

RobôFun: Lições Aprendidas em um Projeto de Atração de Novos Estudantes e de Redução da Evasão

Luan C. Klein, Julia Z. Cordeiro, Guilherme G. Barboza, Cesar A. Tacla

PET Engenharia de Computação (PETECO)
Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná – Curitiba – PR - Brasil

{luanklein,juliacordeiro,guilhermearboza}@alunos.utfpr.edu.br,
tacla@utfpr.edu.br

Abstract. *This article presents the RobôFun project with its double goal: i) to allow students of the Computer Engineering PET group to carry out research, teaching, and extension activities under the umbrella of one project, and ii) as a mean to attract new students to computer engineering, computer science, and alike courses as well as to reduce the freshmen evasion from these courses. The group has developed a robot able to be programmed using a visual language for people without previous knowledge on programming and robotics. The apprentice can command the real robot or simulate its behavior. The results are presented in the form of lessons learned observing aspects like student motivation, learning, and management of the activities for both the design of the robot and the teaching activities.*

Resumo. *Este artigo apresenta o projeto RobôFun e seus objetivos: i) permitir que os estudantes do Grupo PET realizem atividades de pesquisa, ensino e extensão sob um único projeto, e ii) ser um meio de atrair estudantes para os cursos de engenharia de computação, ciência da computação e afins, e amenizar a evasão de calouros desses cursos. O grupo desenvolveu um robô que pode ser programado em uma linguagem visual para pessoas sem conhecimento prévio de programação e robótica. O aprendiz pode comandar o robô real ou simular seu comportamento. Os resultados do projeto são apresentados sob a forma de lições aprendidas observando aspectos como a motivação do estudante, a aprendizagem e a gestão das atividades tanto para o projeto do robô como para as atividades de ensino.*

1. Introdução

O projeto RobôFun, iniciado em 2012, consiste no desenvolvimento de um robô móvel de baixo custo¹, dotado de sensores básicos e de uma interface de programação para usuários leigos em programação e robótica. De forma mais abrangente, o projeto tem por objetivo permitir ao grupo PET aprimorar seus conhecimentos em conteúdos técnicos e metodológicos relacionados ao projeto do hardware e do software de controle do robô e a uma linguagem e interface de programação de alto-nível. Neste projeto, também se trabalham conhecimentos ligados ao ensino e à gestão de projetos aprimorando habilidades sociais (e.g. trabalho em grupo e comunicação) e habilidades de planejamento e execução de tarefas (e.g. metas, acompanhamento e avaliação). Como atividade

¹ em torno de R\$300,00 (out./2020)

extensionista, o projeto ambiciona atrair estudantes do ensino médio para as áreas de computação e engenharia. Busca-se também amenizar o problema de evasão dos recém-ingressos nos cursos de Engenharia de Computação, Bacharelado em Sistemas de Informação e afins.

[Barroso e Falcão 2004], ao fazer hipóteses sobre as possíveis causas de evasão do curso de Física/UFRJ, classificam-nas em socioeconômicas, vocacionais e institucionais. As socioeconômicas estão relacionadas ao conflito do estudante conseguir se manter em cursos que exigem bastante dedicação e a necessidade de trabalhar. As vocacionais se referem à falta de entendimento da profissão, dos conteúdos dos cursos e dos percursos da grade curricular. As causas institucionais dizem respeito às dificuldades da instituição em acolher o estudante. Engloba dificuldade de compreensão das disciplinas (iniciais), falta de integração entre estudante e instituição, estudante e colegas e dificuldades de estudo, seja pela deficiência em conteúdos do ensino fundamental/médio ou em metodologia de estudo. Nos cursos de computação, os estudantes relatam dificuldades nas disciplinas introdutórias como programação, algoritmos, lógica e cálculo. Todas estão ligadas ao desenvolvimento do pensamento computacional que abarca, de forma não exaustiva, representação e raciocínio com símbolos abstratos, raciocínio procedimental, combinatória e modelagem. A atração de novos estudantes para as áreas de computação pode ser vista como o reverso da evasão em relação à causa vocacional. Evasão e atração são os dois problemas abordados pelo projeto.

Este artigo tem por objetivo apresentar os conceitos fundamentais do projeto e como se articulam para instigar nos membros do grupo PET o gosto pela pesquisa e pelo ensino, influenciar na atração de novos estudantes do ensino médio de forma honesta e amenizar a evasão dos recém-ingressos dos cursos de graduação. O artigo inicia com apresentação dos pilares que sustentam o projeto: uso de artefatos concretos e ambientes simulados para estímulo do pensamento computacional e para motivação do aprendizado. Em seguida, apresenta-se a abordagem adotada para desenvolvimento dos robôs e para concepção e realização das oficinas. Finalmente, apresentam-se os resultados obtidos e as conclusões. Os resultados estão na forma de lições aprendidas advindas do projeto de desenvolvimento do robô e com base em coletas de dados de questionários das oficinas ministradas. Ressalta-se que o projeto aborda mais fortemente as questões vocacional e institucional.

2. Fundamentação e Trabalhos Relacionados

O pilar básico do projeto consiste em motivar os estudantes por meio do ensino/aprendizado contextualizado por meio da utilização de artefatos concretos (robô) e introdução ao pensamento computacional de forma intuitiva. Pelo uso desses elementos, mostra-se aos estudantes aplicações de conteúdos abordados nas disciplinas de computação ao mesmo tempo que se demonstra o uso/functionamento de tecnologias que são objeto dos cursos de graduação (para fins de atração).

O ensino da programação de computadores e de robótica tem se tornado algo comum e importante na sociedade contemporânea. Segundo [Wing 2006][Resnick e Siegel 2015], a habilidade de programação no século XXI deveria ser tão importante quanto ler e escrever. Essa afirmação se justifica não somente pelo fato de haver inúmeras oportunidades no mercado de trabalho, mas também para desmistificar a computação,

produzindo uma consciência maior sobre os pontos positivos/negativos, limites e vieses das tecnologias formando massa crítica para lidarmos com problemas relacionados ao tecnocentrismo [Costa e Silva 2013]. Diante disso, surge o desafio de desenvolver essa habilidade em jovens.

2.1 Motivação

A utilização de robôs tem sido bastante frequente no ensino em cursos relacionados à computação e engenharias. [McGill 2012] investigou o impacto da utilização de robôs na motivação de estudantes em disciplinas de programação. Baseando-se em teorias da motivação, a autora decompõe motivação em quatro componentes: atenção, relevância, autoconfiança e satisfação. A atenção se relaciona com o despertar da curiosidade, retenção da atenção, recompensa em aprendizado pelo esforço dispensado e ânimo para dar continuidade no aprendizado. A relevância retrata a importância dos materiais empregados, problemas e do robô no aprendizado de programação. A autoconfiança é um fator que traduz a percepção do aprendiz em relação à sua capacidade de aprendizado. Se percebe algo como muito difícil, há desmotivação. A satisfação está relacionada ao prazer/aborrecimento envolvido no aprendizado. Dentre os componentes, ao avaliar a percepção dos estudantes em cursos de programação que fizeram uso de robôs, a autora relata que a atenção foi o ponto em que houve um incremento mais significativo em relação ao grupo de controle que não os utilizou. O uso de robôs manteve a atenção de forma mais significativa do que a programação sem um artefato concreto.

Outro fator importante da motivação é a contextualização [Cooper e Cunningham 2010]. O contexto de ensino/aprendizado deve, ao mesmo tempo, ser motivador e ter certo grau de formalismo, pois, não se trata apenas de demonstrações sobre produtos que desenvolvedores/engenheiros constroem ou de projetos interessantes. Escolheu-se a robótica móvel como contexto por estar relacionada a problemas reais e por permitir exercitar facetas diferentes da engenharia de computação: solução de problemas, engenharia de sistemas, hardware, eletrônica básica, software, programação e construção coletiva de conhecimento. Em relação ao grau de formalismo, espera-se que os estudantes compreendam os princípios de funcionamento das tecnologias. Explora-se o **como** (métodos, técnicas e ferramentas), o **porquê** (as bases teóricas) e em qual **momento** é abordado no curso de engenharia de computação (período e disciplinas da grade curricular). Ressalta-se que não é intenção gerar falsas expectativas que possam atrair não-vocacionados ou reter recém-ingressos que se equivocaram na escolha do curso.

2.1 Pensamento Computacional

Pensamento computacional é uma estratégia mental utilizada para formular um problema e expressar sua solução de forma que um computador (humano ou máquina) possa realizá-la [Wing 2006]. É baseado em três estágios: abstração (formulação do problema - “o robô seguidor de linha deve ir da localização atual até o ponto B”), automação (representação da solução - “ligar o motor, ler os sensores óticos, girar para corrigir trajetória, se estiver no ponto B parar, ler os sensores óticos, ...”) e análise (execução e avaliação da solução). A execução permite analisar se a solução está correta de acordo com o pretendido e se é eficiente.

O uso do raciocínio lógico e a alfabetização digital são ferramentas úteis em várias áreas do conhecimento. Por ser uma metodologia (e não uma disciplina), o pensamento computacional pode ser utilizado em todos os níveis de aprendizado. Há muitos estudos a respeito de como desenvolver o pensamento computacional em crianças e dos seus benefícios. Por exemplo, [Minchillo 2018] relata utilização de robôs e tablets em atividades para crianças entre 9 e 10 anos e notou que a presença do robô foi motivadora para elas. Na Suíça, recentemente, introduziu-se no currículo nacional o ensino de Ciência da Computação para os níveis fundamental e médio [Lamprou e Repenning 2018]. Os mesmos autores relatam que um grande desafio é ensinar aos docentes responsáveis pela aprendizagem das crianças e adolescentes a pensarem computacionalmente, uma vez que são leigos na área.

2.2. Trabalhos relacionados

Pioneiros em pesquisas a respeito do ensino/aprendizagem de programação, [Papert e Harel 1991] evidenciam como a atividade construcionista nos experimentos de Harel levaram à retenção da atenção de crianças na criação (e na utilização) de software educacional, e à potencialização das instruções dadas pelo professor, destacando o impacto de uma ferramenta, como o computador, na construção do conhecimento.

Criada em 2007, uma das abordagens interativas para ensino/aprendizagem de programação é a plataforma online gratuita desenvolvida pelo Instituto de Tecnologia de Massachusetts (MIT) chamada *Scratch*², uma linguagem visual de programação na qual os comandos são representados graficamente na tela do computador por peças de quebra-cabeça que se encaixam. Há comandos que permitem ao aprendiz controlar os movimentos de personagens em um ambiente simulado; além destes, é possível controlar robôs reais da Lego (*Inventor*). O aprendiz segue tutoriais para construir robôs com blocos e motores que se encaixam e dão forma ao robô. Por ser uma ferramenta virtual, o *Scratch* não possui o impacto motivacional de um robô, porém, pode ser utilizada para programar os robôs *Inventor*³ que são caros e inviáveis para grande parte da população brasileira.

Em pesquisa do uso de robôs como ferramenta no ensino, [McGill 2012] relata a experiência de uso dos robôs CSbots, Qwerkbots, iRobotCreate, RoboCode e Karel the Robot no aumento da motivação dos estudantes da área da computação. Concluiu-se que o uso do CSbots acarretou melhora nas notas dos estudantes, aumento nas tarefas concluídas e no engajamento na disciplina, mas não teve efeito em diminuir a taxa de retenção dos estudantes. Os autores não puderam concluir se a motivação é similar à provocada por robôs físicos quando utilizaram os robôs virtuais RoboCode e Karel the Robot. [Silva 2019] realizou experimentos com calouros de graduação introduzindo programação com *Scratch* e kits de robótica e concluiu que houve impacto positivo no aumento da motivação dos alunos e da compreensão dos conceitos, assim como houve predisposição para continuar trabalhando com a robótica.

3. Abordagem

O projeto RobôFun surgiu de discussões internas feitas pelo grupo PET influenciadas pelo contexto de avanço da robótica na década 2010-2020. A cada ano, o grupo planeja os incrementos do robô, das oficinas/minicursos e suas ofertas. Os incrementos surgem

² <https://scratch.mit.edu/>

³ Robot Inventor custa US\$ 359,99 nos EUA (sem frete e impostos, em fevereiro de 2021) na loja da Lego

das correções de problemas identificados e de sugestões do grupo para possibilitar a pesquisa/desenvolvimento em alguma área específica da computação de seu interesse (e.g. visão computacional). A equipe de projeto é usualmente um subgrupo de 3 a 5 pessoas, parte trabalhando no hardware/software de baixo nível e parte no software de alto nível destinado ao aprendiz. As subseções que seguem apresentam os requisitos básicos do robô, o desenvolvimento do hardware/software e a metodologia de ensino/aprendizagem.

3.1. Requisitos

O requisito básico é que o robô seja facilmente replicável por pessoas sem conhecimento específico em engenharia/computação. O público-alvo são os estudantes e docentes do ensino médio, e os recém-ingressos de cursos de computação e afins. Nas suas casas ou nos colégios, eles podem replicar o projeto do robô e adotar o método de ensino/aprendizagem proposto neste projeto. Os componentes de hardware devem ser facilmente encontrados em lojas, o robô deve ser fácil de montar e o software deve ser fácil de instalar e usar. Decorre do requisito básico que os projetos de hardware e software, bem como os materiais didáticos, devem ser preferencialmente públicos de uso livre (licença CC BY-SA⁴).

Os requisitos funcionais da versão básica do robô móvel são: ser capaz de avançar, parar, girar, seguir linhas marcadas no chão e detectar obstáculos. Para a linguagem e ambiente de programação, devem permitir o aprendizado de blocos sequenciais de comando, laços de repetição, condições e variáveis. O ambiente de programação deve funcionar no modo simulado ou no modo conectado ao robô físico, e oferecer um modo de codificação direta em uma linguagem de programação (*Python*) e outro modo de codificação em uma linguagem visual.

3.2. RobôFun: Hardware e Software

O reconhecimento de imagens via utilização de câmeras e a comunicação computador-robô são funcionalidades futuras do robô. Escolheu-se assim a placa Raspberry Pi Zero W como central de processamento, pois possui alto desempenho, comunicação sem fio e *Bluetooth* embutidos. Também foram fundamentais na escolha a existência de uma grande comunidade usuária, a facilidade de compra no Brasil e seu tamanho reduzido. O Raspberry Pi não possui projeto aberto (*open hardware*), mas o grupo PET desenvolve um *shield* de projeto aberto para realizar a comunicação da central de processamento com os periféricos do RobôFun. O intuito é facilitar a conexão do Raspberry Pi com o RobôFun. O *shield* possui diversas regiões especializadas:

- para inserção de bateria, com as devidas proteções para não ocorrer danificação nos periféricos;
- central, onde tem a conexão com o Raspberry Pi;
- para o controle de motores, utilizando um circuito integrado L293D, que é utilizado como interface entre a central e o motor, podendo utilizar tensões mais altas nos motores sem interferir os outros periféricos;
- de conexão com os sensores;

⁴ Creative Commons – Atribuição-CompartilhaIgual (<https://br.creativecommons.org/licencas/>)

- conversores de nível lógico 5V do Shield para 3,3V (o Raspberry Pi trabalha com tensões de 3,3V) para impedir danos caso receba níveis de tensão maiores do que 3,3 V nas entradas;

O software utilizado para programar as primeiras versões do RobôFun chama-se JIFI⁵. O JIFI permite ao aprendiz programar o robô em uma linguagem visual. Esta linguagem utiliza os símbolos dos fluxogramas (bloco, decisão, loop) possibilitando ao aprendiz acompanhar visualmente a execução do código com animação (acendimento) dos blocos do fluxograma.

A nova versão (JIFI 2⁶) possui uma API (*Application Programming Interface*) baseada no *shield*, que permite programar o robô diretamente em linguagem Python e em uma linguagem visual ao estilo do *Blockly*⁷. Com isso permite-se o ensino de *Python*, uma linguagem de programação bastante difundida entre desenvolvedores, para recém-ingressos e o ensino de uma linguagem visual mais lúdica e fácil para os estudantes do ensino médio. Os comandos que retratam as ações básicas do robô são: avançar, parar, rotacionar de 0 a 360°, seguir linha e andar um bloco (espaço entre dois cruzamentos de linhas). Há outros comandos que permitem a leitura dos sensores do robô. JIFI 2 permite executar um programa de modo simulado (se os robôs não estão disponíveis) ou no modo físico, com atuação e sensoriamento do ambiente. Como exemplo de simulação, cita-se um labirinto, no qual o objetivo do robô é encontrar o menor caminho de um ponto ao outro.

O desenvolvimento do simulador foi realizado na linguagem de programação Python em conjunto com a biblioteca gráfica *pygame*. O uso da primeira se dá pois ela é uma linguagem altamente difundida, gratuita, de fácil uso e com muitas bibliotecas disponíveis que aceleram o desenvolvimento de software. Ela é de fácil aprendizado, contribuindo para que novos participantes do projeto adicionem novas funcionalidades ao sistema (e.g. novos blocos). A escolha da biblioteca *pygame* foi por ela atender a todas as necessidades gráficas do projeto, apresentando baixo grau de dificuldade em termos de aprendizado, desenvolvimento e manutenção. Para a construção do simulador, utilizou-se como inspiração a plataforma de agentes JASON que utiliza a linguagem *AgentSpeak* [Bordini et al. 2007]. No simulador, o robô é visto como um agente, com estados mentais baseados no modelo BDI (*Belief, desire and intentions* - crenças, desejos e intenções) [Georgeff et al. 1999]. O modelo BDI é fundamentado em teorias de raciocínio prático (voltado à ação) e responde de maneira satisfatória às necessidades de um robô móvel. A linguagem *AgentSpeak* apresenta construtos para representação e uso de estados mentais. JASON é a plataforma de execução de um sistema de agentes. Ela permite desacoplar o código dos agentes e do ambiente no qual os agentes se situam. Como *AgentSpeak* é fundamentado em um paradigma de programação de difícil compreensão inicial (misto entre declarativo e procedural), escolheu-se implementar os conceitos principais em Python para facilitar a programação e a integração com o hardware, já que há diversas bibliotecas e APIs disponíveis.

⁵ Java Interactive Flowchart Interpreter - disponível em <http://petrobo.github.io/jifi.html>

⁶ Disponível em <https://github.com/peteco-utfpr/Projeto-RoboFun>

⁷ Plataforma Blockly – disponível em <https://developers.google.com/blockly>

3.2. Oficinas e Minicursos

Para fins de motivação dos aprendizes, decidiu-se que a metodologia de ensino/aprendizagem deve ser contextualizada e concreta, realizada por meio de oficinas ou minicursos. A escolha de oficinas/minicursos centrados na experimentação prática com robôs, em vez de cursos mais longos e expositivos, é justificada pela maior informalidade, interação e liberdade dos aprendizes para criar vínculos entre si e com os instrutores. Nas duas modalidades as atividades são feitas em duplas que avançam no aprendizado resolvendo problemas/desafios na sua própria velocidade. Os instrutores dão apoio individual, porém quando detectam dúvidas similares nas diferentes duplas chamam para si a atenção da turma e realizam uma explicação/discussão geral. Nesses momentos síncronos, as duplas são sempre convidadas a compartilhar soluções e a compará-las, trocando experiências de aprendizado e ampliando o conhecimento sobre as formas de resolver um mesmo problema.

As oficinas são encontros únicos e curtos (de 2h a 3h) e visam despertar o interesse dos estudantes do ensino-médio por meio da utilização da robótica e da programação com uma linguagem visual. Os minicursos são ofertados para os recém-ingressos nos cursos de graduação na forma de encontros repetidos (carga-horária total de 12h em 3 ou 4 encontros) propiciando aos participantes uma introdução à programação, uma melhor compreensão do encadeamento das disciplinas dos cursos de graduação e da atividade exercida por profissionais de computação.

A escolha da robótica se justifica pela ligação que faz entre componentes de hardware e de software, exatamente na lacuna entre engenharia eletrônica e de computação, região na qual um engenheiro de computação desenvolve grande parte de suas habilidades. O simulador, a linguagem de programação, a comunicação de processos e o ambiente de programação permitem vislumbrar a atuação de áreas mais ligadas aos cursos de Sistemas de Informação e Ciência da Computação. Já o robô físico permite abordar tópicos de hardware (Engenharia Eletrônica).

As conexões entre os assuntos abordados nas oficinas/minicursos e as disciplinas da grade curricular dos cursos de graduação revelam a utilidade das disciplinas e da organização da grade curricular em uma linguagem acessível ao aprendiz. A Tabela 1 ilustra as relações identificadas pelo grupo PET entre algumas tarefas do projeto do robô e as disciplinas do curso de Engenharia de Computação. A parte teórica é apresentada de modo intuitivo pelos instrutores com base nos seus conhecimentos prévios.

Tabela 1. Relações entre tarefas de desenvolvimento do robô e disciplinas de Engenharia de Computação na visão do grupo PET

Tarefa RobôFun	Disciplina de Eng. de Computação
Construção das placas de circuito impresso	Introdução a práticas de laboratório em eletricidade e eletrônica, Eletricidade, Circuitos digitais, Circuitos elétricos
Comunicação entre hardware e software	Comunicação de dados, Oficina de integração 1
Criação do simulador	Fundamentos de Programação 1, Técnicas de programação, Projeto e análise de algoritmos, Análise e projeto de sistemas, Engenharia de software
Tornar o robô mais “inteligente” e autônomo	Sistemas Inteligentes, Introdução à lógica para a computação, Estrutura de dados 1 e 2
Definição de design do robô	Desenho técnico aplicado

O material didático contém uma seção explicativa do conceito de pensamento computacional, os comandos disponíveis na linguagem de programação e os problemas organizados por dificuldade: fáceis, médios e difíceis. Cada problema possui um objetivo de aprendizagem, uma lista de conhecimentos envolvidos, o enunciado e, opcionalmente, uma provocação (ou desafio). A tabela 2 ilustra um problema de nível médio. Há um caderno de soluções.

Tabela 2. Exemplo de um problema de nível médio do material didático.

Problema	Barreiras Ortogonais
Objetivo:	Aprender laços condicionais e laços de número fixo.
Enunciado:	O RobôFun deve sair da origem (ponto circular), avançar em direção às barreiras (se aproximar o máximo possível - 20 cm) na sequência indicada na figura e parar a frente da última das barreiras.
Provocação:	Caso o programa que você fez tenha mais de 8 linhas, tente reduzi-lo para o mínimo que conseguir.

Para avaliar os resultados de motivação dos minicursos, foram elaborados dois questionários, pré e pós-curso, inspirados no trabalho de [McGill 2012]. As perguntas versam sobre os componentes da motivação (tabela 3) e os aprendizes graduam as respostas em uma escala de Likert de 1 a 5 (1-discorda totalmente e 5-concorda totalmente).

4. Resultados: Lições Aprendidas

Os resultados estão baseados nas respostas dadas por 48 estudantes aos questionários pré e pós de três das dez oficinas realizadas. Para cada componente (ver tabela 4), calculou-se a média das notas de 1 a 5 dadas pelos estudantes para cada questão. Os resultados mostram que o robô e a metodologia de ensino/aprendizagem são atrativos e despertam interesse inicial. Percebe-se que os valores no formulário pós-curso para atenção e relevância diminuíram. Acredita-se que isto ocorreu por problemas de operacionalização

das oficinas (ver lições aprendidas). Interessante notar que a autoconfiança teve aumento, acredita-se que pela desmistificação dos robôs como sendo algo inacessível.

Tabela 3. questionários de avaliação pré e pós-curso para cada um dos componentes da motivação: atenção, relevância, confiança e satisfação.

Componente	Questionário PRÉ	Questionário PÓS
Atenção	(a) Programar com robôs atraiu minha atenção. (b) A oportunidade de desenvolver minhas habilidades com programação chamou a minha atenção.	(a) Aprender a programar com os robôs estimulou a minha criatividade. (b) O trabalho realizado para aprender a programar manteve a minha atenção. (c) Os vários usos do robô me ajudaram a manter a atenção. (d) Usar o robô para aprender a programar me fez sentir recompensado pelo esforço. (e) Eu gostaria de saber mais sobre o robô.
Relevância	(a) Creio que usar o robô me ajudará a aprender a programar.	(a) Vi utilidade no uso dos robôs no aprendizado de programação que tive. (b) No curso, os problemas, materiais e exemplos me mostraram como programar com robôs pode ser importante para pessoas que estão aprendendo a programar.
Confiança	(a) O uso de robôs me deixa confiante que terminarei o minicurso. (b) Programar robôs aparenta ser intimidador. (c) Sou capaz de resolver desafios lógicos.	(a) Eu pude entender o material usado para ensinar a como programar com o robô. (b) Programar os robôs foi intimidador. (c) Os exercícios com o robô estavam dentro do meu alcance. (d) Sou capaz de resolver desafios lógicos.
Satisfação	Não se aplica	(a) Quero fazer outros cursos que requeiram desafios lógicos. (b) Quero fazer novos cursos com robôs. (c) Solucionar os problemas foi aborrecedor.

Tabela 4. resultados coletados - médias das notas na escala de Likert de 1 a 5.

COMPONENTE	PRÉ	PÓS
ATENÇÃO	4,74	4,37
RELEVÂNCIA	4,16	3,87
AUTOCONFIANÇA	3,55	3,99
SATISFAÇÃO	não se aplica	4,69

Os comentários dos questionários e as experiências em sala nos renderam diversas lições que podem ser categorizadas em *i*) referentes ao desenvolvimento do projeto do robô (construção e planejamento de software e hardware); *ii*) à realização das oficinas/minicursos do RobôFun. As lições abaixo listadas foram construídas em reuniões realizadas pelo grupo PET, em momentos de discussão das oficinas e do projeto do robô, e de conversas informais com ex-participantes das oficinas. Estas lições foram anotadas nos relatórios de atividades do grupo e, finalmente, aqui são compiladas.

Desenvolvimento do projeto

- Superestimar a capacidade técnica dos estudantes dos períodos iniciais e sua capacidade de projetar sistemas: no grupo PET em questão, há predominância de estudantes do 2º ao 5º período (semestre) que cursaram poucas disciplinas específicas de Engenharia de Computação. Portanto, suas habilidades ainda estão em desenvolvimento e podem não conhecer todos os aspectos técnicos ligados à construção do robô e dos sistemas de software relacionados. Os estudantes possuem visão sistêmica ainda em desenvolvimento. Na maior parte das disciplinas desenvolvem projetos que cabem no tempo delas (*toy problems* ou provas de conceito). O RobôFun requer integração de diferentes plataformas (hardware, software, ambiente de programação), portanto uma arquitetura que envolve diferentes módulos, tolerante a falhas e que se preocupe com aspectos de interação humano-computador. Todos estes requisitos, funcionais e não funcionais, são demasiadamente complexos sem a orientação de um ou mais docentes. A orientação deve ser consultiva, os estudantes devem estar no controle do projeto. Esta lição foi aprendida em reuniões de avaliação do projeto.
- Compartilhamento de conhecimento devido à rotatividade dos membros da equipe: por se tratar de um grupo onde há processos seletivos cerca de duas vezes ao ano, a rotatividade de estudantes no projeto é elevada. Esse fator causa uma necessidade de repassar o estado do projeto em pouco tempo aos novos integrantes para que estes possam continuar o desenvolvimento do projeto sem hiatos. A solução é trabalhar em pares e documentar. Esta lição foi aprendida por observações do tutor e discutida em reuniões de projeto.

Oficinas e Minicursos

- Operacionalização é fator importante: robôs funcionando em número adequado, baterias carregadas, software de comunicação testado e funcionando para comandar os robôs são fatores que influenciam no bom andamento das oficinas. Isto requer engajamento dos instrutores/equipe de projeto em preparar o material previamente, manter os robôs e corrigir *bugs* do sistema. Como fator para aumentar o engajamento, colabora muito a avaliação realizada pelos aprendizes que afeta positivamente o comportamento dos responsáveis. Esta lição foi aprendida dos comentários de aprendizes feitos nos questionários pós-curso.
- Integração de estudantes (internos): os recém-ingressos entram em contato com os veteranos do curso e criam ligações que servirão para orientá-los (ex. em conversas de corredor). Observa-se que estas ligações levam muitos participantes das oficinas a postularem vaga no grupo PET. Esta lição foi aprendida pela observação e confirmada nas entrevistas de seleção do grupo.
- Integração de estudantes (externos): os vínculos se formam rapidamente entre os universitários e os estudantes do ensino médio. Fator importante, principalmente nas escolas públicas, é que o grupo PET desmistifica a universidade e o acesso a ela, dialogando sobre formas de entrada (ex. SISU, cotas), outros cursos/atividades e de manutenção (ex. bolsas disponíveis). Esta lição foi aprendida em nossos contatos com docentes/diretores de escolas públicas dos quais recebemos de diferentes escolas a sugestão de tornar as oficinas mais frequentes.

- Turmas pequenas: a metodologia de ensino/aprendizagem adotada e o uso de robôs requerem bastante esforço de atendimento e atenção dos instrutores durante os cursos. Recomenda-se 1 instrutor para no máximo 6 aprendizes. Outro fator limitante é o número de robôs disponíveis e de cenários com linhas no chão e obstáculos. A motivação degrada bastante em relação à satisfação (aborrecimento) e à relevância (para que serve o robô se não funciona?). Esta lição foi aprendida a partir dos comentários feitos em questionários pós-curso.

6. Conclusão

O RobôFun está contextualizado na realidade do grupo PET e do nosso município trabalhando principalmente com as causas de evasão relacionadas à vocação e causas institucionais. Em relação à vocação, ao apresentar a atividade do profissional de computação e possibilidades de atuação, desfazem-se falsas expectativas dos estudantes de ensino médio ao mesmo tempo que os fazem vislumbrar outras possibilidades de atuação profissional. Percebemos que alguns estudantes não associam computação com matemática, pensam que vão “aprender a mexer, jogar”. Além da desmistificação da tecnologia, as oficinas dão oportunidade de desmistificar a universidade. Vários estudantes do ensino médio não têm em seu horizonte a realização de um curso de graduação. Relatam que não conhecem os tipos de cotas e os auxílios financeiros para se manter em uma universidade. Para os aprendizes em geral, o projeto possibilita observar na prática conceitos e conhecimentos que são adquiridos ao longo do curso de computação e possivelmente identificar se o curso está no escopo de interesse deles.

As causas institucionais são abordadas à medida que os estudantes enxergam utilidade nas disciplinas que pareciam soltas na grade curricular. Este fato foi evidenciado informalmente em conversas com ex-participantes das oficinas. A integração estudante-instituição e estudante-estudante é fortalecida por meio do contato com o grupo PET. Os participantes do grupo passam a ser orientadores dos recém-ingressos dando dicas de como agir na instituição, das oportunidades, das trilhas de formação. O projeto age positivamente nos integrantes do grupo PET possibilitando o crescimento pessoal e profissional. Desenvolve neles a capacidade de resolução de problemas, uma vez que o projeto RobôFun é um “problema de escopo aberto”, onde os integrantes devem gerir o grupo e o projeto e encontrar soluções.

Estima-se que o projeto RobôFun tem impacto positivo na sua tarefa de estimular o pensamento computacional, atrair estudantes e amenizar a evasão dos cursos de computação e afins. Os questionários de avaliação permitem capturar a percepção dos aprendizes, mas uma avaliação mais duradoura e sistemática é necessária para se obter uma relação causal forte (se é que é possível, já que há muitas variáveis que influenciam na evasão/atração). Os relatos colhidos por meio dos questionários, entrevistas de seleção e conversas informais são positivos.

No prosseguimento do projeto, pretende-se implementar funcionalidades que permitam aos integrantes do grupo proporem novos desafios relacionados à inteligência artificial e aumentar a visibilidade do projeto para que seja replicado, evolua e se adapte às realidades de outras regiões do país. O grupo poderá contribuir com outras instituições na construção de problemas/desafios abertos, na montagem dos robôs para resolvê-los e na programação, vivenciando como se pratica o ensino de pensamento computacional contribuindo para o letramento digital de forma mais ampla.

7. Referências

- Barroso, M. F. e Falcão, E. B. M. (2004). Evasão Universitária: o caso do Instituto de Física da UFRJ. *IX Encontro Nacional de Pesquisa em Ensino de Física*.
- Bordini, R. H., Hübner, J. F. e Wooldridge, M. (2007). Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology). Editado por John Wiley & Sons Inc., EUA.
- Cooper, S. e Cunningham, S. (2010). Teaching computer science in context. *ACM Inroads*, v. 1, n. 1, p. 5–8.
- Costa e Silva, G. (2013). Tecnologia, educação e tecnocentrismo: as contribuições de Álvaro Vieira Pinto. *Revista Brasileira de Estudos Pedagógicos*, v. 94, n. 238, p. 839–857.
- Georgeff, M., Pell, B., Pollack, M., Tambe, M. e Wooldridge, M. (1999). The Belief-Desire-Intention Model of Agency. *Intelligent Agents V: Agents Theories, Architectures, and Languages*. Springer, Berlin, Heidelberg.
- McGill, M. M. (2012). Learning to Program with Personal Robots. *ACM Transactions on Computing Education*, v. 12, n. 1, p. 1–32.
- Minchillo, L. V. (2018). Na direção de melhores ferramentas e metodologias para o ensino de pensamento computacional para crianças. *Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação*. Campinas, SP.
- Papert, S. e Harel, I. (1991). Situating Constructionism. Em *Constructionism*. Ablex Publishing, EUA.
- Resnick, M. e Siegel, D. (2015). A Different Approach to Coding. *International Journal of People-Oriented Programming*, v. 4, n. 1, p. 1–4.
- Silva, A. S. (2019). A robótica educacional como possibilidade para o ensino de conceitos de lógica de programação. *Dissertação (mestrado) - Universidade do Vale do Taquari*. Lajeado, RS.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33–35.