class="w3-center w3-text-grey"><b>Sobre o Projeto</b></h1>
<!--<img class="w3-round" src="assets/img/IMG_20220208_102256.jpg" style="width:100% ;margin:32px 0">-->
<video style="max-width:100%;height:auto;" controls autoplay>
<source src="assets/img/VID_20220328_120054.mp4" type="video/mp4">
</video>
```


A base do projeto é fazer com que o braço trabalhe, de forma a conseguir pegar e transportar um objeto num eixo de 180º, até onde o utilizador desejar.

Começamos pela montagem do circuito e da programação na aplicação Tinkercad. Logo depois da programação e montagem no Tinkercad estarem concluídas, iniciamos a montagem do braço. Numa base de madeira MDF hidrófugo, o braço (com a sua base incluída) foi aparafusado na base de madeira, junto com o Arduíno Uno, uma placa de ensaio e um suporte para 4 pilhas de formato AA de 1,5 Volts.

Já com tudo totalmente montado, foram feitas as ligações entre os Servos Motores, Arduíno, Placa de Ensaio e o Suporte de Pilhas e os Joysticks, de acordo com a montagem realizada no Tinkecad.

<hr>

<h1>Desenvolvimento do Projeto</h1>

<h1>Ferramentas</h1>
Ferramentas Utilizadas

<img class="mr-3" src="assets/img/transferir.png" width="68" height="68" alt="Generic placeholder image" >

<h5 class="mt-0">TinkerCad</h5>
Tinkercad – Tinkercad foi utilizado para fazer a simulação do funcionamento dos servos motores;

<img class="mr-3" src="assets/img/images.png" width="68" height="68" alt="Generic placeholder image" >

<h5 class="mt-0">Arduíno</h5>
Arduíno é utilizado para programar o braço e para enviar os dados para o braço funcionar e estar programado;

<img class="mr-3" src="assets/img/1.jpg" width="68" height="68" alt="Generic placeholder image" >

<h5 class="mt-0">Suporte 4 pilhas AA</h5>
O suporte de 4 pilhas AA é utilizado para alimentar a placa de ensaio e sucessivamente alimentar só os servos motores;

<img class="mr-3" src="assets/img/2.jpg" width="68" height="68" alt="Generic placeholder image" >

<h5 class="mt-0">Servos Motores</h5>
Servos Motores são utilizados para movimentar o braço na direção que pretendermos;

```
<br>
<div class="media">
<img class="mr-3" src="assets/img/3.jpg" width="68" height="68" alt="Generic placeholder image" >
<div class="media-body">
<h5 class="mt-0">JoySticks</h5>
Os dois JoysSticks servem para controlar a partir dos eixos (x,y) o braço robótico;
</div>
</div>
<br>
<div class="media">
<img class="mr-3" src="assets/img/4.jpg" width="68" height="68" alt="Generic placeholder image" >
<div class="media-body">
<h5 class="mt-0">Base Madeira</h5>
A Base de Madeira MDF cinzenta foi concebida para suportar e reforçar a base do braço robótico assim como incluir a placa de ensaio e o suporte de pilhas AA;
</div>
</div>
<br>
<div class="media">
<img class="mr-3" src="assets/img/5.jpg" width="68" height="68" alt="Generic placeholder image" >
<div class="media-body">
<h5 class="mt-0">Jumpers</h5>
Jumpers Macho – Macho ou jumpers Macho Fêmea serviram para fazer as ligações entre Placa de Ensaio – Arduíno – Servos Motores;
</div>
</div>
<br>
<div class="media">
<img class="mr-3" src="assets/img/shopping.png" width="68" height="68" alt="Generic placeholder image" >
<div class="media-body">
<h5 class="mt-0">Mangas Térmicas</h5>
As Mangas Térmicas serviram para esconder as soldas realizadas;
</div>
</div>
<br>
<div class="media">
<img class="mr-3" src="assets/img/6.jpg" width="68" height="68" alt="Generic placeholder image" >
<div class="media-body">
<h5 class="mt-0">Comando</h5>
Serviu de Base para os Joysticks;
</div>
</div>
</div>
<br><br>
</div>
</div>
</div>
<div class="w3-container w3-padding-64 w3-pale-blue " id="us">
<div class="w3-content">
<h1 class="w3-center w3-text-grey"><b>Simulação</b></h1>
<img class="w3-round" src="assets/img/Screenshot_1.png" style="width:100% ;margin:32px 0">
<br>
<p><i>A simulação no Tinkercad, com a primeira programação realizada serviu para vermos, mais ou menos, como iria ficar a montagem do circuito, a programação dos servos motores e da disposição dos componentes.</i></p>
</div>
</div>

<br><br>
<br>
<br>
<br>
```

```
<div class="container">
<div class="row">
<div class="col-md-20">
<div class="card">
<h1><b>Imagens</b></h1>
<br>
<div class="slideshow-container">
<div class="mySlides fade">
<img src="assets/img/IMG_20211110_104912.jpg" style="width:100%">
</div>
<div class="mySlides fade">
<img src="assets/img/IMG_20211110_104853.jpg" style="width:100%">
</div>
<div class="mySlides fade">
<img src="assets/img/IMG_20211110_121813.jpg" style="width:100%">
</div>
<div class="mySlides fade">
<img src="assets/img/IMG_20211110_111022.jpg" style="width:100%">
</div>

<div class="mySlides fade">
<img src="assets/img/IMG_20211110_113156.jpg" style="width:100%">
</div>

<div class="mySlides fade">
<img src="assets/img/IMG_20211110_123227.jpg" style="width:100%">
</div>

<div class="mySlides fade">
<img src="assets/img/IMG_20211110_124733.jpg" style="width:100%">
</div>

<div class="mySlides fade">
<img src="assets/img/IMG_20211130_111749.jpg" style="width:100%">
</div>

<div class="mySlides fade">
<img src="assets/img/IMG_20211130_111759.jpg" style="width:100%">
</div>

<div class="mySlides fade">
<img src="assets/img/IMG_20220208_102314.jpg" style="width:100%">
</div>

<div class="mySlides fade">
<img src="assets/img/IMG_20220325_093546.jpg" style="width:100%">
</div><!--Foto Final-->

<br>

<div style="text-align:center">
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
```

```
<span class="dot"></span>
</div>
</div>
</div>
</div>
</div>
</div>

<br>
<br>
<br>


<div class="w3-container w3-padding-64 w3-pale-blue " id="us">
<div class="w3-content">
<h1 class="w3-center w3-text-grey"><b>Conclusão</b></h1>
<img class="w3-round" src="assets/img/IMG_0008.png" style="width:100% ;margin:32px 0">
<br>
<p><i>  <br>
<br>     </i>
</p>
</div>
</div>

<footer>
<p>Site desenvolvido por Daniel Nogueira e Nuno Pereira<br></p>
</footer>
<script>
let slideIndex = 0;
showSlides();

function showSlides() {
let i;
let slides = document.getElementsByClassName("mySlides");
let dots = document.getElementsByClassName("dot");
for (i = 0; i < slides.length; i++) {
slides[i].style.display = "none";
}
slideIndex++;
if (slideIndex > slides.length) {slideIndex = 1}
for (i = 0; i < dots.length; i++) {
dots[i].className = dots[i].className.replace(" active", "");
}
slides[slideIndex-1].style.display = "block";
dots[slideIndex-1].className += " active";
setTimeout(showSlides, 3000); // Change image every 2 seconds
}
</script>
</body>
</html>
```