

Introdução à programação em Python utilizando smartphone: um relato de experiência usando a abordagem construcionista de Seymour Papert

Introduction to Python programming using a smartphone: an experience report using Seymour Papert's constructionist approach

Elisrenan Barbosa da Silva¹

Daniel Oliveira Dantas²

Kalil Araujo Bispo³

Resumo

O ensino-aprendizagem de programação é algo desafiador tanto para o aluno quanto para o professor. Dentre os maiores desafios podemos citar: o que fazer para manter os estudantes motivados, como facilitar o entendimento do conteúdo e evitar a exclusão de alunos com poucos recursos. Para tanto, 53 alunos da disciplina de Programação Imperativa conseguiram aprender a programar a partir do uso de smartphones no aplicativo QPython, com o auxílio do aporte teórico de ensino construcionista de Seymour Papert, em conjunto com o WhatsApp, todos tiveram um acompanhamento diário. O objetivo deste artigo é descrever como o método foi aplicado, apresentar em quais conteúdos os alunos encontraram mais dificuldades e discutir a análise feita a partir das informações obtidas. Os resultados mostram que a intervenção realizada obteve sucesso, principalmente no que diz respeito ao aumento da taxa de aprovados e redução da evasão acadêmica.

Palavras-Chaves: Educação. Construcionismo. Ensino Móvel. Ensino Programação. Educação Superior.

Abstract

The teaching-learning of programming is something challenging for both the student and the teacher. Among the biggest challenges we can mention: what to do to keep students motivated, how to facilitate the understanding of the content and avoid the exclusion of students with few resources. To this end, 53 students from the Imperative Programming discipline managed to learn to program using smartphones in the QPython application, with the help of the theoretical contribution of constructionist teaching by Seymour Papert, together with WhatsApp, all of them had a daily monitoring. The objective of this study is to describe how the method was applied, to present the contents where the students found more difficulties, and to discuss the analysis made from the information obtained. The results show that the intervention was successful, mainly with regard to the increase in the rate of approval and reduction of academic dropout.

Keywords: Education, Constructionism. Mobile Learning. Programming Teaching. Higher Education.

¹ Programador, professor de informática básica e de computação no ensino superior, mestre em Ciência da Computação (UFS)

² Professor do Departamento de Computação (DCOMP) e do Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS). Possui Graduação, Mestrado e Doutorado em Ciência da Computação pela Universidade de São Paulo (USP). Suas principais áreas de interesse são processamento paralelo, de imagens médicas e de biossinais.

³ Professor Doutor em Ciência da Computação no DCOMP-UFS desde 2013. Suas principais áreas de interesse são Sistemas Distribuídos, Rede de Sensores sem Fio e Internet das Coisas

1. Introdução

A atual ubiquidade dos dispositivos móveis, em especial dos *smartphones*, viabiliza cada vez mais o acesso das classes sociais C e D a recursos como a Internet (Nielsen, 2015). Uma pesquisa realizada pela Agência Brasil (2019) identificou que 70% da população brasileira tem acesso à Internet e que 97% dela a utiliza pelo *smartphone*, sendo que a maior taxa de crescimento desse uso está nas classes C, D e E.

De acordo com West e Vosloo (2014), muitos educadores em todo o mundo têm utilizado esses dispositivos móveis como ferramenta de ensino, buscando facilitar a aprendizagem com recursos inovadores. Principalmente, quando se pensa no quanto os alunos estão conectados a esses dispositivos, mesmo quando em movimento, de casa para o trabalho ou do trabalho para a faculdade (Mohammad, 2015).

Atualmente, o *smartphone* possui capacidade de processamento similar à de computadores pessoais. Inclusive, existem ferramentas que emulam *integrated development environments* (IDEs) e que podem ser usadas no ensino de programação, permitindo aos alunos que programem em qualquer lugar, mesmo sem um computador ou Internet.

Isso se torna possível, de acordo com Borba e Lacerda (2015), pois os *smartphones* já estão nas mãos dos alunos, o que reduz custos de implantação de ferramentas e de energia elétrica na aplicação dessa prática de ensino que é o modelo *m-learning* (aprendizagem móvel), tornando-se uma opção para o ensino e aprendizagem.

O grande desafio é como implementar o *m-learning* em contextos tradicionais e em que não haja recursos como computadores para o ensino de Introdução a Programação, evitando principalmente a evasão acadêmica. Foi necessário então utilizar algum método de ensino em conjunto que fortalecesse o aprendizado de forma que fosse possível potencializar o aprendizado de todos.

Utilizamos uma abordagem baseada no construcionismo de Seymour Papert, que recomenda a apresentação de problemas familiares aos alunos para estimular seu interesse, como também define que o professor deve realizar um acompanhamento aluno a aluno para identificar as dificuldades específicas de cada um e fazer de tudo para saná-las.

Este trabalho propõe um método de ensino de programação baseado no uso de *smartphones*. Foi escolhido o aplicativo QPython em conjunto com o Python⁴ como linguagem

⁴ Python, disponível em: <https://python.org.br/> e Qpython3, disponível em: <https://www.qpython.com/>

introdutória para aprender a programar, uma vez que segundo Sotomayor-Beltran (2019), este é um dos mais utilizados no mundo, tanto por empresas quanto por professores de cursos técnicos e de graduação como forma de introduzir o ensino de programação aos seus alunos, seja em um ambiente *desktop*, *web* ou *mobile*.

Para permitir a inclusão social de alunos com poucos recursos foi utilizado o aplicativo de comunicação WhatsApp para acompanhamento e interação durante e após as aulas, tendo em vista que de acordo com Romero (2019) essa é uma das opções ideais para realizar um debate, tirar dúvidas, receber e enviar documentos.

No decorrer do período letivo foram dados 135 problemas para serem resolvidos individualmente e oito projetos para serem trabalhados em equipe. Em uma turma composta por 53 alunos. Os problemas são agrupados em três níveis de dificuldade: fácil, intermediário e difícil.

Para alcançar os objetivos de ensino foi utilizado como base parte do modelo Goal Questions Metric (GQM), proposto por Basili e Weiss (1984), em que foram definidas questões de pesquisa que pudessem ser resolvidas com os resultados obtidos durante o período letivo. Apresentamos assim, os índices de cada conteúdo e as dificuldades encontrados pelos alunos durante o período letivo. Essas informações foram obtidas com os exercícios respondidos em sala de aula e por meio dos projetos realizados pelos estudantes até o fim do período.

2. Referencial teórico

A grande quantidade de *smartphones*, que já ultrapassa mais da metade do número de habitantes no Brasil (INEP, 2021), permite a acessibilidade de aprendizagem para todos. Os dispositivos móveis possuem uma grande capacidade de auxiliar o ensino de linguagens de programação introdutório.

Nosso foco foi pesquisar a utilização do *m-learning* no ensino de Introdução à Programação com uso dos dispositivos móveis tanto em sala de aula, quanto fora dela. Os alunos realizaram atividades utilizando uma IDE de programação e um aplicativo de mensagens os ajudaram na aprendizagem.

2.1. Introdução a programação

Foi necessário investigar trabalhos que evidenciem de forma empírica os resultados relativos ao ensino de programação por meio de aplicativos de *m-learning*, ao realizar um mapeamento com as palavras-chaves (“*teaching programming*” or “*m-learning*” or “*mobile learning*”) and “*mobile devices*” na base científica internacional IEEE Xplore, que contém diversas revistas, workshops e conferências entre o ano de 2015 a 2021. Foram encontrados 10 estudos, dos quais apenas três possuem informações sobre o tema que são Ortiz et al. (2015), Amro e Romli (2019) e Utesch et al. (2020).

Ortiz et al. (2015) apresentam uma experiência baseada em *m-learning* no Ensino Superior, que utiliza uma aplicação *web* para a linguagem C e Java que pode ser programada em qualquer dispositivo via navegador.

Amro e Romli (2019), buscaram identificar, dentre os aplicativos disponíveis no Google Play e Apple Store, alguns que pudessem ser utilizados para apoiar técnicas de programação.

Utesch et al. (2020) desenvolveram um jogo sério para *Android* chamado *Pic2Program*, que pode ser utilizado na fase pré-escolar.

A fim de investigar pesquisas no cenário nacional, as mesmas palavras-chaves não obtiveram resultados. Então modificou-se para “*m-learning*” or “*mobile learning*” or “*aprendizagem móvel*” entre os anos de 2015 a 2021. A busca na base dos Anais do Simpósio Brasileiro de Informática na Educação resultou em 16 estudos. Já nos Anais dos Workshops do Congresso Brasileiro de Informática na Educação foram oito estudos encontrados e na Revista Brasileira de Informática na Educação obtivemos quatro estudos, dos quais não foram encontrados trabalhos próximos ou relativos ao ensino de programação via *smartphone* ou *tablets*.

Utilizamos as seguintes palavras-chaves “ensino de programação” and (*m-learning* or “*mobile learning*” or “*aprendizagem móvel*” no Google Acadêmico entre os anos de 2015 a 2021, com o intuito de identificar trabalhos que não haviam sido encontrados nas bases mencionadas anteriormente. Obtivemos 194 resultados e destes se destacaram quatro estudos que utilizaram um ambiente móvel para o ensino de programação: Santana et al. (2017), Ribeiro e Brasileiro (2016), Carvalho et. al (2017) e Andrade (2019). Os demais estudos encontrados tiveram seus focos em aplicações *web* como App Inventor e Scratch, sendo esses artigos aplicados apenas em contextos com uso de computadores.

Santana et al. (2017) tiveram como foco o ensino do aplicativo ScratchJr, uma ferramenta de programação em blocos voltadas para crianças desde os quatro anos até os dez

anos de idade. Eles relataram uma experiência de introdução à programação no ambiente escolar, em que participaram os pais e os alunos em um Scratch Day que é um dia para aprender brincado durante 130 minutos.

Carvalho et al. (2017), em uma turma de Introdução à Programação para 27 alunos, eles usaram uma aplicação chamada *ARDisk* para celular. Esta ferramenta utiliza realidade aumentada com o objetivo de ensinar um único conteúdo: de laços de repetição com *for* e o uso da função *range* em Python.

Ribeiro e Brasileiro (2016) propõe um ambiente híbrido onde os alunos de computação possam utilizar o mesmo servidor da universidade para programar em um computador, *notebook* ou *smartphone*.

Andrade (2019) utilizou o *m-learning* em um projeto na Universidade de Cabo Verde, na África Ocidental. Foi aplicado em duas turmas do 1º ano do curso de Engenharia Eletrotécnica, na disciplina de Introdução à Programação, uma de controle e outra experimental, divididos em três módulos.

Face aos estudos analisados, identificamos que a literatura dentro do que foi pesquisado carece de trabalhos que abordem o ensino de programação por *smartphones* e/ou *tablets*, seja no nível Fundamental, Médio ou Superior.

3. Metodologia

Nesta seção, relataremos uma experiência de ensino de Introdução Programação para alunos do Ensino Superior, utilizando o *smartphone* como recurso principal. Na subseção 3.1 são abordados o objetivo e a métrica utilizada para avaliar os resultados. Na subseção 3.2 são apresentadas as ferramentas utilizadas. Na subseção 3.3 é descrito o método de ensino abordado. Na subseção 3.4 é descrita a implementação do projeto.

3.1. Objetivo e métrica

O objetivo desse relato de experiência é descrever uma estratégia de ensinar a linguagem Python utilizando um *smartphone*, usando como aporte de ensino o construcionismo de Papert, elencando as maiores dificuldades encontradas pelos alunos tanto nos conteúdos como também no uso dos dispositivos móveis e mostrar os benefícios e pontos fracos dessa abordagem.

Para alcançar os objetivos de ensino foi utilizado como base parte do modelo GQM. Foram definidas oito questões de pesquisa (QP) que fazem referência aos conteúdos necessários para que um aluno possa aprender a programar. As questões são:

- **QP1.** Os alunos conseguiram aprender o conteúdo de Estrutura Sequencial? Quais foram as dificuldades encontradas?
- **QP2.** Os alunos conseguiram aprender o conteúdo de Estrutura de Decisão? Quais foram as dificuldades encontradas?
- **QP3.** Os alunos conseguiram aprender o conteúdo de *String*? Quais foram as dificuldades encontradas?
- **QP4.** Os alunos conseguiram aprender o conteúdo de Estrutura de Repetição? Quais foram as dificuldades encontradas?
- **QP5.** Os alunos conseguiram aprender o conteúdo de Lista? Quais foram as dificuldades encontradas?
- **QP6.** Em quais assuntos, dentre os explorados no semestre, os alunos apresentaram maior dificuldade?
- **QP7.** O que foi significativo na utilização da abordagem construcionista de Seymour Papert?
- **QP8.** Qual o impacto causado pelo uso dos smartphones com relação ao uso dos computadores?

Essas questões de pesquisa foram utilizadas para definir o método de aula, o planejamento e criação de atividades, servindo como métrica o percentual de acerto das resoluções dos problemas. Os alunos fizeram as atividades na ferramenta QPython, que foram corrigidas manualmente pelo professor responsável, o que possibilitou responder as questões de pesquisas definidas anteriormente.

3.2. Ferramentas utilizadas

Nas subseções 3.2.1, 3.2.2 apresentamos as ferramentas que auxiliaram no processo de aprendizagem e que foram utilizadas nesta metodologia de ensino.

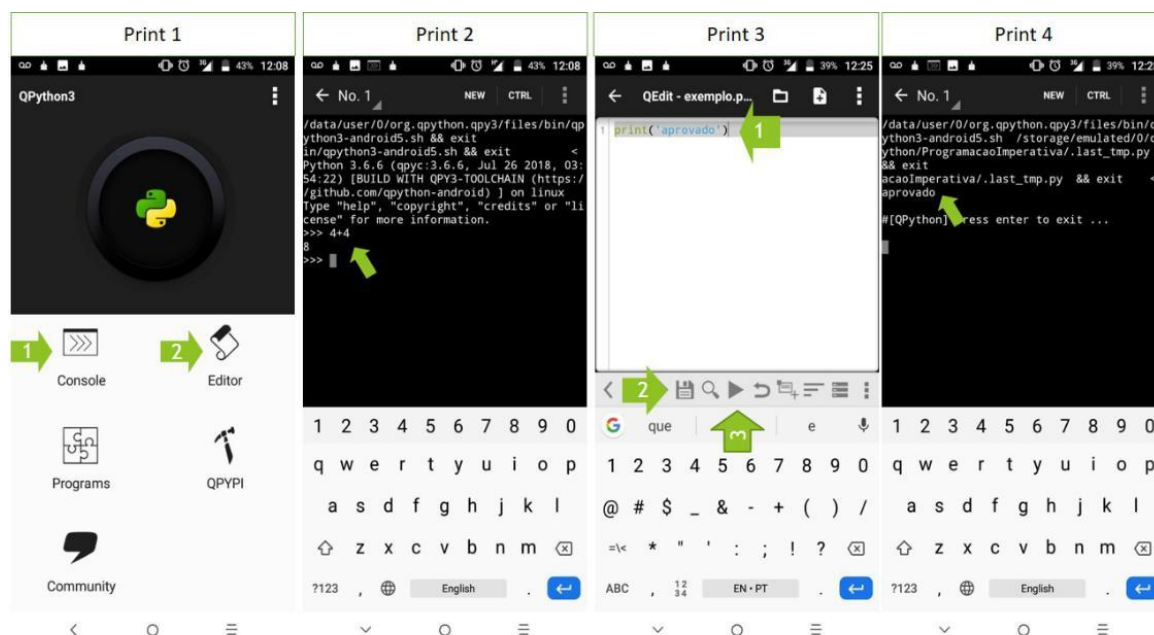
3.2.1. QPython

O QPython é uma IDE Python para o sistema operacional Android. Nela, o usuário pode criar, editar e executar programas em um console disponível no próprio aplicativo. Isso é possível através de um interpretador Python versão 3.2.1 (versão utilizada na época).

O QPython foi desenvolvido pela comunidade QPython e possui código aberto. Para fazer o download da IDE, a partir de dispositivos Android, basta acessar a loja de aplicativos do Google Play.

A figura 1 mostra quatro capturas de tela do QPython. A primeira apresenta a tela inicial onde é possível escolher entre programar diretamente via console ou editar um código fonte e depois testá-lo. A segunda mostra um código sendo executado diretamente via console. Já a terceira mostra um programa simples que pode ser salvo em uma pasta definida pelo usuário. A quarta mostra a execução via console do código feito na terceira captura de tela.

Figura 1. Screenshots de algumas telas do QPython 3L



Fonte: Acervo pessoal

3.2.2. WhatsApp

O WhatsApp é um aplicativo de troca de mensagens instantâneas multiplataforma que possui uma versão *desktop* e *web*. Com esse aplicativo, é possível enviar mensagens de texto, áudio e voz, imagens, vídeos e anexar arquivos. Além disso, também é possível criar grupos de conversas com temas específicos.

Alguns trabalhos utilizaram o WhatsApp como ferramenta de suporte ao ensino em vários níveis escolares, no Ensino Fundamental (Lima et al. 2016), no Ensino Médio, (Lopes e Vaz 2016), como também no Ensino Superior, mais especificamente no ensino de programação (Martins e Gouveia 2019).

Martins e Gouveia (2019) apresentaram uma experiência de utilização do aplicativo WhatsApp para a realização de atividades fora da universidade com foco no ensino de programação. Eles criaram um grupo que serviu como um fórum de dúvidas e respostas em horários pré-estabelecidos e ao final do semestre foi verificada sua eficácia no ponto de vista de 32 alunos que responderam a um questionário online. Constatou-se que o WhatsApp pode ser utilizado de forma efetiva como uma extensão da sala de aula.

3.3. Método de ensino

O método de ensino utilizado foi baseado no construcionismo de Seymour Papert, que define o aluno como protagonista do seu processo de aprendizagem. Em seu livro *A Máquina das Crianças*, é abordado como o sujeito gera conhecimento ao partir da prática do abstrato para a construção do concreto através do uso do computador (Papert, 2008). Ou seja, o professor é responsável por provocar situações em que o aluno aprende a aprender. As circunstâncias criadas podem abordar problemas relacionados a abstração, formulação de hipóteses, reconhecimento de padrões, decomposição dentre outras. Esses conteúdos devem valorizar a vivência diária de cada um (Bem-Ari, 1998)

De acordo com Papert (1980), o construcionismo faz com que os alunos aproveitem ao máximo as suas capacidades criativas, pois com esse método os alunos se sentem e se tornam parte da atividade, seja um jogo, um problema ou, até mesmo, uma brincadeira.

Para Papert, ao utilizar a abordagem construcionista em sala de aula, o professor se torna o mediador responsável por fomentar a aprendizagem dos seus alunos, trazendo atividades e conteúdo de acordo com a realidade ou experiência de cada um.

Ainda segundo Papert, o aluno é um ser pensante capaz de construir suas próprias estruturas cognitivas, ou seja, ele consegue aprender mesmo sem ser ensinado (Papert, 1986).

Um exemplo da aplicação do construcionismo é no trabalho de Santos e Lima (2018), que aplicaram as cinco dimensões de aprendizagem propostas por Papert. Esse estudo foi realizado com alunos do 1º ano de um curso técnico de informática do Ensino Médio, sendo o tema principal abordado o estudo de robótica. Para os autores, o método baseado no construcionismo foi de grande relevância para o aprendizado, visto que os estudantes conseguiram adquirir conhecimento de forma autônoma.

3.4. Relato de experiência

O ambiente no qual o relato de experiência foi aplicado se dá em uma turma da disciplina de Introdução à Programação com carga horária de 60 horas cuja ementa possui os assuntos: entrada e saída de dados, condicionais, *strings*, laços de repetição, listas, matrizes, dicionários e bibliotecas de cálculos matemáticos. A turma contou com a matrícula de 53 alunos, dos quais 26 eram mulheres e 27 homens entre 18 e 22 anos de idade dos cursos de Engenharia, Matemática e Física da Universidade Federal de Sergipe (UFS), durante o semestre de 2019.2 (outubro de 2019 a março de 2020).

No início de cada período é definido um laboratório e/ou uma sala sem computadores. Esta turma ficou em uma sala com um projetor portátil, uma lousa, sem computadores disponíveis. Coube, então ao docente responsável, indicar algum ambiente de programação para que os discentes pudessem praticar em sala de aula e em casa. Cada docente pode escolher a linguagem de programação a ser usada. Coube também ao docente criar algum método de ensino capaz de suprir a falta de recursos. Foi escolhida a linguagem de programação Python que pode ser utilizada através de IDE's em *desktop*, *web* e *mobile*.

Foi explicado aos alunos no primeiro dia de aula como funcionaria o método de ensino e que poderiam praticar programação pelo *smartphone*. Depois foi pedido para que todos aqueles que possuíssem um *smartphone* com sistema operacional Android baixassem o aplicativo QPython. Para aqueles que possuíam iPhone, foi indicado o aplicativo Python3IDE disponível na App Store que realiza a mesma função do QPython.

Após a instalação dos aplicativos em sala de aula, foi disponibilizado um manual contendo as instruções de como configurar a ferramenta no celular e no computador, criar e testar código no console e no editor, além do conteúdo de 135 exercícios individuais que estão

disponíveis na página *web* do Python Brasil⁵, que foram resolvidos em sala de aula. As atividades possuíam três níveis de dificuldade⁶.

Ademais, foram trabalhados oito projetos colaborativos⁷ com foco no construcionismo, que foram de autoria dos próprios alunos, e que foram resolvidos em equipe dentro e fora da universidade. Foi aplicada a teoria construcionista, segundo a qual o aluno se desenvolve melhor ao ser apresentado aos problemas relacionados ao seu cotidiano. De acordo com Brehm et al. (2019), projetos colaborativos motivam os alunos e melhoram seu desempenho em conteúdos relacionados à programação. Desse modo, a turma foi dividida em grupos de cinco ou seis pessoas, para que pudessem resolver os projetos elaborados por eles mesmos.

Por fim, foi informado aos alunos que seria criado um grupo no WhatsApp para suporte acadêmico, o qual não tinha restrição de horário. Esse grupo serviria como um fórum de discussão, cujos estudantes poderiam compartilhar seus conhecimentos, tirar dúvidas, compartilhar código e receber *feedback*.

Foi realizado um piloto, com quatro encontros de duas horas cada, para averiguar se a ferramenta atenderia à necessidade das aulas, bem como testar sua usabilidade na criação, edição e teste dos programas em Python. Durante este período, foram resolvidos no *smartphone* problemas sobre o primeiro conteúdo de estrutura sequencial.

A evolução dos alunos foi perceptível. Conseguiram resolver os problemas propostos e compartilhar os programas em Python, tanto com os colegas do grupo quanto com o professor, por meio do WhatsApp. Depois de identificar dois alunos sem *smartphone* nem computador pessoal, foi permitido que eles utilizassem papel e lápis para fazer os exercícios fora da Universidade. Para poderem praticar em sala de aula, o docente levou dois aparelhos, um *tablet* e um *smartphone*, para que não fossem prejudicados e desistissem da disciplina.

Segundo Papert (1980), é preciso acompanhar cada aluno após a explicação do conteúdo e a cada exercício. Através dessa abordagem, foi possível perceber que cada aluno tem o seu tempo e método de aprendizagem, o que torna necessária a criação de novos exemplos e

⁵Disponível em: <https://wiki.python.org.br/ListaDeExercicios>

⁶Disponível em:

<https://www.dropbox.com/s/wxqxiga4f34shpm/ORGANIZA%C3%87%C3%83O%20DAS%20ATIVIDADES.pdf?dl=0>

⁷Disponível em: <https://www.dropbox.com/s/hxf1jftokhvjzol/PROJETOS%20COLABORATIVOS.pdf?dl=0>

atividades, a depender do perfil do aluno, aplicando assim, uma estratégia de ensino⁸ que possibilitasse o desenvolvimento de todos.

Na primeira unidade foram abordados os assuntos de estrutura sequencial e estrutura de decisão, expressões booleanas, alguns métodos de *string* como *in* e *not in* que retornam valores booleanos e a biblioteca *math* que efetua cálculos matemáticos. Os exemplos de expressões booleanas foram aproveitados ao ensinar sobre operadores lógicos *if*, *elif* e *else*.

Na segunda unidade foi aprofundado o assunto de manipulação de *string* com o ensino de estrutura de repetição. Por ser uma cadeia de caracteres que pode ser percorrida e alterada com algumas funções da própria *string*, foi importante ensinar como se utilizam os laços com *for* e *while*. Foram ensinadas tuplas e listas, assim como sua relação com *strings*, que são listas de caracteres, como indexar, acessar, modificar e excluir itens.

Na terceira unidade foi ensinado como criar uma função e como reutilizá-la dentro do código, tratamento de exceções, listas compostas com dicionários e matrizes.

Ao final foi solicitado aos alunos que desenvolvessem projetos colaborativos que abordassem todo o conteúdo passado em sala de aula, eles ficaram livres para criar um problema e para solucioná-lo, especificado e implementado no QPython para aqueles que utilizavam *smartphones* ou no IDLE, ferramenta que acompanha o Python quando configurado no *desktop* e/ou *notebook* pelos próprios discentes.

Durante as aulas o WhatsApp foi bastante importante para que uma turma sem laboratório tivesse uma dinâmica e interação relevante, já que durante o período letivo os alunos afirmaram ter utilizado dentro e fora da sala de aula o *smartphone* em mais de 80% dos exercícios, 15% utilizaram um *desktop* e/ou *notebook* pessoal ou ambos os recursos e os outros 5% utilizaram um dispositivo móvel e papel para programar.

Quando os alunos precisavam tirar dúvidas referentes a algum código, eles compartilhavam no grupo e o docente transmitia de forma instantânea no projetor. Já fora da sala de aula, as dúvidas que surgiam eram compartilhadas enviando o código diretamente da ferramenta QPython e o professor conseguia dar *feedback* no mesmo momento.

Após o término de cada unidade, era passado aos alunos um questionário online, feito no Google Forms, para que eles pudessem expor de forma anônima as suas dificuldades ou

⁸Disponível em:

<https://www.dropbox.com/s/dzrmf9fbi2qaxuz/ESTRAT%C3%89GIA%20DE%20ENSINO%20UTILIZADA.pdf?dl=0>

facilidades durante o aprendizado de um determinado assunto, foram aplicados quatro questionários.

O primeiro questionário⁹ foi referente aos assuntos de entrada e saída de dados, formatação de saída, concatenação, blocos de condições simples e aninhados, variáveis explícitas e implícitas, operadores lógicos e manipulação de *string*. O questionário foi respondido por 45 alunos. O segundo¹⁰ questionário aplicado foi referente aos assuntos de laços de repetição simples e compostos com *for e while*, tuplas, listas e matrizes. Foi respondido por 38 alunos. Antes da terceira avaliação foi aplicado um terceiro¹¹ questionário que buscou identificar a opinião dos alunos no que diz respeito aos conteúdos de dicionários, funções, tratamento de exceções e listas compostas com dicionários. Foi respondido por 33 alunos.

Ao final do período foi aplicado um último questionário¹² respondido por 31 alunos, que tinha como objetivo coletar informações sobre como o método foi aplicado, sugestões, saber quais recursos os alunos mais utilizaram, o que acharam do acompanhamento no WhatsApp e a opinião deles com relação ao QPython.

Os resultados obtidos serviram como base para comparar as respostas dadas com o que foi produzido em sala de aula. Por fim, comparamos também as informações das turmas anteriores, no que diz respeito à assiduidade, quanto ao total de aprovados, reprovados e a quantos desistiram da disciplina.

4. Apresentação e análise de resultados

Nas subseções a seguir são mostrados os resultados obtidos por meio das atividades respondidas em sala de aula pelos alunos.

4.1. QP1. Os alunos conseguiram aprender o conteúdo de Estrutura Sequencial? Quais foram as dificuldades encontradas?

⁹ Questionário 1: <https://forms.gle/AFPt5BRFRRbvYzNP8>

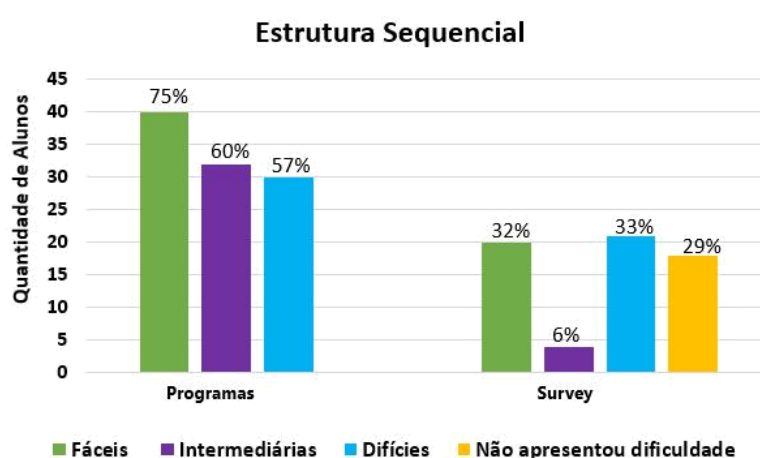
¹⁰ Questionário 2: <https://forms.gle/XoKrUgKjHTVktFnTA>

¹¹ Questionário 3: <https://forms.gle/xrHJdWt7hwuNpoVG9>

¹² Questionário Final: <https://forms.gle/JbsD6qGLEvCuZn49A>

Sobre o conteúdo de Estrutura Sequencial, esperava-se que os alunos aprendessem a criar variáveis implícitas e explícitas, a resolver problemas lógicos sem o uso de estrutura de decisão e a formatar saída de dados. Os alunos acertaram mais questões fáceis e a diferença de acertos das questões intermediárias e difíceis foi pequena. Podemos concluir que a maioria conseguiu colocar em prática o que aprendeu. Na opinião dos alunos, a maior dificuldade encontrada foi na parte difícil do conteúdo, que trata de formatação de saída. A figura 3 mostra os resultados dos 18 exercícios e das 33 respostas obtidas no questionário.

FIGURA 3. Resultados obtidos para o conteúdo de Estrutura Sequencial



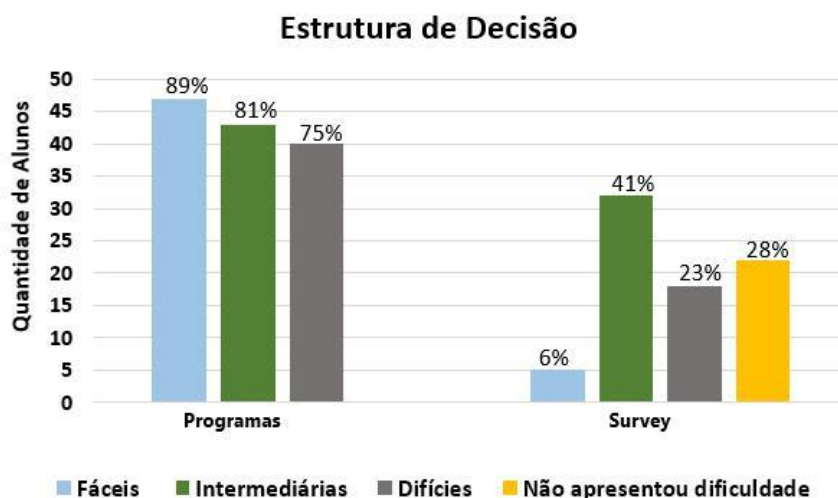
Fonte: Acervo pessoal

4.2. QP2. Os alunos conseguiram aprender o conteúdo de Estrutura de Decisão? Quais foram as dificuldades encontradas?

Sobre o conteúdo de Estrutura de Decisão esperava-se que os alunos aprendessem a criar programas mais elaborados, utilizando os operadores lógicos *and*, *or* e *not* em conjunto com *if*, ou *if e else* ou ainda *if, elif e else*.

No total foram 28 problemas resolvidos pelos estudantes. A figura 4 mostra que houve uma diferença pequena entre os índices de acertos dos três níveis de questões. Na opinião dos alunos, obtida com o questionário, as questões intermediárias foram as que apresentaram maior dificuldade, que envolvem comparações lógicas com *and*, *or* e *not*.

FIGURA 4. Resultados obtidos para o conteúdo de Estrutura de Decisão



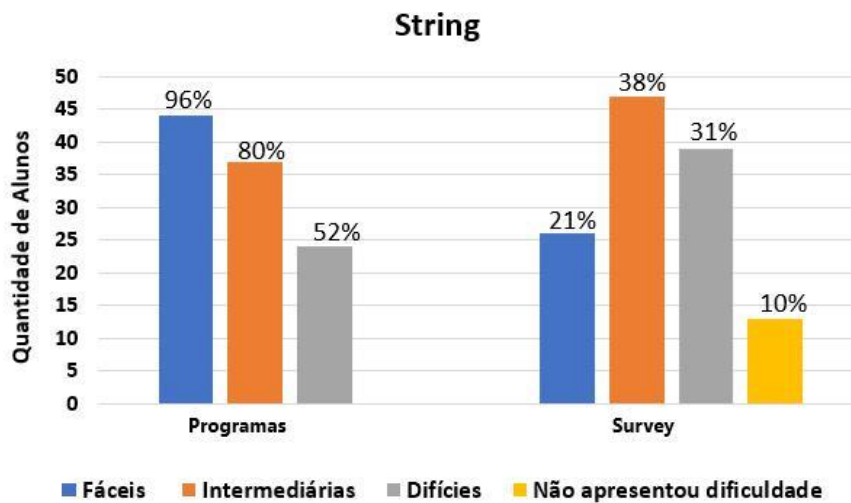
Fonte: Acervo pessoal

4.3. QP3. Os alunos conseguiram aprender o conteúdo de *String*? Quais foram as dificuldades encontradas?

O conteúdo de *String* foi visto em paralelo com uma introdução aos conceitos de Estruturas de Repetição. Sobre o conteúdo de *String*, esperava-se que os estudantes aprendessem como manipulá-las e como alterar texto usando por exemplo operadores de pertencimento, como *in* e *not in*.

A figura 5 mostra os resultados obtidos nos 14 exercícios sobre o conteúdo de *String*. Os alunos acertaram 96% dos problemas fáceis e 80% dos intermediários, demonstrando boa capacidade de interpretação e de colocar em prática o que foi abordado. Os alunos tiveram uma maior dificuldade em manipulação de *String* quando se era necessário interpretar o enunciado das questões visto que nessas questões era necessário decidir quais operadores ou funções eles deveriam utilizar naquele momento. Essas dificuldades observadas pelo professor foram as mesmas relatadas pelos alunos através do questionário, com alguns adendos como por exemplo, o uso da função *split()*.

FIGURA 5. Resultados obtidos para o conteúdo de *String*



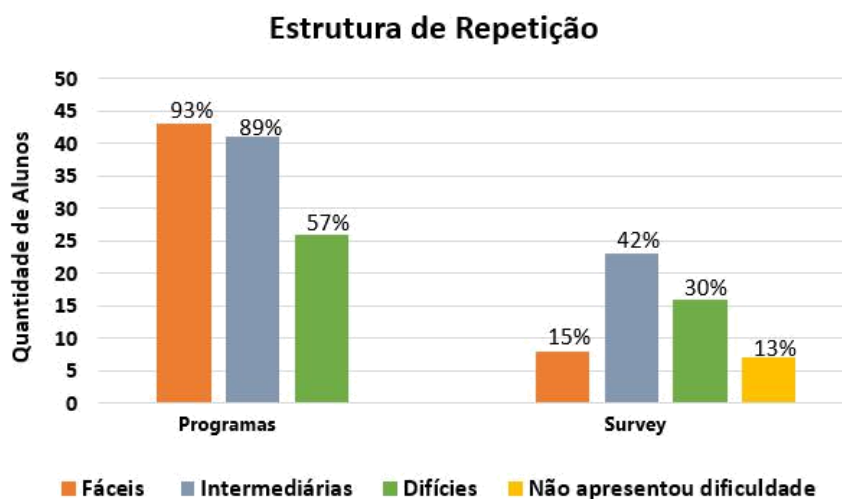
Fonte: Acervo pessoal

4.4. QP4. Os alunos conseguiram aprender o assunto de Estrutura de Repetição? Quais foram as dificuldades encontradas?

O conteúdo de Estrutura de Repetição foi ministrado aproximadamente a partir da metade do período letivo. Nesse momento, esperava-se que os alunos fossem capazes de resolver diversos tipos de problemas, como cálculos matemáticos, manipulação de *strings* e problemas lógicos mais complexos. As estruturas de repetição ensinadas foram *while* e *for*.

A figura 6 mostra os resultados obtidos nos 51 programas referentes a Estrutura de Repetição. Os índices de acerto de questões fáceis e intermediarias foram próximos entre si, mas bem maiores que os das questões difíceis. Ou seja, uma parte considerável dos alunos não conseguiu compreender a parte do conteúdo mais difícil referente ao uso do *while*.

FIGURA 6. Resultados obtidos para o conteúdo de Repetição



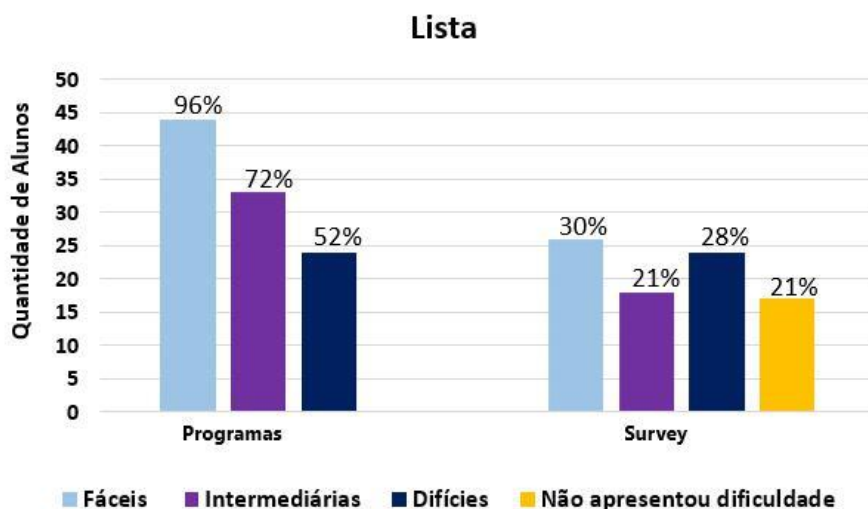
Fonte: Acervo pessoal

4.5. QP5. Os alunos conseguiram aprender o conteúdo de Lista? Quais foram as dificuldades encontradas?

Sobre o último conteúdo abordado, de Listas, esperava-se que os alunos soubessem criar, inserir, excluir e alterar elementos de uma lista. Esperava-se também que tivessem compreendido e soubessem utilizar todos os conteúdos anteriores. A figura 7 mostra os resultados dos 24 programas sobre Lista e do questionário.

Foi atingido um ótimo índice de 93% de acertos nas questões fáceis, o que significa que os alunos aprenderam a ideia principal desse conteúdo. Tiveram, porém, dificuldades nas questões difíceis, que abrangem os métodos *reverse()*, *copy()*, *replace()* e *extend()*. Houve também dificuldade em aprender o conteúdo de Matrizes. No questionário, os alunos afirmaram que tiveram mais dificuldade em aprender a parte do conteúdo classificada como fácil, que inclui inserir, remover e atualizar um determinado elemento em uma posição.

FIGURA 7. Resultados obtidos nas atividades de Lista



Fonte: Acervo pessoal

4.6. QP6. Os assuntos que os alunos mais tiveram dificuldade?

Quanto à questão de pesquisa QP6, os assuntos que os alunos mais tiveram dificuldade foram laço de repetição com *while*, manipulação de *string* e aplicação de matrizes. Acredita-se que, por serem conteúdos mais abstratos, tenha-se gerado um pouco mais de complexidade.

4.7. O que houve de significativo na utilização da abordagem construcionista de Seymour Papert?

Os resultados obtidos com a aplicação do método construcionista são animadores, mesmo precisando de ajustes em turmas futuras. Encerrou-se a matéria com uma taxa de aprovação de 83%, 11% foram reprovados por falta ou média, 4% reprovados por média e apenas 2% de evasão. Ao final do curso, os alunos responderam um questionário no Google Forms, onde relataram suas dificuldades e deram a sua opinião sobre o método de ensino. Alguns relatos são transcritos abaixo:

- **Aluno 1:** “Muito boa e estratégica ao deixar com que os alunos fossem os principais responsáveis por sua aprendizagem.”

- **Aluno 2:** “Instigante. O professor oferecia todo suporte necessário ao aprendizado dos alunos. Estando disponível para esclarecer dúvidas e ajudar quem estava interessado, incluindo em horário extraclasse. Sempre pronto para ajudar.”
- **Aluno 3:** “Muito diversificada, na questão de ter mais de uma forma de entender os assuntos, e bastante dinâmica e prática com exercícios que aplicamos diariamente”
- **Aluno 4:** “Um método novo, criativo e de fácil entendimento que resultou em aulas descontraídas e leves.”
- **Aluno 5:** “Definitivamente um método maravilhoso, extremamente pertinente e didática, conseguindo através de jogos e listas facilitar o aprendizado e fixar o assunto com mais facilidade.”
- **Aluno 6:** “A opção de utilizar o Python pelo celular foi importante pela facilidade. Pelo celular, com um clique era possível enviar o código para qualquer pessoa e a interação é maior e mais rápida. O método adotado foi muito bom, desde o primeiro dia de aula, programando o básico na folha de papel.”

Na figura 8 é possível ver os alunos programando em equipes e usando os *smartphones*. Além disso, ainda temos um dos discentes utilizando um *notebook*.

FIGURA 8. Alunos programando no QPython



Fonte: Acervo pessoal

Do ponto de vista construcionista, a capacidade do professor em saber mitigar as dificuldades por meio de projetos, explicações e gincanas que permitissem ao aluno entender o

conteúdo, estimulando cada um a ser o principal responsável por sua aprendizagem, pode ter influenciado nos resultados positivos apesar da falta de recursos.

4.8. Qual o impacto causado pelo uso dos smartphones com relação ao uso dos computadores?

Sobre o uso de dispositivos móveis, os discentes informaram que 80% deles utilizaram um *smartphone* ou *tablet* e 20% utilizaram *notebooks*.

Ainda de acordo com eles, o uso dos *smartphones* possibilitou que pudessem praticar questões de programação com maior frequência, quer fosse no ônibus, no pátio da universidade ou durante as aulas. Além disso, a facilidade de poder compartilhar o código com o professor pelo WhatsApp e tirar dúvidas possibilitou um melhor desenvolvimento.

A análise que foi feita durante o acompanhamento no dia a dia de aula é de que muitos alunos tinham facilidade na digitação pelo celular, de maneira que eles conseguiam ser tão rápidos quanto os alunos que utilizavam um *notebook*.

O impacto causado pelo o uso dos smartphones na sala de aula nesse estudo foi extremamente positivo e ao contrário do que se possa pensar, não houve desvio de atenção durante as aulas principalmente por ser uma matéria prática e que requer bastante atenção. O engajamento por parte dos alunos trouxeram resultados de aprovação animadores

Vale ressaltar que, é muito importante não tornar os aparelhos móveis o centro das atenções dentro do ensino, mas deve-se ter a preocupação de inserir esse recurso como forma de atingir a inclusão social e aprendizagem de todos.

5. Considerações finais

Este artigo explora um método de ensino que utiliza o *smartphone* como recurso tecnológico para abordar conteúdos introdutórios de programação e relata a experiência adquirida neste período. Utilizar os desafios disponíveis na plataforma Python Brasil e projetos com contexto relacionado ao cotidiano dos alunos, conforme a teoria construcionista, apresentou ótimos resultados na aprendizagem dos alunos.

Concluimos que o método obteve êxito apesar de não haver estrutura para a prática de programação. Os resultados encontrados evidenciam que o método pode ser utilizado no ensino introdutório de programação no Ensino Superior.

A construção desses resultados só foi possível graças ao engajamento dos alunos na disciplina. Acredita-se, pelos comentários dos alunos, que os projetos colaborativos, os desafios, as gincanas e o suporte via WhatsApp fora da universidade aos discentes tenham gerado esse resultado.

Quanto ao QPython, foi possível observar que, como os alunos tinham uma habilidade com os *smartphones*, a diferença de tempo que um aluno levava para fazer um código pelo o computador e pelo celular quase não era perceptível. Já com relação ao WhatsApp é preciso destacar que não houve problemas e que, pode servir como ferramenta de suporte para a educação em disciplinas de computação.

A experiência vivida e os índices apresentados mostram que é possível ensinar e aprender utilizando apenas um *smartphone*. Contudo, foi necessário empenho por parte do professor, pois se tratava de uma turma com 53 alunos e acompanhamento individual por conta da abordagem construcionista. Estar atento a cada dificuldade exposta pelos alunos, analisá-las e preparar estratégias para sanar as dúvidas durante as aulas e plantões, mostrou-se uma abordagem importante para minimizar as desistências na disciplina e no curso.

Buscamos não somente trazer uma nova perspectiva de ensino, mas também mostrar sua importância ao praticar inclusão social e digital, permitindo que todos os envolvidos tivessem acesso à aprendizagem com os recursos disponíveis.

Como trabalhos futuros, estamos ampliando a estratégia de ensino voltada para o EAD devido à pandemia. Pretendemos, também, aplicar o novo método em turmas do Ensino Médio assim que as aulas retornarem. Esperamos conseguir utilizar três plataformas diferentes para o ensino de programação, IDE em *desktop*, *web* e *mobile*, em uma mesma turma, empregando o mesmo método e, com isso, comparar os resultados.

Agradecimentos

Os autores agradecem a todos os alunos da disciplina de Programação Imperativa.

Referências

- AGÊNCIA BRASIL. Sete em cada 10 brasileiros usam a Internet. AgênciaBrasil, 2019. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2019-08/sete-em-cada-dez-brasileiros-acessam-internet-diz-pesquisa>. Acesso em: 30/09/2019.
- AMRO, J. S. e ROMLI, R. Investigation on the learning programming techniques via mobile learning application. *In: 2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering*, IEEE, 2019, p. 1-7.
- ANDRADE, E. A. **Mobile Learning e Programação: O desenvolvimento de apps como metodologia para a aprendizagem da programação em contexto universitário**. 2019. 373 p. Tese de Doutorado - Universidade de Lisboa, Portugal, 2019.
- BASILI, V. R. e WEISS, D. M. A methodology for collecting valid software engineering data. *IEEE Transactions On Software Engineering*, n. 6, p. 728-738, 1984.
- BEN-ARI, Mordechai. Constructivism in computer science education. *Acm Sigcse Bulletin*, v. 30, n. 1, p. 257-261, 1998.
- BORBA, C. M. e LACERDA, H. D. G. Políticas públicas e tecnologias digitais: Um celular por aluno. **Educação Matemática Pesquisa: Revista do Programa de Estudos Pós-Graduados em Educação Matemática**, v. 17, n. 3, p. 490-507, 2015.
- BREHM, L. et al. Collaborative learning with COZMO to teach programming in SCRATCH and Python. *In: 2019 IEEE Global Engineering Education Conference*, IEEE, 2019, p. 448-452.
- CARVALHO, M. F. et al. F. Ensino da estrutura de repetição for em Python com realidade aumentada através do Aurasma. *In Anais do Congresso sobre Tecnologias na Educação*, Mamanguape, PB, Brasil, 2017, p. 443-454.
- INEP. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira: **Censo escolar**. INEP, 2021. Disponível em: https://download.inep.gov.br/censo_escolar/resultados/2021/apresentacao_coletiva.pdf. Acesso em: 01/02/2022.
- LIMA, M. et al (2016). Experiência de aprendizagem móvel: o uso do WhatsApp no Ensino Fundamental. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, 2016, p. 820.

LOPES, G. C. e VAZ, B. B. O Uso Pedagógico dos grupos do Whatsapp no Ensino de História. *In V Congresso Internacional de História - Novas Epistemes e Narrativas Contemporâneas*, Goiás, Brasil, 2016.

MARTINS, E. e GOUVEIA, L. Uso do WhatsApp em atividades educativas extraclasse online no Ensino de Programação. *In Anais do XXVII Workshop Sobre Educação em Computação*, Porto Alegre, Brasil, 2019. p. 141–150.

MOHAMMAD, S. Effectiveness of M-Learning in Blended Learning-Design of Prototype Framework for AOU Bahrain. In IEEE (Ed.), 2015 **FIFTH International Conference On E-learning**. Manama, Bahrain: IEEE Xplore, 201, p. 201–206

NIELSEN. Brasileiros com Internet no smartphone já são mais de 70 milhões. Nielsen, 2015. Disponível em: <https://www.nielsen.com/br/pt/press-releases/2015/brasileiros-com-internet-no-smartphone-ja-sao-mais-de-70-milhoes/>. Acesso em: 27/09/2019.

ORTIZ, O. *et al.* M-learning tools: The development of programming skills in engineering degrees. **IEEE Revista Iberoamericana de Tecnologias del Aprendizaje**, 2015. v. 10, n. 3, p. 86–91.

PAPERT, S. *Mindstorms*. New York: Basic Books, 607, 1980.

PAPERT, S. *Logo: computadores e educação*. São Paulo: Brasiliense, 1986.

PAPERT, S. *A máquina das crianças: repensando a escola na era da informática*. Porto Alegre: Artmed, 2008.

RIBEIRO, M. M. B. e BRASILEIRO, T. S. A. Ambientes virtuais híbridos. Uma possível solução para ensino de programação de computadores em uma universidade amazônica. *In XVIII Simpósio Internacional de Informática Educativa*, 2016, Vol. 15, No. 1, p. 173.

ROMERO, L. M. M. Methodology with Python Technology and Social Network Analysis Tools to Analyze the Work of Students Collaborating in Facebook Groups. In IEEE (Ed.), 2019 **14th Iberian Conference On Information Systems and Technologies**, Coimbra, Portugal: IEEE Xplore, 2019, p. 1–6.

SANTANA, A. L. M. et al. Scratch Jr-Brincando em Família: um relato de experiências sobre introdução a programação no ambiente escolar. *In: Anais do XXIII Workshop de Informática na Escola*, SBC, 2017, p. 432-441.

SANTOS, G. J. T. e LIMA, S. J. F. Robótica educacional e construcionismo como proposta metodológica para o desenvolvimento de ambientes de aprendizagem significativa. **Revista Novas Tecnologias na Educação**, 2018, Vol. 16, No. 2, p. 1–10.

SOTOMAYOR-BELTRAN, C. First year undergraduate electronic engineering students from a peruvian university learn to program with Python. *In* **2019 IEEE World Conference on Engineering Education**, 2019, p. 1–4.

UTESCH, M. C. et al. Pic2program - an Educational Android application teaching computational thinking. *In* **2020 IEEE Global Engineering Education Conference**, 2020, p. 1493-1502.

WEST, M. e VOSLOO, S. Diretrizes de políticas para a aprendizagem móvel, **UNESCO DOC Digital Library**, 2014.