



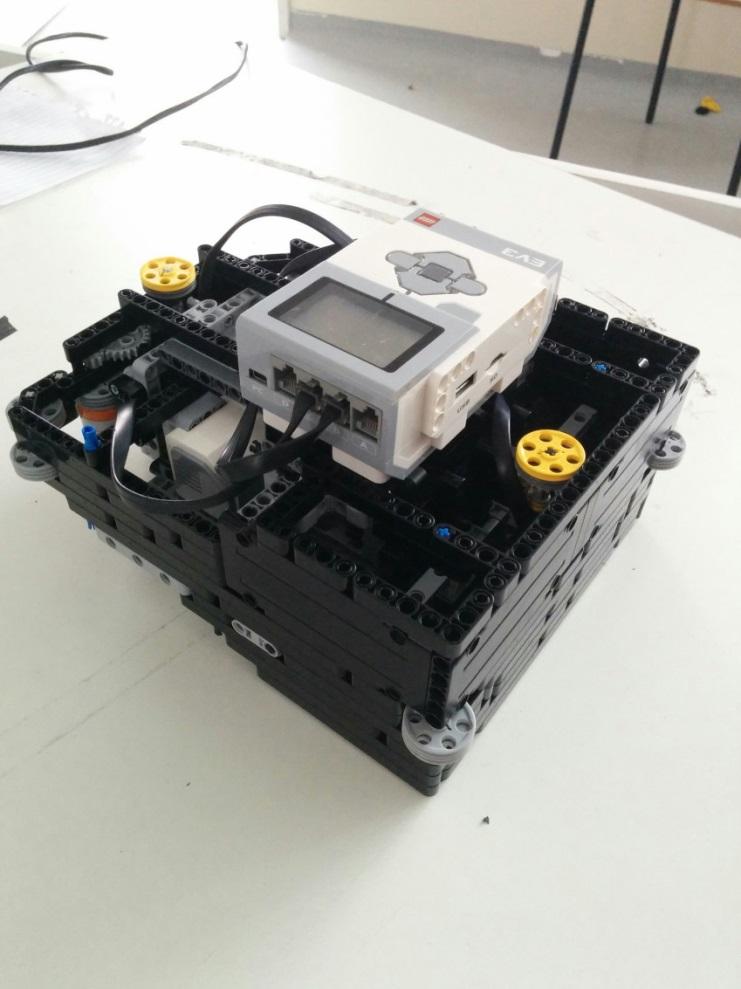
Rodolfo



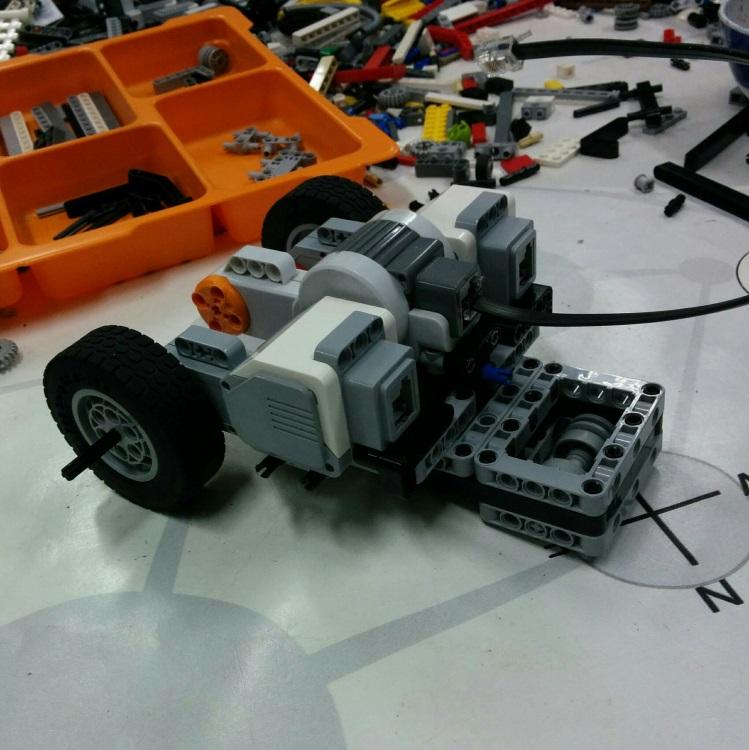
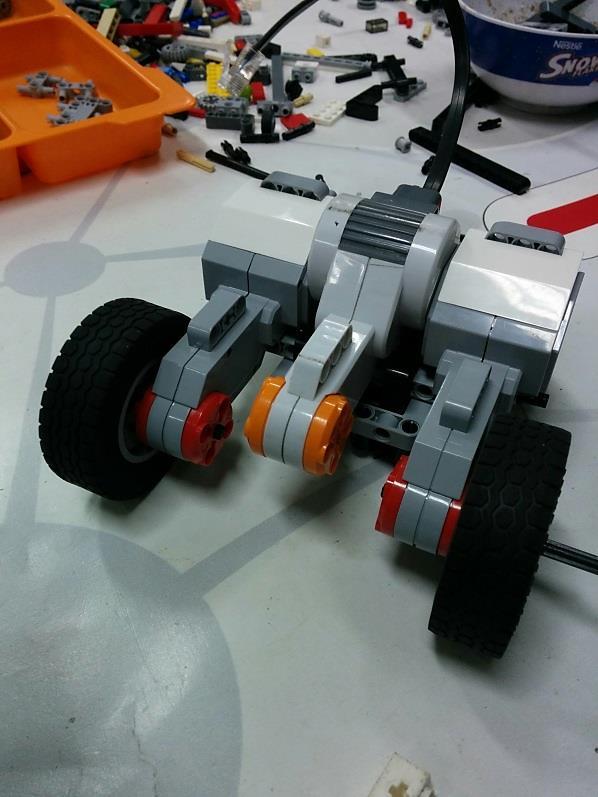
Evolução do robô

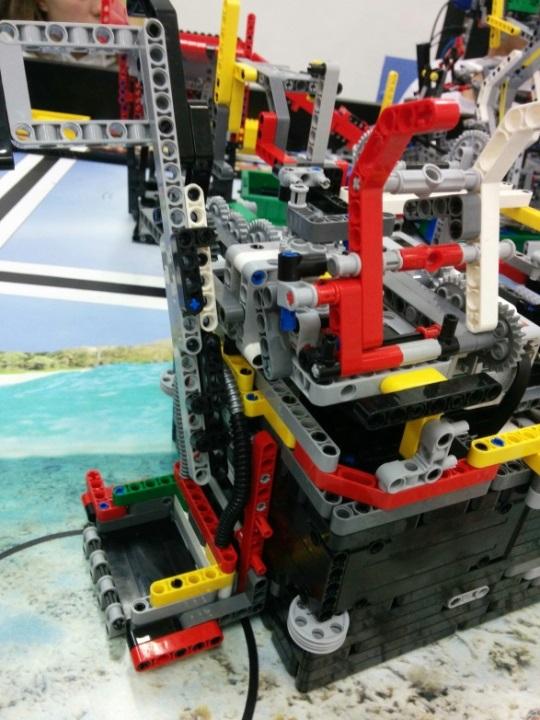
A construção de **Rodolfo** (nome do robô) foi feita com base em **estudos**, **testes** e toda nossa experiencia adquirida em diversos anos de FLL. Nosso robô possui **quatro motores**, sendo dois do EV3 e dois do NXT, na qual os do EV3 estão sendo usados para a tração do robô e os do NXT estão sendo usados para equipamentos. Depois de diversos testes, optamos por usar a **tração dianteira**, pois assim conseguimos fazer curvas de um mesmo ângulo ocupando um espaço menor da mesa. Nós posicionamos os dois motores de **equipamento estrategicamente**, um fica na traseira e outro na dianteira do robô. Nos dois motores dos equipamentos nós utilizamos um **sistema de polia** para o encaixe dos equipamentos, pois assim conseguimos um rápido e fácil encaixe sem perder o torque e a velocidade. Usamos a roda 32019 da Lego, pois essa roda tem um diâmetro bom, assim conseguimos uma média boa de **velocidade** sem forçar o motor, e também pelo fato de nosso robô ser pesado, precisando de **torque**, e com os tamanhos dessa roda conseguimos **compensar a baixa velocidade** do torque, mas sem ter que forçar muito o motor, pois quanto maior o diâmetro da roda, maior a distancia que ele ira percorrer num mesmo giro do motor. Outra qualidade desta roda é o fato dela ser **rígida**, ou seja, por mais pesado que o robô seja, ela não ira se deformar, e ainda adicionamos um tubo de Lego dentro de cada roda, para diminuir mais a chance da roda se deformar, assim conseguindo sempre fazer toda a área de contato da roda tocar no tapete. Usamos 2 polias em cada ponta do robô, pois assim conseguimos se ajeitar na parede com maior **facilidade**, e quanto estivermos andando encostado na parede, diminuímos a área de contato do robô com a parede, assim diminuindo o atrito que ira dificultar o robô.

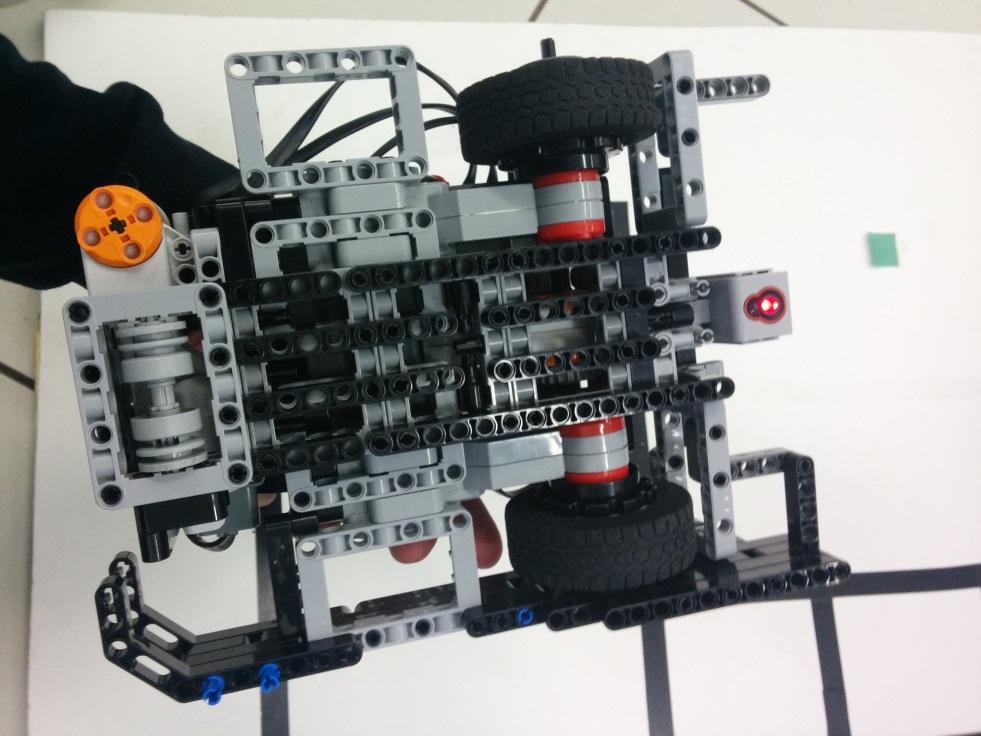
Usufruímos de 2 **sensores de cor** do EV3 na dianteira do robô, para conseguirmos seguir a linha ou simplesmente encontrar alguma linha, para diminuir ao máximo a chance de erro, deixando o robô o mais **preciso** possível. Também utilizamos 2 rodas Lego 20896 sem a borracha na parte traseira do robô como apoio, pois depois de diversos testes descobrimos que este é o melhor apoio, pois elas conseguem oferecer **pouco atrito** ao tapete mas ao mesmo tempo uma estão muito bem presas, fazendo assim uma roda de apoio de ótima qualidade tanto para retas quanto para curvas. E tudo isso dito acima, esta muito bem preso através de vigas presas em um sistema de **treliças**, presas no chassi e na carenagem do robô, assim tendo um robô **robusto** e preciso feito de através de uma **construção sólida** e **sem reparos**.



**Foto do nosso primeiro robô que era grande demais.**

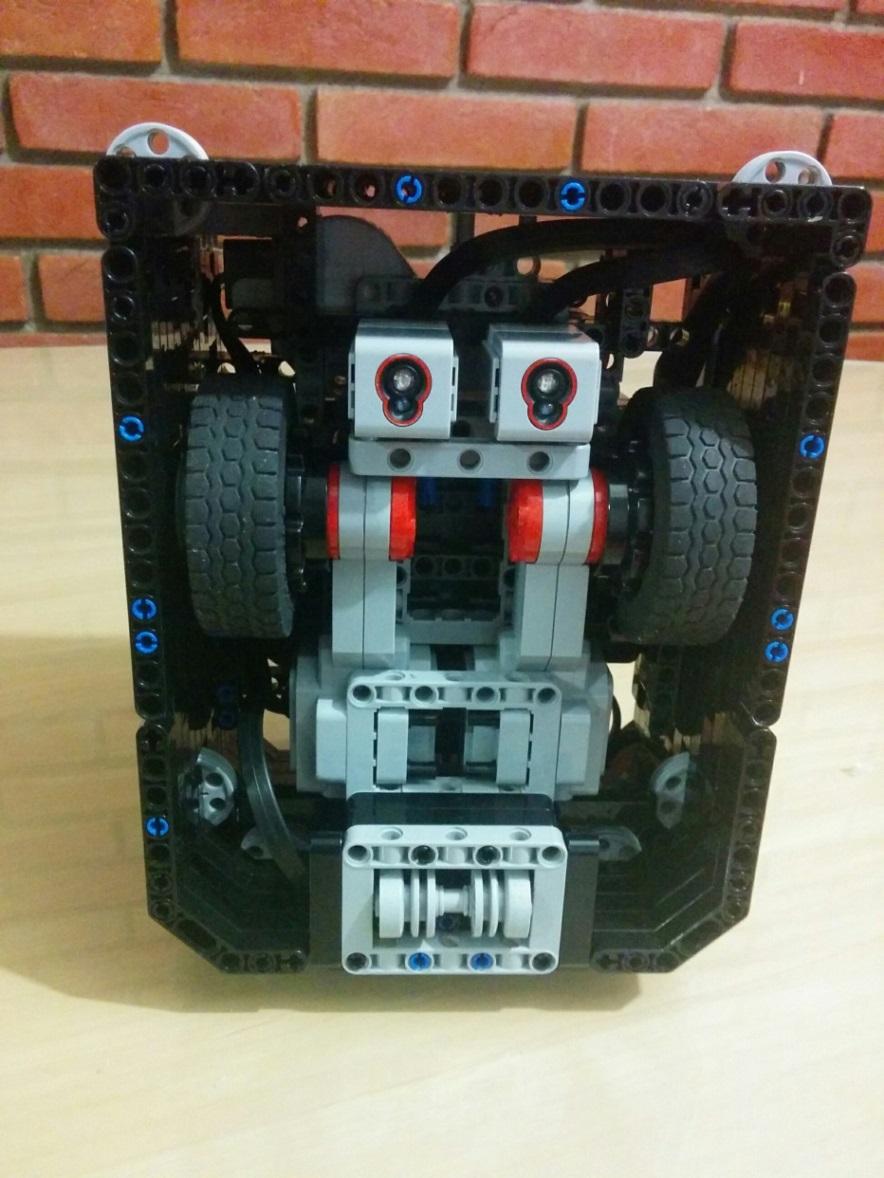
**Fotos do Chassi do robô antigo onde o entre-eixos era maior**

**Foto do robô antigo com uma garra pronta, esse robô mantinha as mesmas idéias e estratégias do robô atual, só que decidimos compactar o máximo possível.**

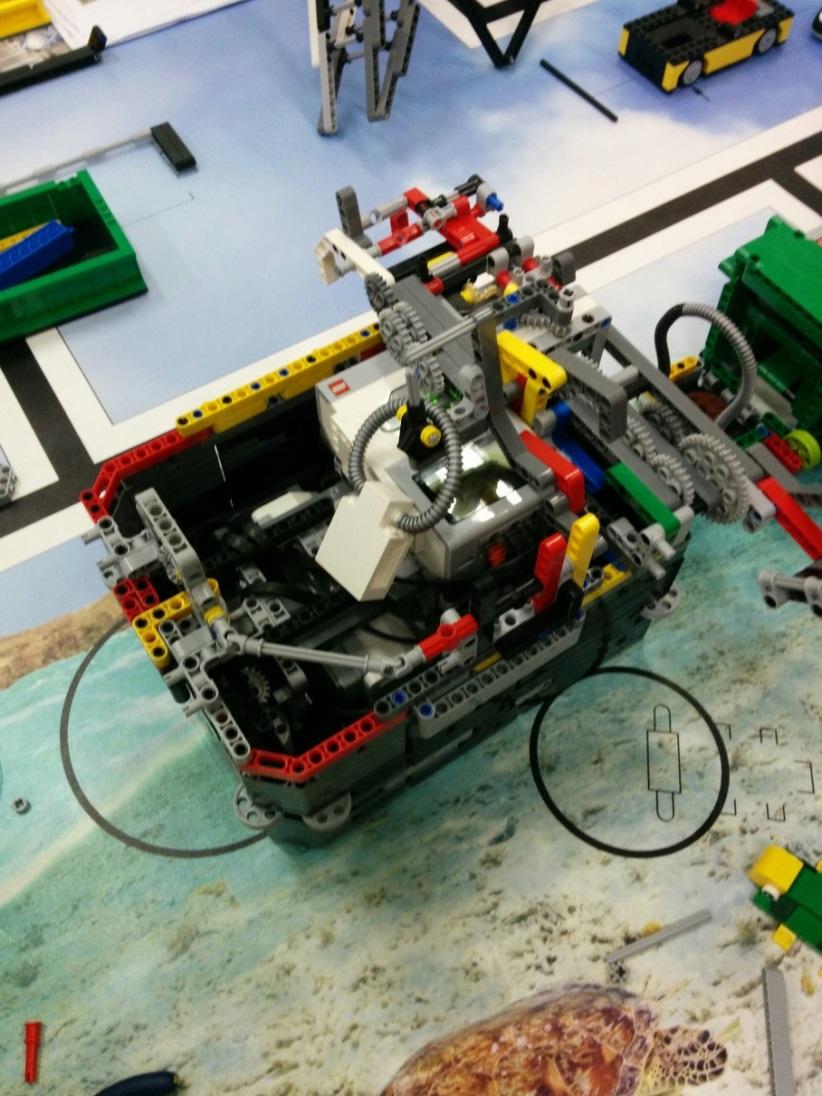




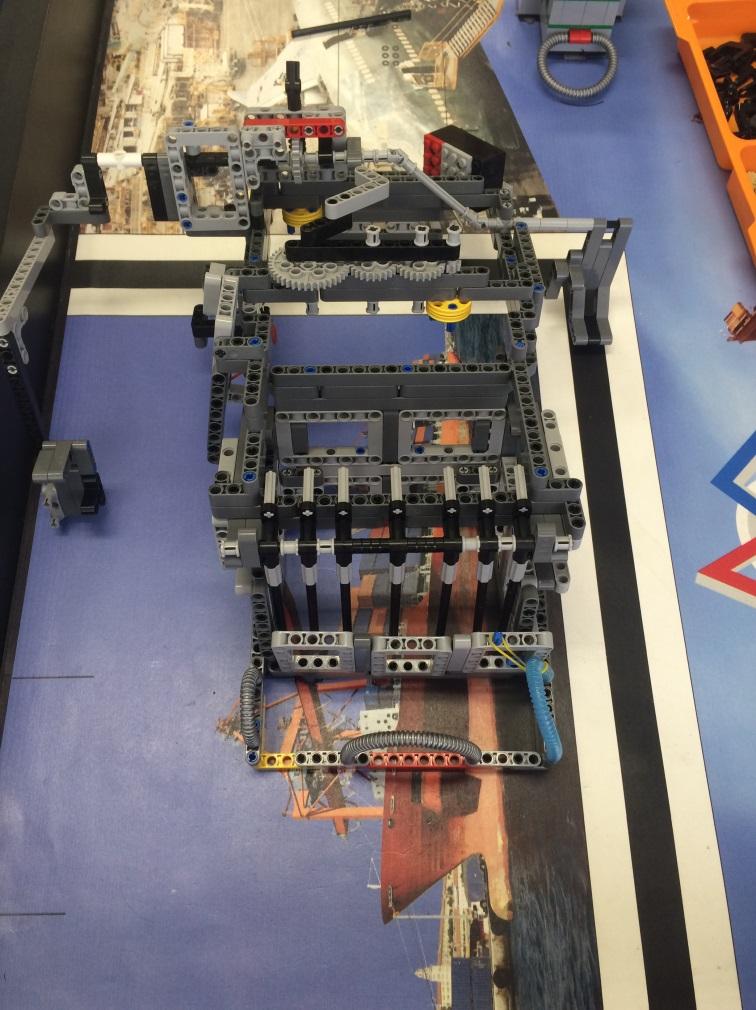
Fotos do robô atual



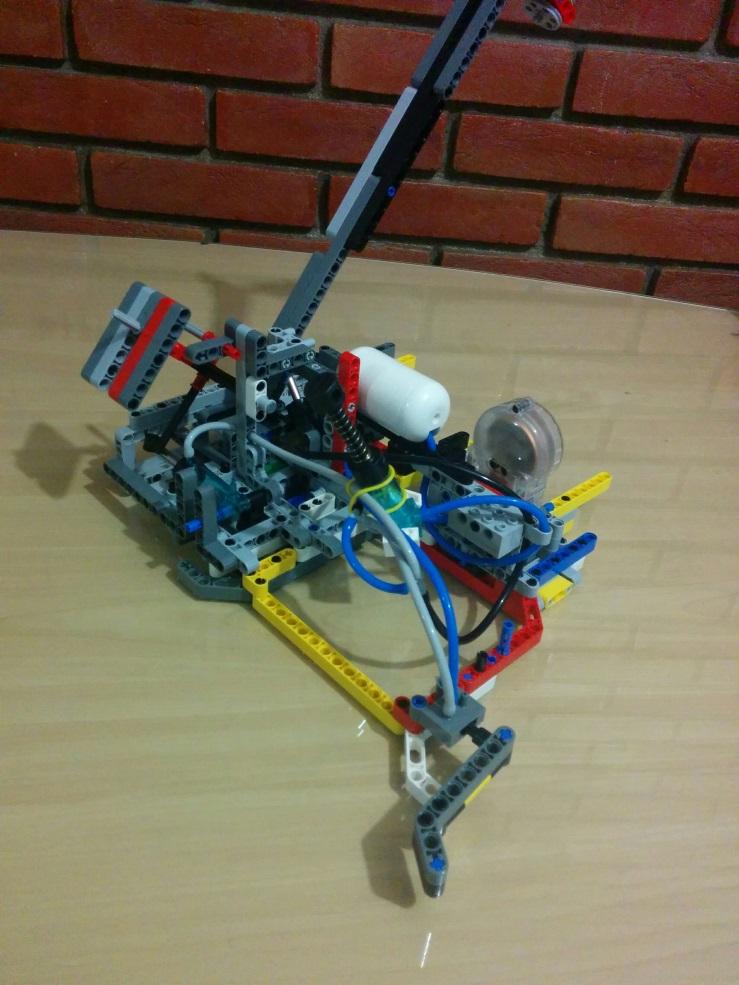
Evolução dos equipamentos



**Equipamento antigo**

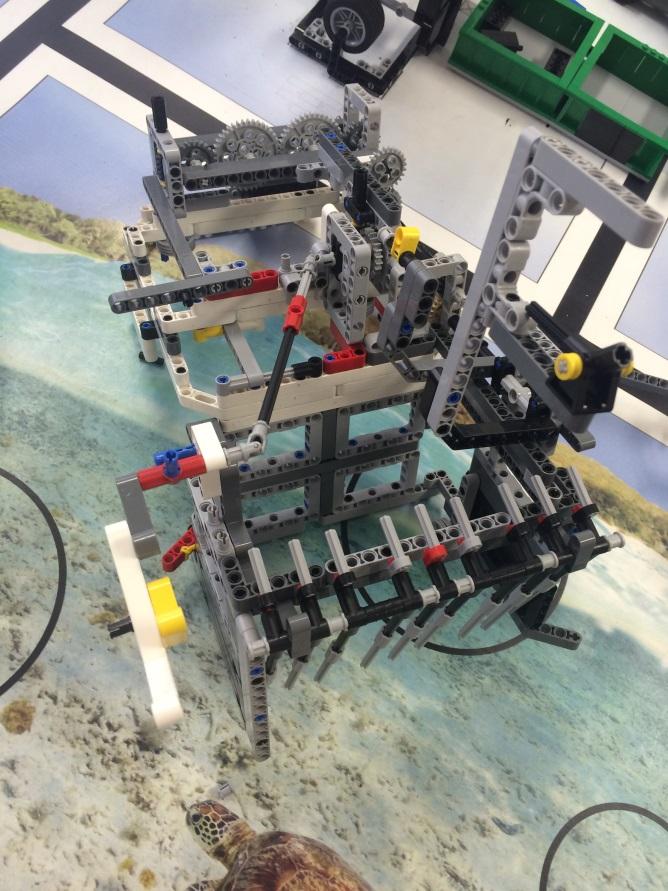


**Equipamento Novo**

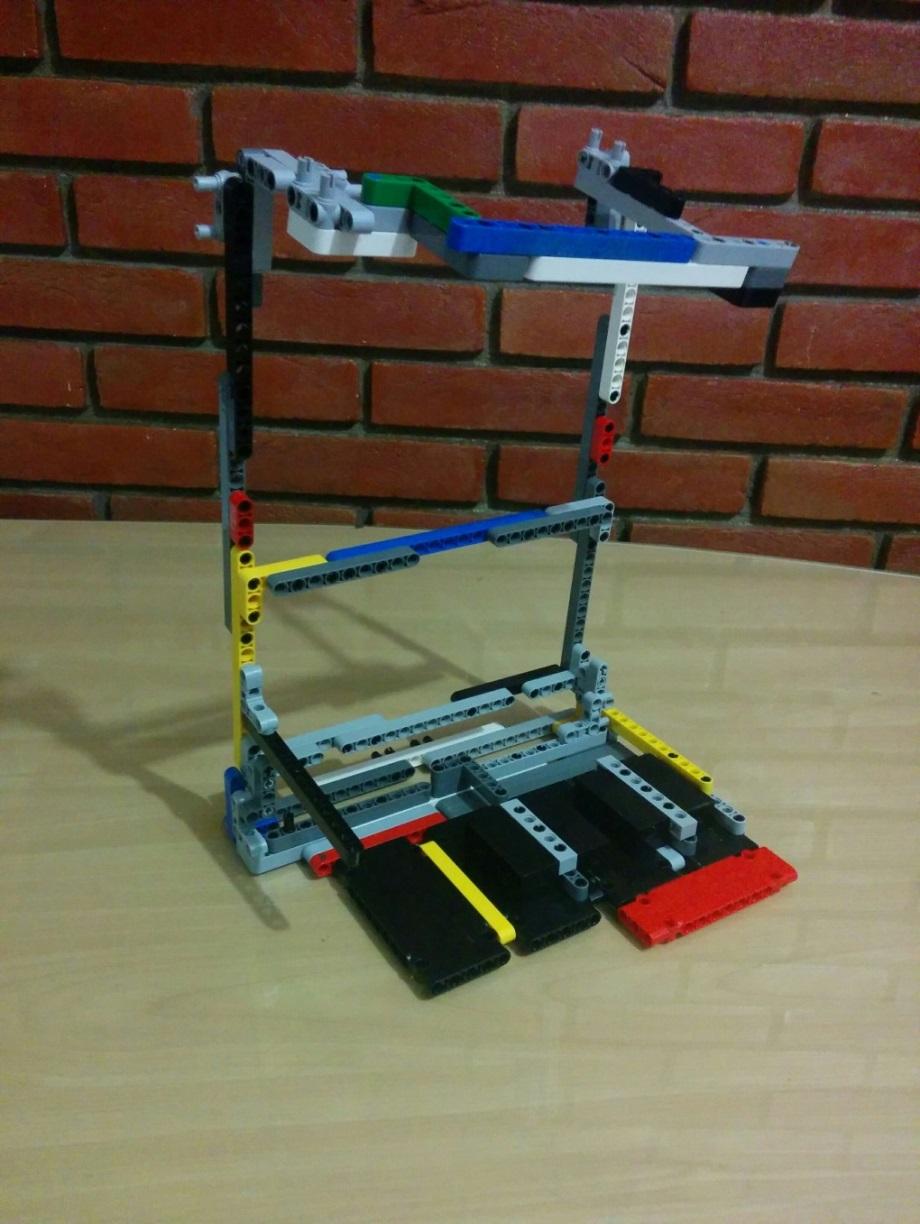
**Equipamento antigo**



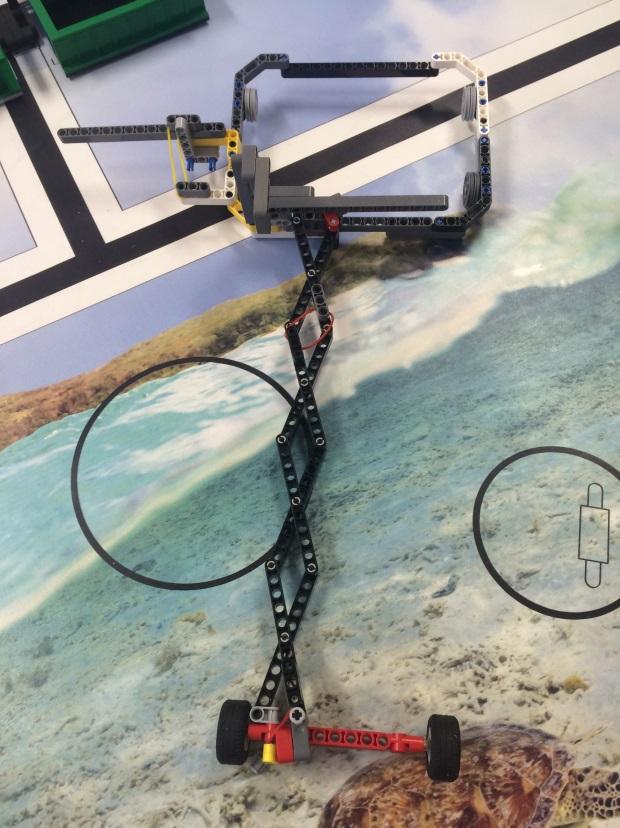
**Equipamento novo**



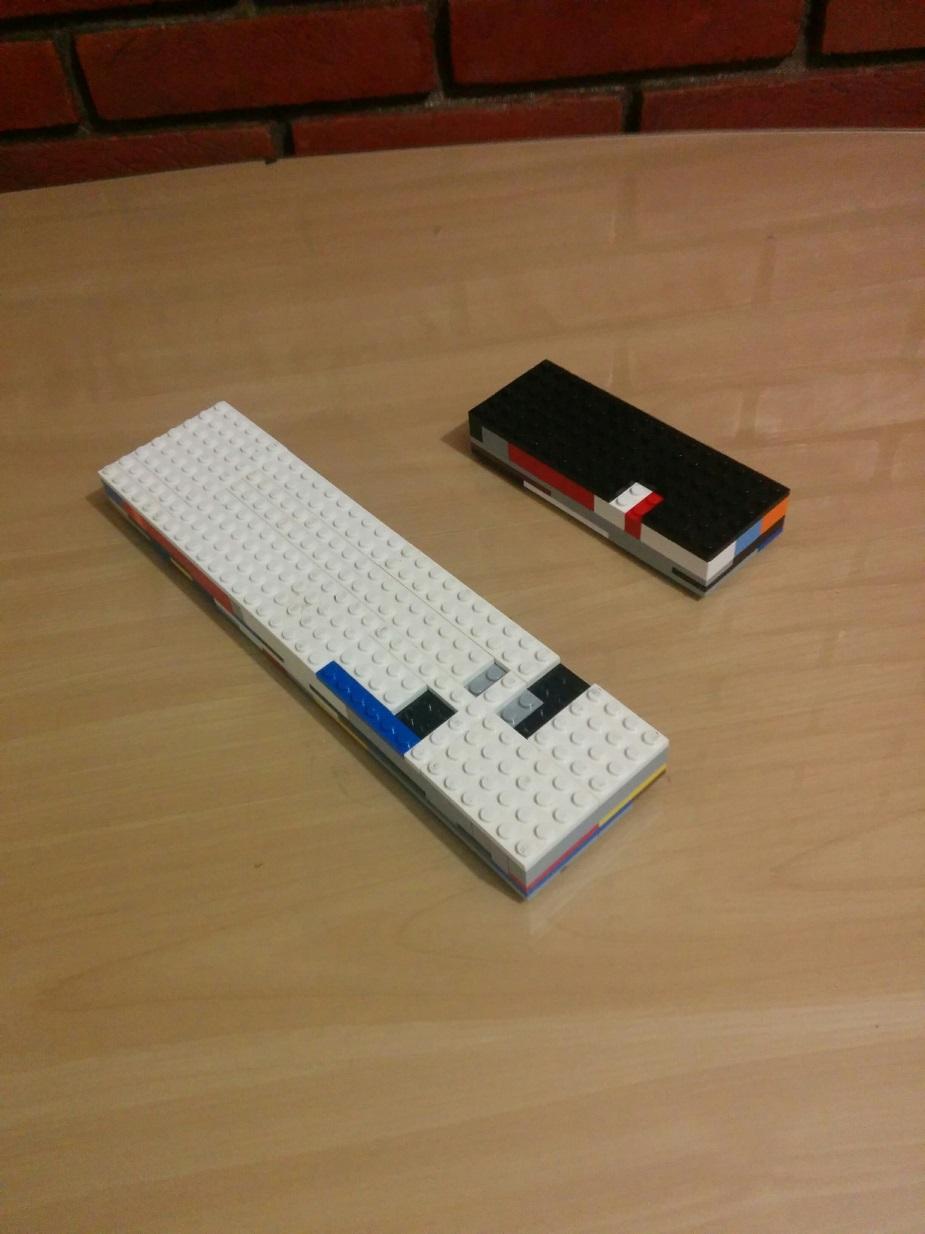
**Garrarito antigo**



**Garrarito novo**



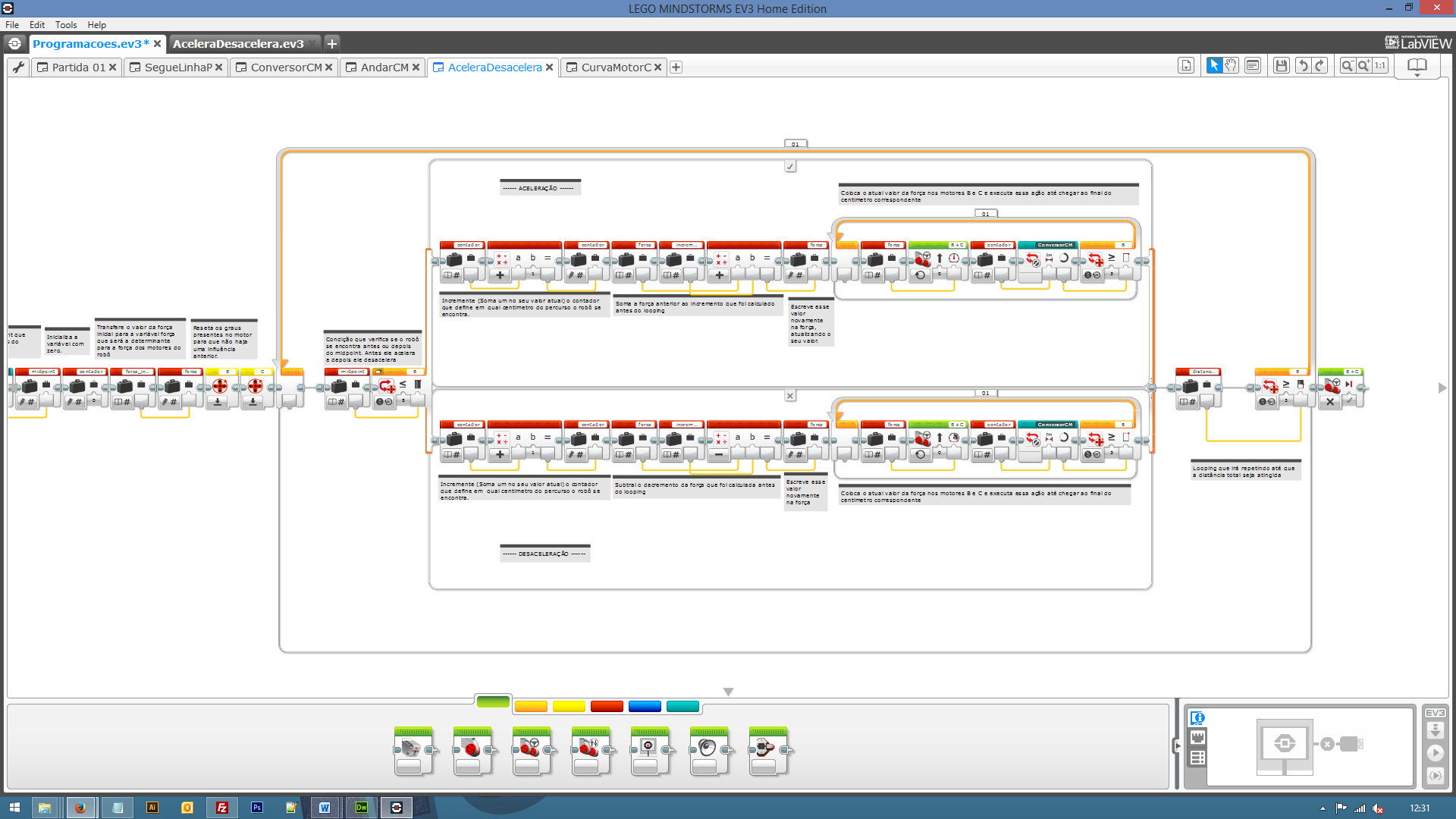
**Gabarito antigo**



**Gabarito novo**

Programações

Aceleração e Desaceleração em CM

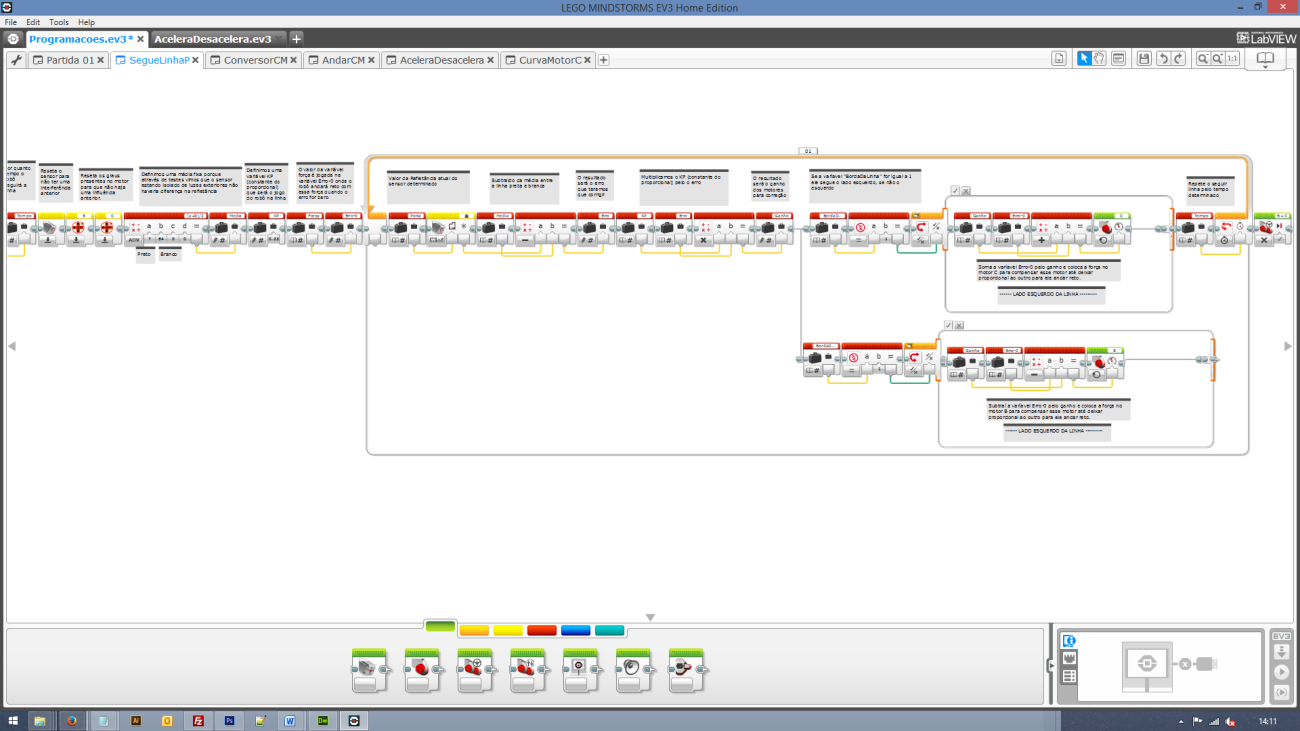


**Duas entradas de dados:** Força Máxima e Distância em CM.

**Outras váriaveis:** Midpoint (Distância da metade do caminho total), Força inicial e Incremento ((Força maxima – Força Inicial) / Midpoint)

**Lógica:** O robô calcula o incremento com base na força máxima e distância até o midpoint. O robô tem que chegar no Midpoint (Metade do caminho) com a força máxima que determinamos e para isso, a cada centimetro contado na variável “contador” ele vai adicionando na força atual do robô até chegar no Midpoint, depois ele faz o mesmo processo só que inversamente desacelerando.

**Utilidade:** Essa programação serve para não desastabilizar o robô quando ele for sair de uma posição de repouso para não chegar em uma posição de repouso muito rápidamente.

Seguidor de Linha Proporcional com os dois lados da linha

**Quatro entadas de dados:** Força, Borda da Linha, Porta e Tempo;

**Outras varíaveis:** Média e KP.

**Lógica:** Logo depois que determinamos as váraveis do nosso seguidor de linha proporcional, o robô começará a fazer o algoritmo de seguir linha pela equação:

Luminosidade Atual – Média = Erro (Variável que determina o erro que o robô tem que corrigir para se alinhar na linha)

KP \* Erro = Ganho (Quanto de força ele jogará nos motores para se alinhar)

LADO ESQUERDO DA LINHA:

Ganho + Força (Erro-0) = Força do motor B

Ganho – Força(Erro-0) = Força do motor C

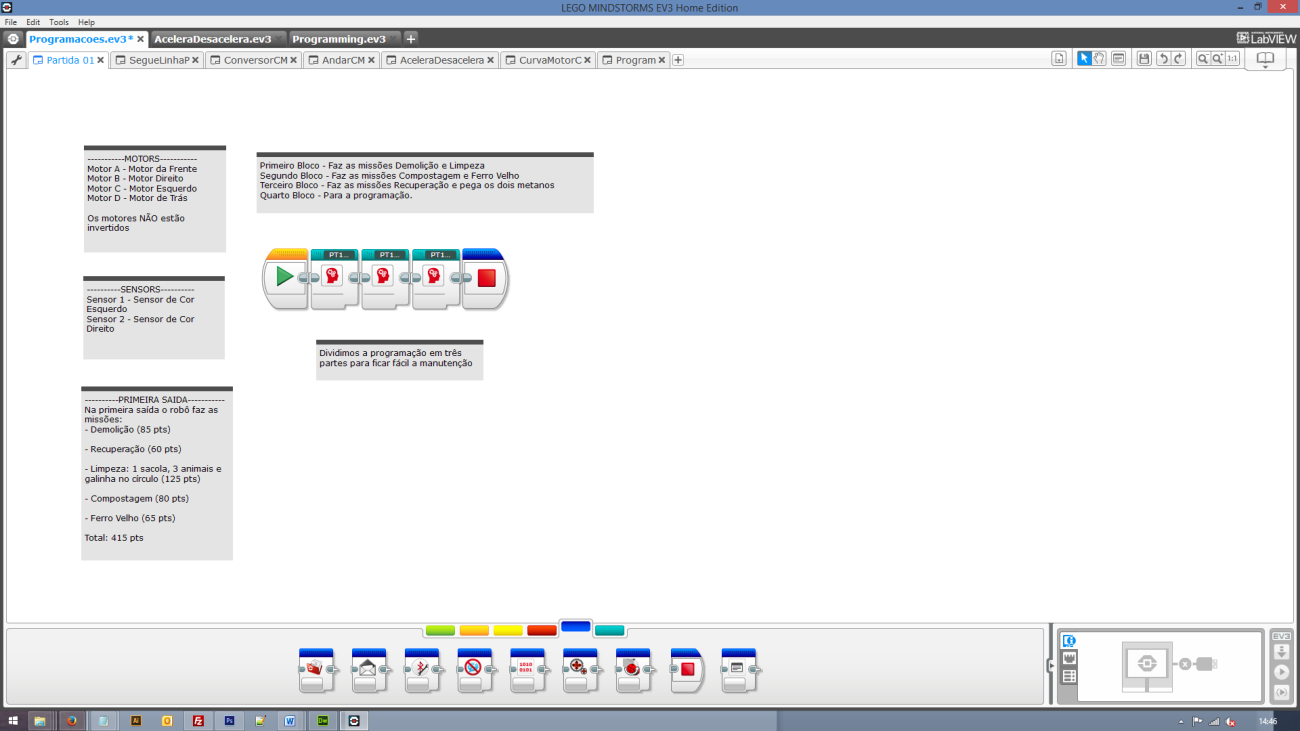
LADO DIREITO DA LINHA:

Ganho + Força (Erro-0) = Força do motor C

Ganho – Força(Erro-0) = Força do motor B

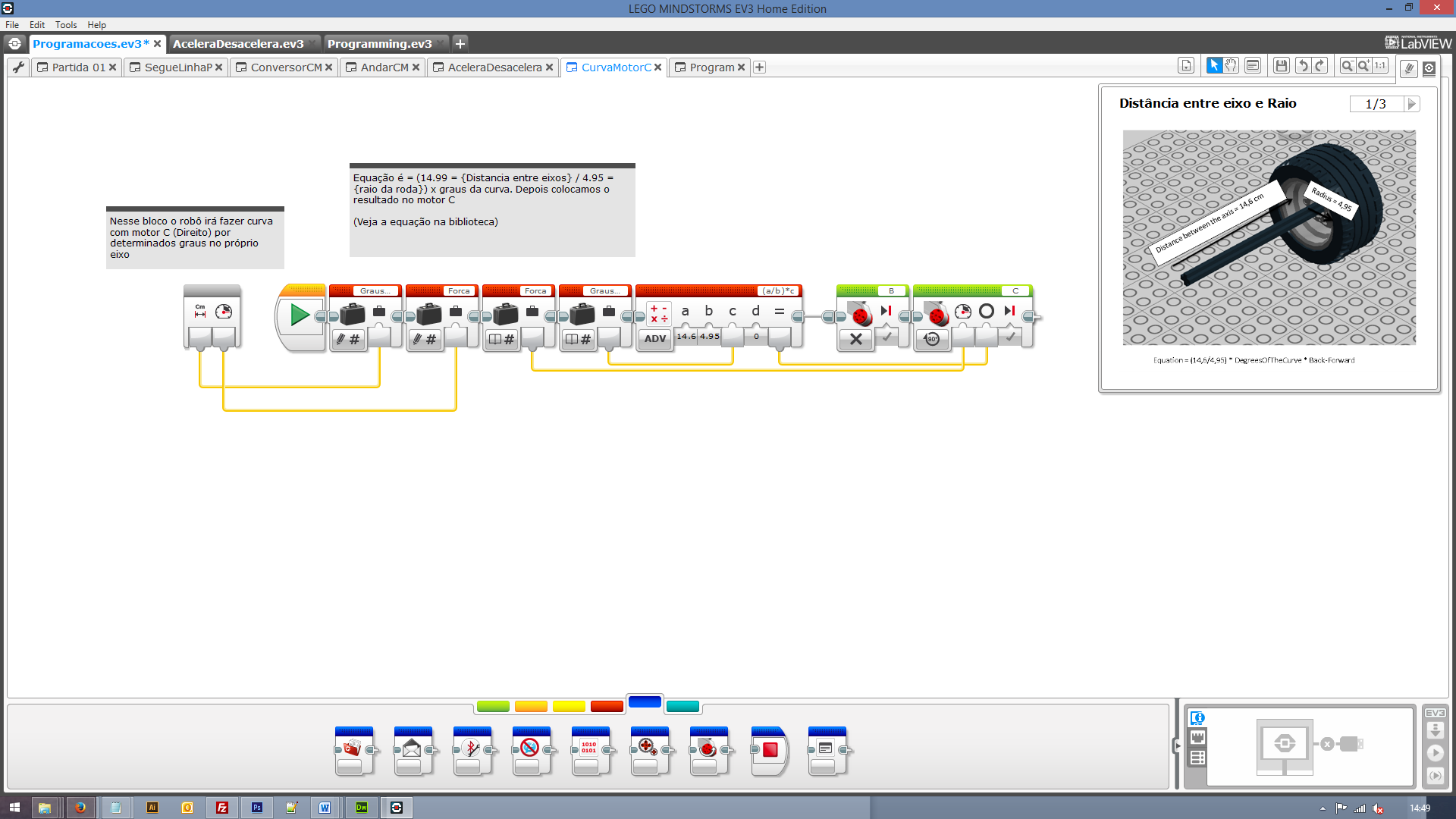
Ele repetirá esse processo por um tempo determinado para que se alinhe na linha e zere totalmente o erro.

**Utilidade:** Essa programação é necessária para dar precisão do robô zerando o erro através do seguidor de linha. Podemos seguir o lado esquerdo ou direito da linha.

Programação modulada e dividida

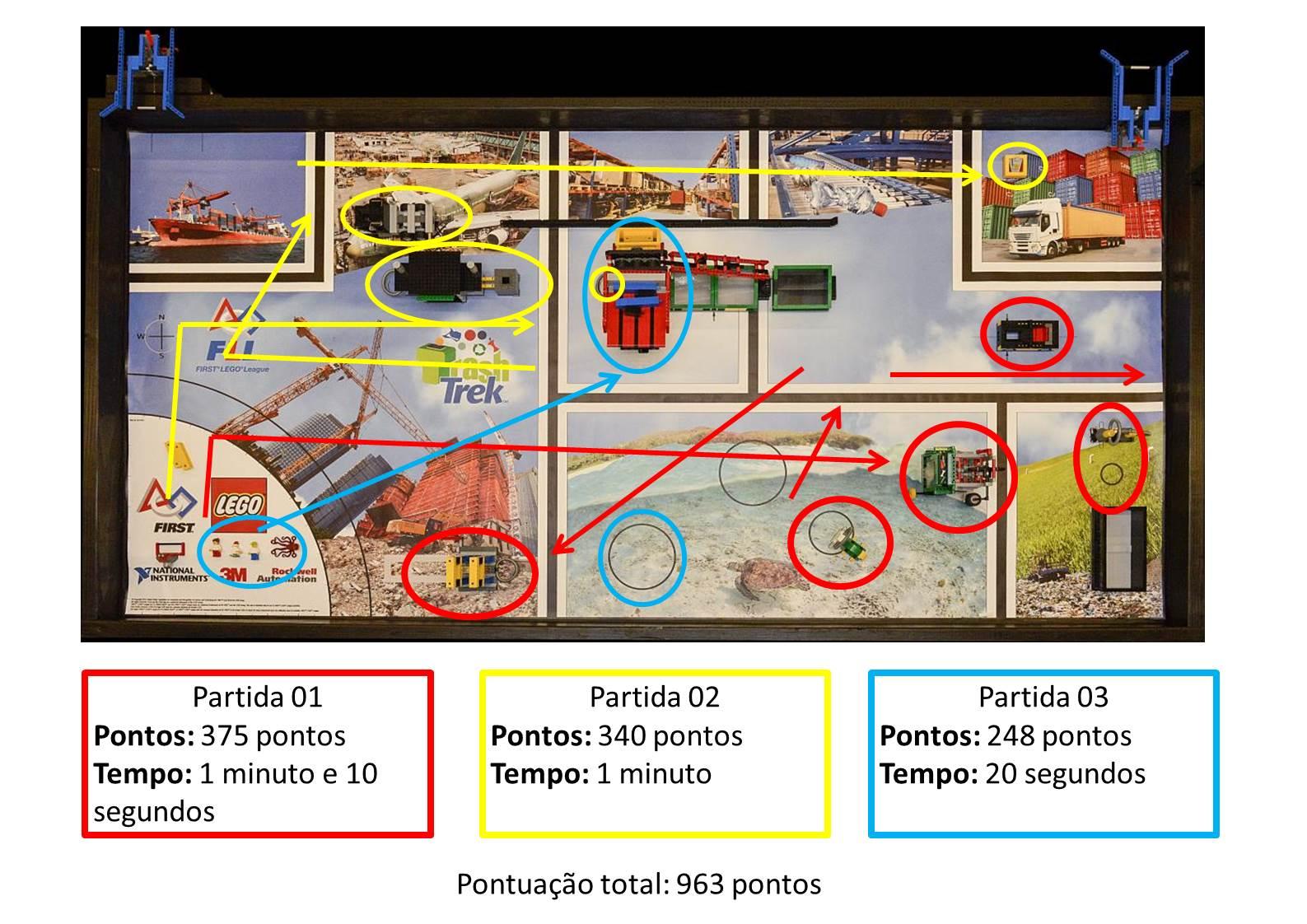
**Utilidade:** Dividimos em partes para ficar fácil de fazer qualquer alteração em todos os lugares da programação. Ela também fica mais compacta e organizada e você sabe em que setor você está cada missão e trajeto.

Curva no próprio eixo



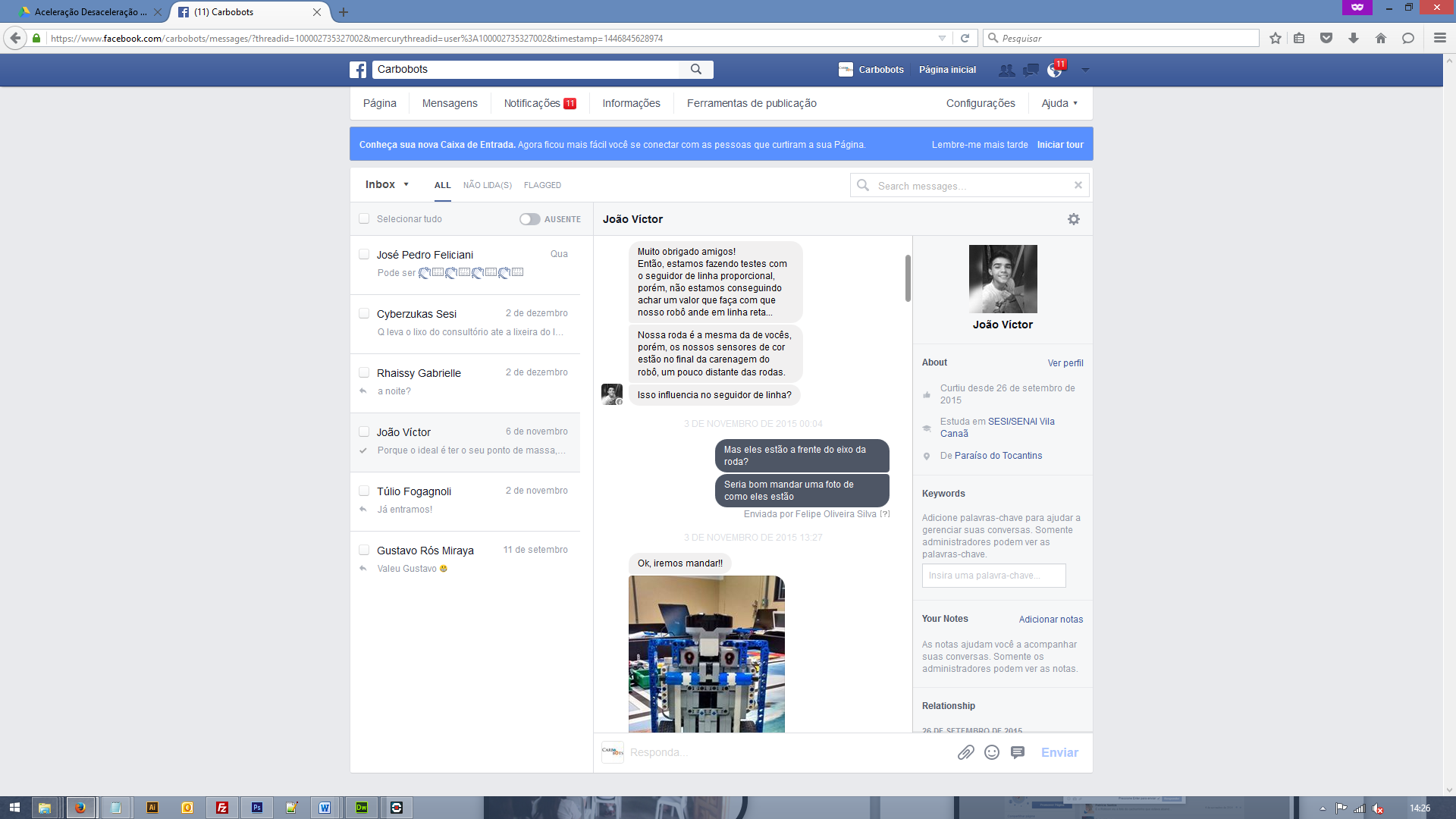
**Utilidade:** Essa programação utiliza de uma equação para que o robô faça uma curva no proprio eixo. Medimos a distância entre eixo e o raio da roda para fazer a equação:  
(Entre eixo – Raio da roda) \* Graus da Curva = Motor C ou B

Estratégia das saídas



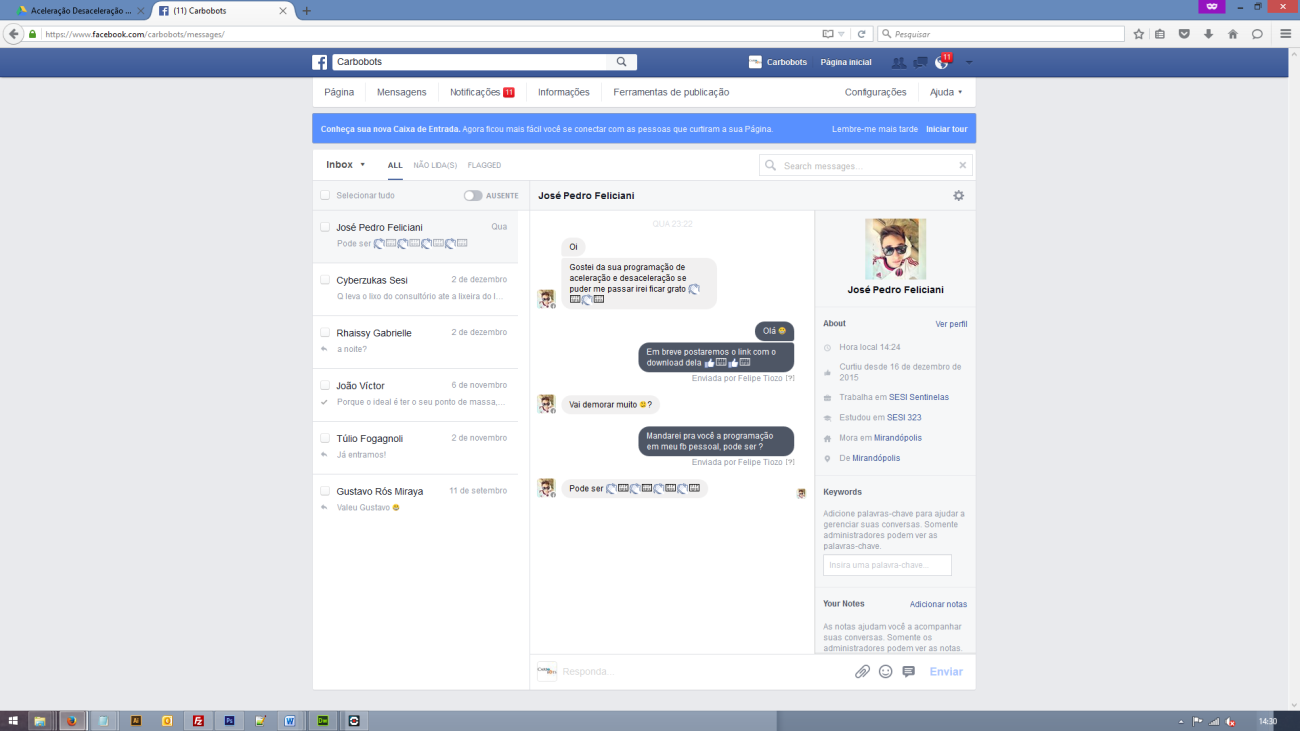
Compartilhamento

Conversamos com outras equipes para trocar nossas experiências de robô, programação e estratégias, além de criar novas amizades! Print da conversa do membro João Vitor do time Gametech Canaã - GO



Participamos de um grupo de programadores chamado “Think FLL” onde trocamos experiências, criamos programações e compartilhamos com as outras equipes.



Compartilhando a programação “Aceleração e Desaceleração” com o João Pedro Feliciani do time Sentinelas - SP

Visitamos nossos amigos do time AC / DC / EG de São Caetano do Sul (SP) para trocarmos experiências e fazer uma parceria! Além de um time se unir ao outro, nos tornamos amigos!