



# ROBMASTER

Grupo 5

Diogo Borrego 2018262079

Fabian Dias 2016107423

João Nascimento 2016255675

José Mota 2017241108

1 de Julho de 2020

MIEEC



## SUMÁRIO

1. Informações gerais sobre o produto
  - 1.1 Proposta
  - 1.2 Escopo
  - 1.3 Arquitetura do produto e diagrama geral de blocos
2. Padrões do projeto e regulamentos governamentais aplicáveis
3. Premissas e Dependências
4. Limitações
5. Descrição do Projeto
  - 5.1 Versão física
  - 5.2 Versão Virtual
6. Atributos
7. Manutenção e suporte
  - 7.1 Requisitos de manutenção
  - 7.2 Requisitos de suporte
  - 7.3 Requisitos de adaptabilidade
8. Período de execução
9. Design
10. Interfaces
11. Confiabilidade
12. Segurança
13. Ambiente
14. Qualidade
15. Testes
16. Referencias

## 1. Informações gerais sobre o produto

### 1.1. Proposta

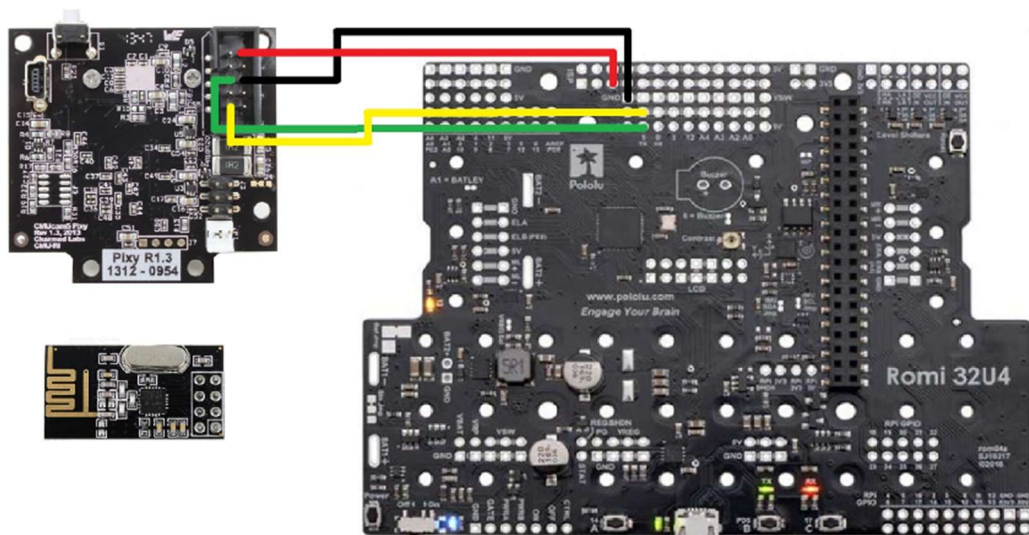
O propósito deste projeto é implementar num robot, a possibilidade de partilhar informações, objetivos e ordens com outros. Para isso seriam usados diversos sensores e um protocolo de comunicação entre eles. Em breve teremos uma rotina completamente automatizada em que todos os aparelhos se comuniquem entre si. Para que todas as tarefas funcionem o mais eficientemente possível. Precisamos de interação de aparelhos tais como smartbands, as quais vão proporcionar coordenadas das pessoas para o swarm seguir. Estes ao comunicarem entre si, podem reduzir gastos tanto energéticos quanto temporais. Carros ao andarem atrás uns dos outros numa auto estrada tendo assim mais aerodinâmica ao aproveitar o arrasto do carro na frente economizando energia, e ao comunicarem entre si é possível haver controlo do tráfego de modo a aliviar o fluxo do trânsito e economizar tempo caso seja preciso que os elementos do swarm precisem de andar mais rápido (com melhor controlo do fluxo, mais espaço há para o swarm andar). As aplicações são quase infinitas, porém devemos começar as suas implementações o quanto antes, para que seja possível desfrutar dos seus benefícios.

### 1.2. Escopo

Cada vez mais, robots são utilizados na indústria e no âmbito doméstico. Com o aumento da procura de sistemas automatizados, podemos prever que em breve estes serão indispensáveis no dia a dia. Não só no presente, mas mais interligados com a sociedade que conhecemos hoje. Swarm de robots já é uma tecnologia utilizada nos principais armazéns das maiores lojas online, para armazenar, procurar e entregar as encomendas. Também como uma forte implementação no tráfego, carros inteligentes que comunicam entre si estimando rotas mais rápidas e viagens mais seguras.

### 1.3. Arquitetura do produto e diagrama geral de blocos

O robot será programado em linguagem Arduino com bibliotecas adicionadas da câmara Pixy, Pololu Romi 32U4 e RF24 wireless. Porém após a reestruturação do projeto toda a sua implementação foi feita através do MatLab.



## 2. Padrões do projeto e regulamentos governamentais aplicáveis

Este projeto tem um carácter educacional, portanto os padrões não se enquadram em nenhum regulamento. Caso a tecnologia venha a ser aplicada no mundo real, terá de seguir os regulamentos pedidos tanto a nível governamental, o qual varia de país em país, e os requerimentos a nível de qualidade e de durabilidade pedidos em projetos como estes.

## 3. Premissas e Dependências

É esperado que devido a fase muito inicial existam algumas limitações, físicas e virtuais. O protótipo é feito com conexões rápidas e muitas partes moveis não específicas podendo assim esperar com que o robot tenha uma autonomia baixa e uma capacidade de processamento muito reduzida junto a um curto alcance de sinal e campo de visão limitado. Eventualmente a otimização de software e hardware pode ser feita de modo a contornar esses problemas junto com a produção de um novo chassi sendo preciso um redesign do robot se for usado num ambiente mais extremo

## 4. Limitações

A realidade é que a composição destes robots não está preparada para lidar com a maioria dos ambientes reais, sendo assim necessário upgrades que leva a um maior custo de cada robot. A inserção de tais tecnologias no mercado necessita um baixo custo e uma grande quantidade limitando a sua presença na maioria dos setores. Inicialmente é esperado que seja adquirido pelo setor industrial e de maneira gradual nos espaços domésticos, infiltrando-se gradualmente na sociedade.

## 5. Descrição do Projeto

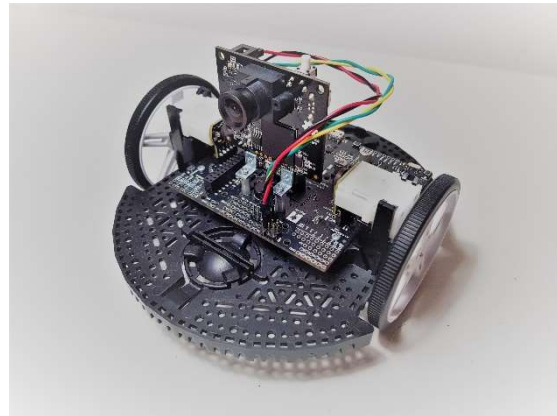
O nosso robot funciona em enxame e tem como função estar em constante comunicação com todos os outros robots dos outros grupos. O processo consiste em colocar todos os robots no mesmo ponto e o primeiro a ser colocado no ponto inicial procura informação transmitida por outros robots. No caso de não encontrar, torna-se o master e começa ele próprio a transmitir informação para todos os outros, enquanto que os restantes têm de seguir a informação transmitida pelo master. Caso algum dos outros robots já esteja a transmitir informação torna-se servo e segue essa informação. No entanto, se essa informação o levar a bater em algum objeto, ele vai encontrar uma alternativa desviando-se, de modo a chegar ao ponto em que o master se encontra ou diz para se dirigir.



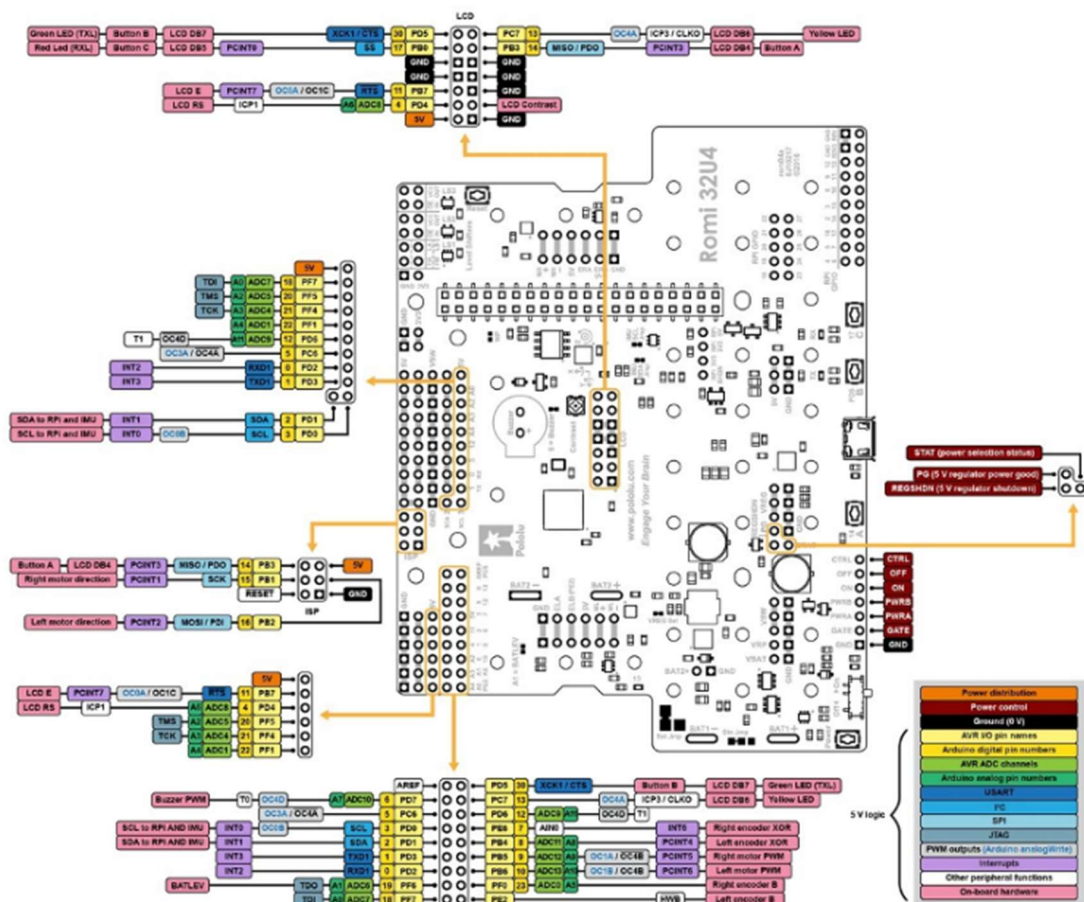
## 5.1. Versão física

### Sobre O Romi:

A placa Pololu Romi 32U4 é composta por Atmel Microcontrolador ATmega32U4 AVR e é pré-carregado com um sistema compatível com Arduino. A placa de controle possui instalada duas drivers para acionar os motores, acelerômetros de 3 eixos, detecção de inércia, botões de pressão, um buzzer, LEDs para indicação, um conector para um monitor LCD e uma entrada micro USB para comunicação com IDE através de um computador. Para a comunicação I2C a placa funciona como master tendo 7 bits para dados, pode operar a 5v e 3.3v



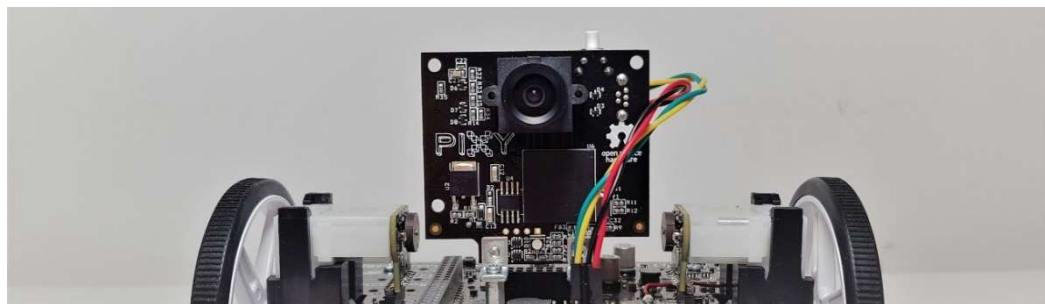
A atribuições de pinos pode ser verificada no manual do Romi na secção 3.8 e na seguinte imagem: Pinout diagram of the Romi 32U4 Control Board (ATmega32U4 pinout, peripherals, and board power control).



### Sobre a Pixy:

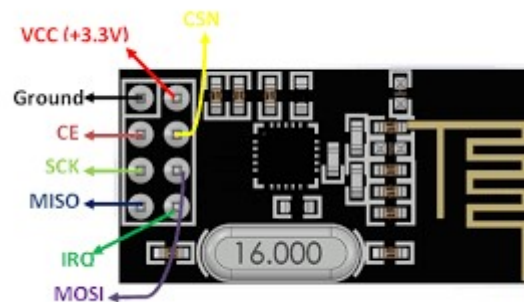
A comunicação da placa Pixy é dada através dos terminais 2 (5v), 10 (gnd), 5 (I2C SCL) e 9 (I2C SDA). A comunicação I2C é um protocolo que possui uma interface de 2 fios (SDA e SCL) que permite a comunicação entre microcontroladores através da interação master e slave. O Pino SCL (Serial Clock Line) é usado para sincronizar os dados entre o master e o slave, onde apenas o master tem controle sobre o clock, assim tendo controle sobre a comunicação com os demais circuitos. Já o pino SDA (Serial Data) envia os dados entre os controladores permitindo a interação mais complexa entre os sistemas.

Relativamente aos dados transmitidos pela Pixy, através do serial monitor lê-se (Block/Sig/X/Y/Width/Height) em que o Bloco define o objeto detectado, Sig ou sinal define o tipo de objeto, mais especificamente a cor, X e Y é a localização no centro do objeto em relação ao campo de visão da câmera que varia de (0,0) até (315,207 e Width/Height (largura/altura) representam as dimensões do objeto.



### Sobre a Comunicação:

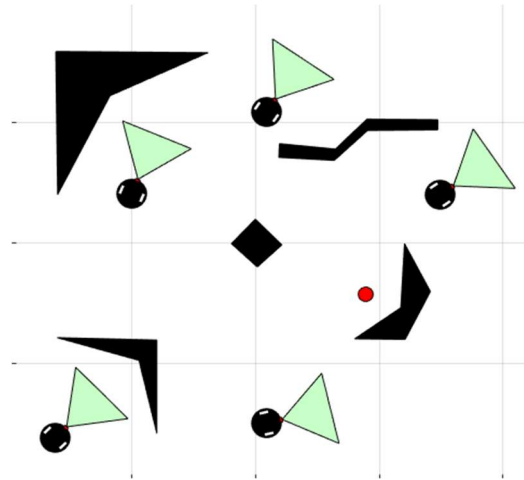
Em termos da comunicação, o modo de funcionamento do projeto é usar o módulo NRF24L01 para transmitir qual o papel de cada robot, líder ou escravo. No caso de ser líder e de ser usar sensores sonares para navegar em sítios desconhecidos, informar os outros robots quando devem virar através do uso das coordenadas X e Y sendo (0;0) o ponto de partida dos robots, para que lado e a quantidade de graus que precisa de virar. E com base na configuração do local onde os robots vão andar e na quantidade destes por swarm, adaptar a formação no qual vão andar. Para se conseguir transmitir estas informações, programa-se a placa Romi3ue através do Arduino (programação baseada em C/C++) com a ajuda da biblioteca nRF24 e do serial monitor o qual permite verificar o sucesso do envio dos parâmetros para que o projeto funcione da melhor maneira possível. A integração do módulo de transmissão à placa Romi é feita através do uso de 7 cabos fêmea-macho os quais vão ligar os VCC, GND, CE, SCN, SCK, MOSI, MISO NRF24L01 à placa.



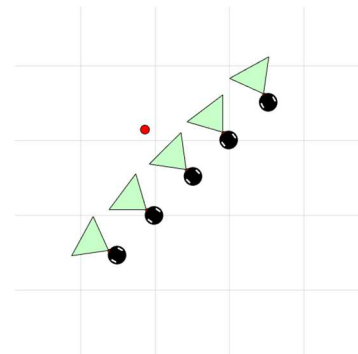
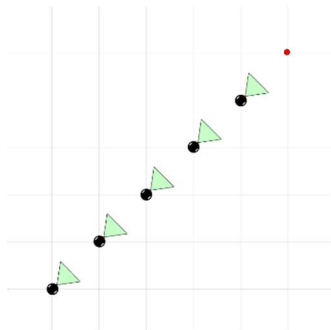


## 5.2 Versão virtual

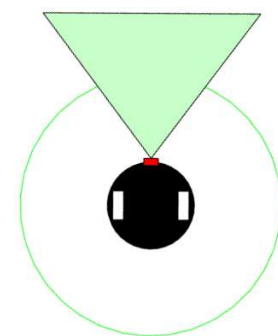
Após a reestruturação do projeto, devido à impossibilidade de execução presencial, tivemos de transacionar o projeto para a dimensão virtual. A simulação será feita em Matlab com auxílio de toolboxes para que possamos criar um ambiente mais semelhante possível ao da realidade. Algumas alterações foram feitas de maneira a favorecer a sua implementação, tais como, a substituição do sensor visual para um sensor de radio uma vez que a cor não importa e a manipulação do objeto a ser seguido é feita através de um controlo remoto (uma aplicação para telemóvel, RoboRemo-Free) a área na qual os sensores irão identificar a bola identificada como o triângulo (verde-claro) e o objetivo que irá ser usado para identificar o master, entre os robots, a vermelho. Os obstáculos os quais o swarm iria encontrar na vida real (paredes, pessoas) estão identificados a preto.



Foram criadas duas formações de swarm. A primeira (série) tem como vantagem a sua agilidade, uma vez que o master define a sua rota os outros robots não tem de se preocupar em desviar de objetos. A segunda (paralelo), uma versão oposta a primeira exige que cada robot constantemente mapeie o seu ambiente e essa torna-se a maior vantagem desta formação, uma vez que ao se deslocar pelo mapa obtém uma maior quantidade de dados sobre o seu ambiente.



Cada Robot conta com um raio de comunicação como na sua versão real e que teria um alcance máximo do seu sinal Bluetooth. De tal maneira que ao identificar o master este envia comandos aos robots em seu alcance e assim sequencialmente entre eles.



Foram também criadas forças de interação entre os robots que permite com que eles não colidam entre si, mas também ao explorar o mapa mantenham-se próximos uns aos outros.



## 6. Atributos

Os principais atributos do nosso robot são os seguintes:

- A movimentação dos robots vai ser coordenada com a movimentação do master, a qual vai ser feita através do ponto objetivo, bola.
- Existe flexibilidade de utilização quando se fala a um nível de quantidade de robots existentes no swarm(dependendo da dimensão do espaço onde andam).
- Usando este modelo de simulação no mundo real, seria usado a câmara Pixi, a qual usaria cores para identificar os objetos desejados, para encontrar a “bola” a qual iria atrás ou se desejado encontrar um objeto com um QR code e lê-lo, em vez dos sensores.

## 7. Manutenção e suporte

### 7.1 Requisitos de manutenção

Os requisitos de manutenção são constantes verificações das partes mecânicas do robot como rodas, motores, sensores e verificar se as pilhas estão carregadas. Futuras correções de código caso este seja corrompido de alguma maneira.

### 7.2 Requisitos de suporte

Os requisitos de suporte necessários são a colocação das devidas pilhas, para que o robot tenha a energia necessária para o funcionamento, utilização das rodas e dos motores especificados de modo a funcionar da melhor maneira possível com o código fornecido.

### 7.3 Requisitos de adaptabilidade

Os requisitos de adaptabilidade foram a utilização de um suporte para a câmara(caso seja aplicado num ambiente real). Alterou-se para uma versão virtual devido à epidemia que ocorreu e num caso de utilização num ambiente real, dependendo do ambiente os robots vão navegar, estes terão de receber novas peças que se adaptam melhor como rodas, motores e outros.

## 8. Período de execução

Em relação ao tempo gasto na realização do projeto, inicialmente estava previsto utilizar as horas relacionadas com a aula prática de Projeto II, 3 horas semanais, e mais horas extras. No entanto, devido ao surto de Covid-19 e a suspensão das aulas presenciais, algumas alterações ao plano foram efetuadas. Deixaram de existir as 3 horas semanais da aula prática, logo para combater este incidente, cada aluno reforçou a análise e a pesquisa em casa, atenuando assim a perda dessas 3 horas semanais. Deste modo a nossa previsão de horas necessárias para a realização do projeto é 120h individuais distribuídas da seguinte forma:

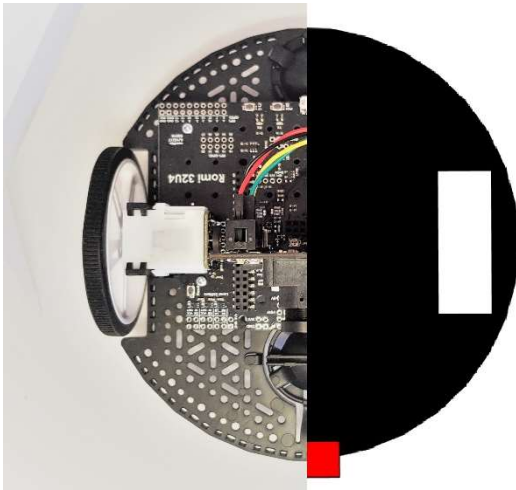
- Montagem inicial do Romi e soldar os pinos necessários e afixação dos sensores (5H)
- Desenvolvimento e implementação da Comunicação (25H)
- Desenvolvimento e implementação da Pixy (25H)
- Programação (40H)
- Desenvolvimento do Website e apresentação (10H)
- Desenvolvimento do Relatório (15H)

Porém foram necessárias horas adicionais dedicadas a pesquisa e implementação nova versão do projeto. Para isso contamos com mais 60 horas dedicadas a criação da simulação.



## 9. Design

Inicialmente o chassi cobriria a superfície do robot deixando apenas uma pequena abertura para a



câmara, esta teria um padrão de cores na backend para que pudéssemos identificar cada robot. Com a mudança para um projeto virtual no matlab decidimos manter a aparência do robot como igual ao chassi. Para que a simulação fosse o mais fiel possível o robot tem como parâmetros de criação as mesmas dimensões que o robot original. Assim ao mover o robot temos de calcular a sua velocidade baseada na velocidade angular das rodas. Como na simulação não vamos utilizar nenhum método de distinção de cores, os sensores foram substituídos por sensores de distância que procuram o objeto de maneira semelhante a sensores sonares. .

## 10. Interfaces

Ambiente virtual:

Neste ambiente usa-se a aplicação RoboRemo-Free como interface para movimentar a nossa “bola”, a qual os robots vão seguir e depois do primeiro robot encontrá-la, irá ser identificado como master. A segunda interface será a nossa comand window do matlab a qual irá nos mostrar as mensagens enviadas entre os robots.

Ambiente real:

As interfaces que vamos utilizar são muito reduzidas devido ao facto de o nosso projeto ser robots que se movem em sintonia uns com os outros, logo só vamos ter duas interfaces: a primeira é um ecrã que irá mostrar qual a cor do robot que é o líder e algumas mensagens; a segunda interface vai ser um led RGB que vai identificar se o robot é o líder ou se está a receber ordens de outro robot.

## 11. Confiabilidade

Devido à falta de condições para a realização dos testes em ambiente real, apenas é possível ver o desempenho do projeto em ambiente virtual o qual, vai depender tanto das características/quantidade dos robots como do mapa criado com os diferentes obstáculos.

## 12. Segurança

Devido o projeto pertencer em um âmbito académico seguranças (criptografia, desligamento automático e deteção de erros) não foram implementadas. Sendo este swarm implementado em ambientes profissionais, estes terão de ter segurança tanto física.



### 13. Ambiente

Com base nas especificações dos componentes que compõem o robot, este projeto é possível ser operado em condições típicas do dia-a-dia. Posto isto, se for usado num sítio com condições mais extremas, como em locais onde existe neve ou bastante precipitação, o robot teria de ser adaptado, com mudanças nos seus componentes em geral, como por exemplo, a adição de uma cobertura para proteger o circuito de qualquer líquido ou a mudança das rodas para umas mais apropriadas para o ambiente onde esteja a ser utilizado.

A nível de perturbações vibratórias, o robot não tem propriamente uma composição robusta, quer seja a nível de componentes, quer seja a nível do chassi do robot. Em condições onde estas perturbações sejam maiores ou que ocorram com maior frequência, a câmara Pixy poderá ter problemas a identificar objetos ou outros robots, logo poderá haver um desgaste nos componentes e no robot, afetando assim o seu desempenho em geral.

A nível de shock loading, depende do tipo de carga que o robot irá transportar, pois se for uma carga instável ou pesada, o robot irá ter algumas dificuldades, com base no seu tamanho e na robustez da sua composição, sendo possível mudar isso com melhores e maiores componentes.

Este robot não é muito resistente a poeira ou a líquidos, como o spray névoa salina, pois estes com o tempo irão acumular nos seus componentes, como por exemplo, nas rodas, perturbando assim o funcionamento. A nível de gases depende do qual for usado.

### 14. Qualidade

Ambiente virtual:

Neste ambiente é difícil de quantificar a qualidade do produto, mas com a utilização de funções como mapgenerator a qual cria , o mais próximo possível, uma versão do ambiente no qual os robots vão operar testando assim o seu desempenho num ambiente real, sendo assim possível ver se as necessidades do cliente vão se alinhar com o funcionamento deste swarm.

Ambiente real:

Pode-se dizer que em comparação ao custo da produção do robot, a qualidade que se pode esperar será elevada, pois com a configuração correta este poderá ter bastantes aplicações, as quais podem variar entre clientes, dependendo das necessidades.

Em caso de avaria ou de mau funcionamento na parte física ou computacional do robot, este irá ser revisto de forma a poder corresponder às expectativas do utilizador.

Para garantir que a experiência do cliente com o produto seja boa, irá adquirir-se peças em excesso, para substituição no caso de avaria de qualquer componente integral para o funcionamento do projeto. No entanto, para prevenir estes acontecimentos, os produtos serão submetidos a testes antes de serem integrados nos robots, garantido assim um bom desempenho para o funcionamento deste.

No caso de se querer adicionar componentes aos robots, os quais não teriam sido requisitados pelo cliente anteriormente, irá ser discutido a melhor forma possível para ser implementado.

As qualidades métricas que precisam de acompanhamento ao longo do uso do robot serão a distância que irá percorrer, que afeta os componentes eletrónicos, como os motores das rodas e essas mesmas rodas; e o tempo de uso do robot, o qual vai influenciar o desempenho deste.

### 15. Testes

Como o projeto foi feito num ambiente virtual, foi possível testar o funcionamento do swarm com todo o tipo de variáveis, tais como diferentes velocidades angulares, tamanhos de rodas e velocidades de cada robot. Com estes diferentes testes optimizou-se assim o projeto, ignorando de certa maneira, as limitações das partes que iriam compor o swarm, inicialmente planeado para fazer nesta cadeira.



## 16. Referencias

<https://www.pololu.com/docs/0J69/5.2>

[https://www.pololu.com/docs/pdf/0J69/romi\\_32u4\\_control\\_board.pdf](https://www.pololu.com/docs/pdf/0J69/romi_32u4_control_board.pdf)

[https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:hooking\\_up\\_pixy\\_to\\_a\\_microcontroller\\_-\\_28like\\_an\\_arduino-29](https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:hooking_up_pixy_to_a_microcontroller_-_28like_an_arduino-29)

<https://www.arduino.cc/en/reference/SPI>

<http://maniacbug.github.io/RF24/classRF24.html#a30a2733a3889bdc331fe2d2f4f0f7b39>

<https://www.pololu.com/docs/0J68/4>

<https://www.pololu.com/docs/0J69/4>

[https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:porting\\_guide&s\[\]=pixy&s\[\]=pins](https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:porting_guide&s[]=pixy&s[]=pins)

<https://en.wikipedia.org/wiki/I%C2%B2C>

[https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:ccc\\_api](https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:ccc_api)

<https://www.pololu.com/docs/0J69>