

ProgressCode: uma Ferramenta para Monitorar o Progresso de Alunos no Ensino Não-formal de Programação

Lucas F. Silva¹, Iago C. Corrêa¹
Andrea S. Charão¹, João V. F. Lima¹

¹ Universidade Federal de Santa Maria (UFSM)
Santa Maria – RS – Brasil

{lferreira, icorrea, andrea, jvlima}@inf.ufsm.br

Resumo. *O ensino não-formal de programação ao público em geral, sob forma de projetos e atividades extra-curriculares, tem sido alvo de várias iniciativas. Nessas práticas, marcadas por interesses, atividades e embasamentos heterogêneos, o acompanhamento do aprendizado é um desafio. Neste trabalho, apresenta-se ProgressCode, uma ferramenta criada com o intuito de facilitar o registro e apresentação de observações feitas por tutores durante atividades de ensino não-formal de programação. A ferramenta compreende um aplicativo móvel para coleta de dados e um sistema web que permite visualizar informações sobre o progresso de cada participante. Testou-se ProgressCode durante oficinas de programação, nas quais a ferramenta mostrou cumprir seus objetivos.*

Abstract. *Non-formal teaching and learning of programming for the general public, in the form of extra-curricular projects and activities, has been the target of several initiatives. In these practices, characterized by heterogeneous interests, activities and foundations, the monitoring of learning is a challenge. In this work, we present ProgressCode, a tool created with the purpose of facilitating the recording and presentation of observations made by tutors during activities of non-formal programming teaching. The tool comprises a mobile application for data collection and a web system that allows one to view information about the progress of each participant. ProgressCode was tested during programming workshops, in which the tool showed to fulfill its objectives.*

1. Introdução

Em muitos níveis e modalidades de ensino, as práticas didáticas usufruem de meios tecnológicos para dinamizar e facilitar a aprendizagem, mesmo em atividades presenciais. No ensino-aprendizagem de programação e fundamentos de computação, pode-se contar com recursos variados, desde a simples propagação *on-line* de materiais de apoio (exercícios, tutoriais sobre linguagens, animações de algoritmos, etc.) até a existência de ambientes interativos e colaborativos, em que grupos de alunos recebem *feedback* imediato ao resolver problemas de programação.

Entretanto, mesmo com a variada gama de funcionalidades das tecnologias de apoio, ainda assim muitos fatores acabam sendo desconsiderados na dinâmica de ensino adotada, mesmo em atividades presenciais com presença de tutores. A cada prática realizada por alunos, é gerada uma grande quantidade de informações que caracterizam seu

progresso no aprendizado. Porém, fatores subjetivos às dificuldades e entendimento de cada aluno, mesmo quando percebidos presencialmente pelos tutores, acabam não gerando registros e, por isso, podem não ser considerados em etapas subsequentes.

No âmbito do ensino não-formal de programação, essa situação agrava-se ainda mais. Seja na forma de projetos, oficinas, clubes ou outras atividades extra-classe, o ensino não-formal tem características mais flexíveis que o ensino formal, como por exemplo: ser voluntário (e não compulsório), ser centrado em interesses do aluno e não ser sequencial [Eshach 2007]. Essa flexibilidade gera práticas marcadas por interesses, atividades e embasamentos heterogêneos. Assim, a observação individualizada do processo de aprendizagem assume um papel importante no planejamento, execução e avaliação de cada encontro com alunos.

A ferramenta ProgressCode, apresentada neste artigo, insere-se neste contexto e foi desenvolvida no âmbito de um programa de extensão universitária que, há mais de três anos, promove o ensino não-formal de programação a crianças e jovens, com atividades presenciais intermediadas por tutores. O objetivo da ferramenta é facilitar o registro e visualização do desempenho dos participantes nas atividades, auxiliando os tutores no processo de acompanhamento dos alunos. Isso possibilita a centralização e refinamento da grande quantidade de informações acerca do progresso dos alunos, de forma a realimentar o processo de ensino-aprendizagem.

No decorrer deste trabalho é apresentado o contexto que inspirou a criação do ProgressCode (seção 2), o detalhamento do processo de desenvolvimento, arquitetura e requisitos da ferramenta (seção 3), os resultados obtidos e a avaliação do uso em um cenário real (seção 4) e por fim as conclusões obtidas do uso e as aspirações para trabalhos futuros (seção 5).

2. Contextualização

2.1. Problema e Motivação

O cenário deste trabalho é vivenciado em muitos projetos de ensino de programação para crianças e jovens, em que tutores auxiliam os participantes das atividades propostas tentando sanar as dúvidas que vão surgindo em cada exercício. Na Universidade Federal de Santa Maria (UFSM), desenvolve-se um programa de extensão universitária que promove várias atividades de ensino não-formal, incluindo oficinas e minicursos de aprendizado de lógica computacional. Nestas atividades, vivencia-se um problema recorrente que é a descentralização das informações de desempenho de cada aluno e turma, já que é feito o uso de ferramentas variadas em cada atividade. Desta forma, passa a existir a necessidade de métodos e ferramentas que facilitem o acompanhamento dos participantes das aulas, para que se possa usufruir com mais eficiência das observações particulares feitas sobre o desempenho de cada aluno, o que desencadeia em um melhor aproveitamento da proposta didática como um todo.

A partir do estudo de caso neste programa extensionista, constatou-se a necessidade de uma solução que pudesse auxiliar os tutores no processo de acompanhamento dos alunos, bem como uma solução para centralizar a grande quantidade de informações relacionadas às atividades e seus respectivos participantes. Tais demandas foram os fatores instigantes para a criação do ProgressCode.

2.2. Trabalhos Relacionados e Soluções Existentes

A ideia de acompanhar o progresso de pessoas em atividades não é nova, sendo alvo de diferentes soluções.

Aplicações de “*progress tracking*”, como Strides [Conquer Apathy 2016] e Habit List [Habit List 2017], permitem que o usuário estipule metas para tarefas diárias e, na medida em que cada meta é alcançada, o próprio usuário a registra no aplicativo. Esses aplicativos permitem que sejam elaborados registros relacionados aos mais diversos contextos, porém, impedem que tais anotações sejam realizadas por terceiros, ou seja, apenas o usuário que irá realizar a atividade pode atualizar o andamento da mesma.

Com outra abordagem, existem também os “Ambientes Virtuais de Aprendizagem” ou AVAs, que apresentam-se como estruturas tecnológicas virtuais que visam auxiliar o aprendizado de quem os usufrui. Por meio desses ambientes, usuários têm acesso a exercícios e materiais de ensino que podem ajudá-los em seu desenvolvimento educacional em determinada área. Os AVAs são comumente utilizados para facilitar o acesso a conteúdos didáticos [de Brito et al. 2013], sendo também úteis a professores e instrutores, quando dispõem de recursos para acompanhar a realização de atividades. Plataformas desse tipo tendem a ser bastante robustas e completas, mas não costumam permitir registros detalhados sobre a realização de atividades. Além disso, a grande quantidade de recursos pode se tornar um problema no ensino não-formal, em que não se pode dedicar muito tempo para que os alunos aprendam a utilizar o ambiente.

Um outro exemplo de ambiente educacional é o usado na Hora do Código, projeto mundial do Code.org, que visa desmistificar a ideia de que programação é algo difícil [Code.org 2017]. Na plataforma desenvolvida para as atividades da Hora do Código, é possível a criação de turmas e o acompanhamento do desempenho dos participantes das atividades. Porém, essa “gestão” de turmas é bastante acoplada às atividades do projeto, não sendo possível fazer registros sobre outras atividades não previstas.

Em edições passadas do Workshop sobre Educação em Computação (WEI), alguns trabalhos apresentaram possíveis soluções para a problemática de acompanhar o desempenho dos alunos. Uma das propostas foi o TestWeb [de Aquino et al. 2006], uma ferramenta para avaliação de alunos do ensino à distância por meio da análise de respostas de questões de estilos variados, não só as de múltipla escolha. Outro trabalho segue na direção da avaliação e correção automática de exercícios de programação, propondo mais facilidade para os professores na hora da correção das atividades e avaliação dos alunos [Moreira e Favero 2009].

Ambos os trabalhos se concentram em avaliações formativas, porém não permitem o registro de anotações e observações que possam retratar aspectos mais particulares de cada aluno, mas que tendem a refletir diretamente no seu desempenho no processo de aprendizagem.

3. Desenvolvimento da Ferramenta

O sistema proposto é basicamente composto por duas tecnologias: um aplicativo móvel e um sistema web. O primeiro tem como objetivo auxiliar os tutores na supervisão e coleta de dados dos alunos, já o segundo é destinado a persistência e visualização centralizada das informações coletadas pelos tutores por meio do aplicativo.

3.1. Requisitos

Um dos principais requisitos do ProgressCode é permitir o registro de observações detalhadas sobre o progresso de alunos em atividades. Essas observações devem ser registradas por tutores, de forma rápida, sem prejudicar o processo de tutoria.

Com o aplicativo móvel, torna-se possível que os tutores das atividades registrem o desenvolvimento do aluno durante atividades realizadas. Esse procedimento pode ser feito por meio de “emojis” para selecionar, de uma forma interativa e objetiva, a opção que retrata o rendimento observado na atividade que está sendo executada.

A plataforma web é responsável por agrupar as informações provenientes da observação de cada tutor em seu aplicativo. O gerenciamento é simples: um responsável cadastra um evento e seus respectivos participantes e encontros, os quais são compostos por atividades como desafios ou exercícios. Há também, armazenado no banco de dados, informações sobre os inscritos nos eventos propostos, permitindo uma visualização de todos os registros de desempenho por atividade de cada um dos participantes.

Ainda no sistema web, é possível o acesso a recursos como a visualização do progresso de cada aluno ou turma em determinada atividade, ou também o andamento geral das tarefas de um encontro.

Com as informações dos tutorados em mãos, é possível ao tutor e responsável avaliarem suas formas ensino e didáticas empregadas nas atividades, em função dos objetivos atingidos ou não. Um estudo dos dados coletados dá condições aos tutores de mapear alunos e atividades que requeiram mais atenção na tutoria.

3.2. Arquitetura e Implementação

O projeto como um todo, como já citado anteriormente, é composto por um sistema web e um aplicativo móvel. O sistema web foi desenvolvido utilizando-se a linguagem de programação Python, versão 3, sobre o framework Django 1.10. Já o aplicativo móvel foi desenvolvido com suporte a API 19, ou superiores, do sistema operacional Android. O

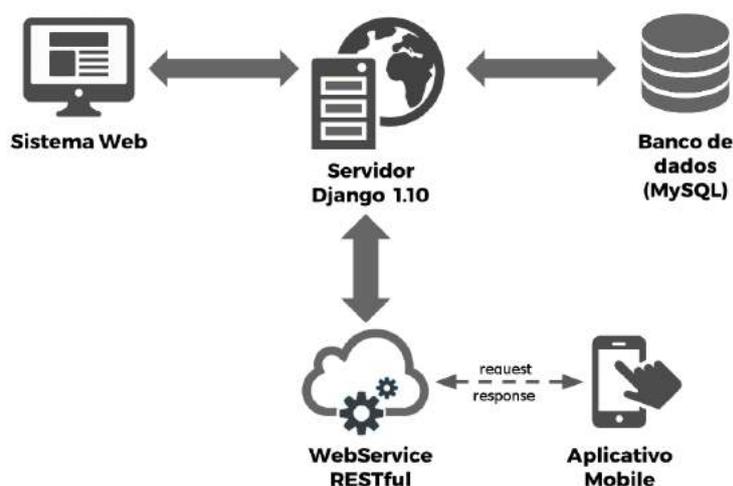


Figura 1. Diagrama arquitetural do ProgressCode

ProgressCode baseia-se em uma arquitetura *mobile-desktop* em que tarefas como criação de eventos, cadastro de turmas, análise dos dados coletados, entre outras, se concentram somente na aplicação web, permitindo assim que o aplicativo móvel seja responsável somente pela coleta dos dados. Este modelo de funcionamento vem como uma facilidade para os tutores, que necessitam apenas de um *smartphone*, que pode acompanhá-los durante o auxílio dos alunos, para o registro das observações.

Como pode ser visto no diagrama arquitetural (Figura 1), os dados coletados com o aplicativo mobile são enviados para um banco de dados MySQL por intermédio de um Webservice RESTful que, assim como o sistema web, está em execução em um servidor com Django.

3.3. Interface com o Usuário

A plataforma web é composta por um menu lateral que permite acesso a todas as funcionalidades responsáveis pelo gerenciamento das turmas e eventos, como por exemplo: página de cadastro dos tutorados, adicionar e remover eventos, bem como, seus respectivos encontros e atividades. Também é possível encontrar no sistema a lista dos tutores responsáveis por aplicar as atividades de cada encontro, além da possibilidade de vinculá-los ou desvinculá-los a um determinado encontro.

Tanto os registros de desempenho quanto os eventos, encontros e atividades, são mostrados ao usuário em forma de listas, que permitem uma visualização objetiva das informações de cada item. Também é possível navegar através das listas buscando atividades de cada encontro ou encontros de cada evento. A figura 2 é um exemplo dessas listas: nela é apresentada a página que contém a listagem das observações realizadas pelos tutores das atividades por meio do aplicativo mobile instalado em seus *smartphones*.

Status	Data do Registro	audio	Atividade	Inscrito	Tutor
3	Nov. 26, 2016, 9:39 a.m.		Hora do Código - Labirinto	Amanda	
3	Nov. 26, 2016, 9:39 a.m.		Hora do Código - Outro	Amanda	
3	Nov. 26, 2016, 9:39 a.m.		App - Baixar imagens e som	Amanda	
3	Nov. 26, 2016, 9:40 a.m.		Hora do Código - Labirinto	Rafael	
3	Nov. 26, 2016, 9:40 a.m.		Hora do Código - Outro	Rafael	
3	Nov. 26, 2016, 9:40 a.m.		App - Baixar imagens e som	Rafael	
3	Nov. 26, 2016, 9:42 a.m.		Hora do Código - Labirinto	Rafael	
3	Nov. 26, 2016, 9:42 a.m.		Hora do Código - Outro	Rafael	
3	Nov. 26, 2016, 9:42 a.m.		App - Baixar imagens e som	Rafael	

Figura 2. Lista de observações coletadas pelos tutores com o aplicativo

O usuário dispõe de duas opções para adicionar participantes de atividades. Na primeira, ele pode adicioná-los manualmente, inserindo um por um. Na segunda forma, é possível a importação de um arquivo .csv para o cadastro de um grupo inteiro de alunos, buscando automatizar o processo de inserção de inscritos no sistema.

Para a coleta dos dados, o usuário (tutor) deve se autenticar no aplicativo, utilizando credenciais que foram cadastrados na plataforma web (Figura 3 (a)).

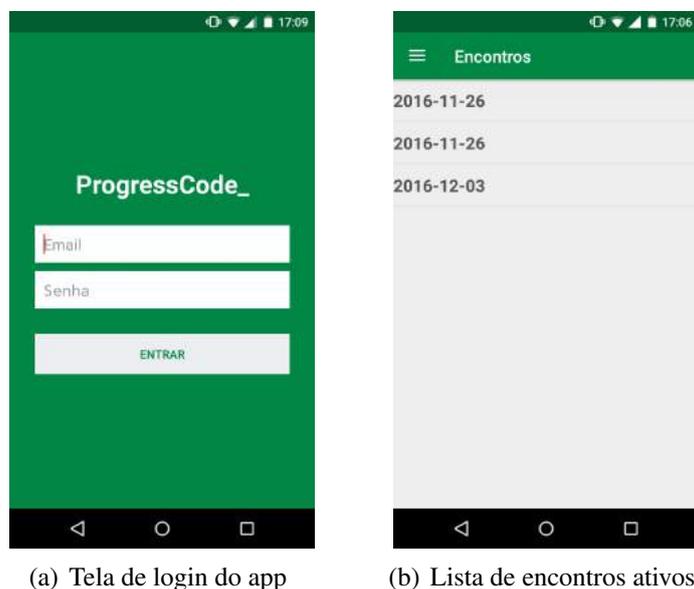


Figura 3. Telas iniciais do aplicativo

Depois de autenticado, o tutor é direcionado para uma tela contendo todos os encontros cadastrados na página web e que ainda estão ativos (Figura 3 (b)). Ao selecionar o encontro desejado apresenta-se a tela contendo todos os participantes cadastrados para o encontro em questão (Figura 4 (a)) e então, para o tutor começar a fazer as observações do aluno, basta selecionar na lista o nome do mesmo.

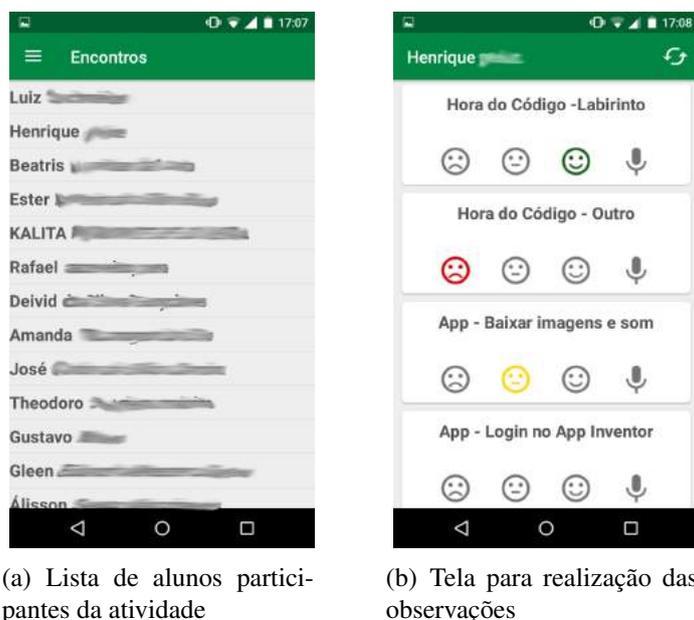


Figura 4. Telas do processo de registro de observações sobre o aluno

Quando um tutor seleciona um determinado nome na lista de alunos, o aplicativo apresenta uma tela contendo todas as atividades do encontro anteriormente selecionado

(Figura 4 (b)) e, a partir de botões em forma de “emojis”, o usuário pode classificar o desempenho do participante de forma interativa e objetiva. Esses botões podem ser vistos na figura 4 e apresentam “um rosto triste”, “um rosto neutro” e “um rosto feliz”, sendo que cada um representa respectivamente: “muita dificuldade para fazer a atividade”, “um pouco dificuldade para fazer a atividade” e “concluiu a atividade sem dificuldades”. Essa forma de registrar observações visa permitir ações rápidas por parte dos tutores, ao mesmo tempo que dá flexibilidade para se definir atividades em maior ou menor grau de detalhamento.

O botão restante, em forma de microfone, é uma funcionalidade destinada para a gravação das observações em forma de áudio.

4. Testes e Resultados

Assim que uma versão utilizável do sistema ficou disponível para uso, fazia parte do intento dos desenvolvedores testá-la em uma situação real, buscando avaliar se os requisitos levantados na fase de modelagem do projeto atendiam todas as necessidades demandadas pelo usuário final.

Como primeiro teste, o ProgressCode foi utilizado por tutores do programa de extensão da UFSM e intitulado: “Clube de Computação para o Público do Ensino Médio e Fundamental”, em uma de suas oficinas que promovem o ensino de programação a crianças e jovens. O aplicativo foi utilizado em três encontros, por cinco tutores, gerando registros reais sobre o progresso de tutorados nas atividades.

Notou-se que, ao que se propõe, o ProgressCode cumpriu com as expectativas, pois auxiliou os tutores do projeto a lembrarem das dificuldades evidenciadas pelos tutorados com os registros efetuados no decorrer da atividade. Mesmo o sistema estando ainda em fase de aprimoramento de funcionalidades, foi perceptível aos tutores do Clube sua vantagem de uso durante as atividades.

5. Conclusão e Trabalhos Futuros

Neste trabalho, apresentou-se uma alternativa para o registro de observações sobre o progresso de alunos em atividades de ensino não-formal de programação.

Dadas suas motivações, o ProgressCode cumpre os objetivos propostos, tornando-se uma ferramenta aliada em projetos de ensino não-formal de programação, pois apresenta-se como uma solução simples e objetiva para o auxílio no acompanhamento dos alunos, acomodando diferentes tipos de atividades, em maior ou menor grau de detalhamento.

Como trabalhos a serem desenvolvidos, é vislumbrado um aprimoramento da parte dedicada à visualização do progresso dos alunos no sistema web, trazendo, por exemplo, estatísticas sobre os exercícios que os alunos tiveram mais ou menos dificuldades.

Visando contribuir com projetos que enfrentem a mesma problemática, o ProgressCode é disponibilizado como um software de código aberto e está disponível em um repositório público [ProgressCode 2016], podendo ser baixado, estudado ou mesmo modificado.

Referências

- Code.org (2017). Hour of code. <https://hourofcode.com/br>.
- Conquer Apathy (2016). Strides: Track all your goals e habits in one place. <http://www.stridesapp.com/>.
- de Aquino, C. L. C., de Souza, G. Q., da Silva, J. C. T., e da Conceição Almeida Pfister, M. (2006). TestWeb - A Implementação de Uma Ferramenta de Avaliação à Distância. In *Anais do Workshop sobre Educação em Computação*.
- de Brito, L. M., Júnior, J. R. G., Gomes, S. G. S., e Mota, J. B. (2013). Ambientes virtuais de aprendizagem como ferramentas de apoio em cursos presenciais e a distância. *RENOTE - Revista Novas Tecnologias na Educação*.
- Eshach, H. (2007). Bridging in school and out-of-school learning: formal, non-formal, and informal education. *Journal of Science Education and Technology*, 16(2):171–190.
- Habit List (2017). Habit list: Build a better you. <https://habitlist.com/>.
- Moreira, M. P. e Favero, E. L. (2009). Um Ambiente para Ensino de Programação com Feedback Automático de Exercícios. In *Anais do Workshop sobre Educação em Computação*.
- ProgressCode (2016). Progresscode repositório git. <https://github.com/LucasFerreiraDaSilva/ProgressCode>.