

## Uma Abordagem Contextualizada para o Ensino de Programação na Educação Profissional em Informática

Luis Gustavo de Jesus Araujo<sup>1</sup>, Roberto Almeida Bittencourt<sup>1</sup>,  
David Moises Barreto dos Santos<sup>1</sup>

<sup>1</sup> UEFS – Universidade Estadual de Feira de Santana  
Av. Transnordestina, s/n, Novo Horizonte  
Feira de Santana – BA, Brasil – 44036-900

luisaraujo.ifba@gmail.com, roberto@uefs.br, davidmbs@uefs.br

**Abstract.** *Brazilian vocational high school education in Computing faces problems such as course failure and dropout, plus dropout from the programs, factors that directly contribute to the deficit of such professionals. This paper presents an experience report of a contextualized and spiral approach carried out with students of the technical program in Informatics. The approach aims to facilitate learning and motivate students through the tools JES and PPlay, and the Python language. Learned lessons suggest the adequacy of Python as first language, and of contextualized activities, context reuse and challenges as factors contributing to motivation and learning.*

**Resumo.** *A modalidade do Ensino Profissional na área de Informática enfrenta diversos problemas tais como reprovação e evasão em disciplinas, além de abandono dos cursos, fatores que impactam diretamente no déficit de profissionais nesta área. Este artigo apresenta um relato de experiência de uma abordagem contextualizada e em espiral realizada com estudantes do curso técnico em Informática, na modalidade Integrada. A abordagem tem como objetivo facilitar a aprendizagem e motivar os estudantes, através das ferramentas JES e PPlay e da linguagem Python. As lições aprendidas sugerem a adequação de Python como linguagem para iniciantes, de atividades contextualizadas, de reuso de contextos e desafios como fatores que contribuem para a motivação e a aprendizagem.*

### 1. Introdução

A modalidade de Ensino Profissional de Informática enfrenta diversos problemas ligados à permanência dos estudantes nos cursos e ao prosseguimento dos egressos na área de formação, atuando como técnicos [Observatório Softex 2010]. O relatório da SOFTEX aponta que os estudantes que cursam o ensino profissional, na modalidade integrada, dão continuidade aos estudos ao invés de trabalharem como técnicos, provocando um déficit de profissionais da área de Computação.

Ao desinteresse após a formação, somam-se os altos índices de evasão e reprovação em disciplinas relacionadas a matemática e programação [Machado et al. 2006, Cravo 2012]. Quanto à programação, percebe-se que, em grande medida, estes problemas se relacionam com a dificuldade em aprender a programar a partir de uma metodologia tradicional onde apresenta-se o conteúdo e os estudantes

devem resolver listas de exercícios. Estes exercícios são, em sua maioria, dissociados das vidas dos alunos, o que os leva a achar a programação enfadonha e excessivamente técnica [Guzdial 2003].

Deve-se, no entanto, considerar o perfil destes estudantes e trazer para a sala de aula contextos que lhes despertem interesse. Os estudantes dessa modalidade interagem, cada vez mais, nas plataformas de redes sociais, imergem em jogos complexos e utilizam aplicativos de imagens e música. Como afirma Prensky (2001), estes estudantes já nasceram em meio a um turbilhão tecnológico e interagem diariamente com estas mídias.

Diversas ferramentas e linguagens têm sido desenvolvidas para possibilitar a adoção desses contextos no ensino de programação e, ao mesmo tempo, suprir as necessidades dos programadores iniciantes. Exemplos destas ferramentas são: Scratch, Logo, Kturtle, App Inventor, Lego Mindstorms, Jython Environment for Students (JES) e PPlay [Bordini et al. 2016, Guzdial 2003, IC-UFF 2017]. Associados a estas ferramentas, diversos estudos têm sido realizados utilizando histórias, simulações, robótica, jogos [Rebouças et al. 2010, Marques et al. 2011] e mídias [Araujo et al. 2017, Santana et al. 2017] para motivar os estudantes e minimizar estas dificuldades. Tais ações contribuem potencialmente para a aprendizagem dos estudantes, para o aumento do número de concluintes e, de certo modo, incentivam os estudantes a prosseguir na área de formação.

Apesar da variedade de iniciativas, pouco se tem realizado sobre o uso de mídias na educação profissional brasileira. Percebemos, neste cenário, uma possibilidade de exploração do contexto de mídias em associação com uma abordagem educacional que vise reduzir a sobrecarga cognitiva e apresentar conceitos de forma gradual, como o ensino em espiral [Bruner 1966].

Esse trabalho apresenta um relato de experiência de uma abordagem de ensino-aprendizagem de programação desenvolvida em uma turma de terceiro ano do Curso Técnico em Informática na modalidade integrada. Visando a motivação e aprendizagem dos estudantes, a abordagem tem como premissa a contextualização das atividades, o uso de ambientes adequados aos iniciantes e uma abordagem em espiral. Nesta abordagem, os conteúdos são apresentados mais de uma vez, em níveis cada vez mais complexos, através de blocos contextualizados que associam um contexto a uma ferramenta.

## **2. Trabalhos Relacionados**

O uso de contextos específicos no ensino de programação é frequente na literatura. Diversos trabalhos apresentam o uso de jogos, robótica, automação e simulação, dentre outros [Bordini et al. 2016]. Neste trabalho, analisamos dois contextos em especial: jogos e mídias.

O desenvolvimento de jogos é um contexto utilizado em uma experiência de concepção de um curso para o ensino de programação [Rebouças et al. 2010]. Os autores observaram que a linguagem Python, combinada com a biblioteca PyGame – uma biblioteca para o desenvolvimento de jogos, oferece uma combinação efetiva para o uso de jogos no ensino de programação.

Marques et al. (2011) apresentam a aplicação de uma oficina no ensino médio com o desenvolvimento de jogos como elemento motivacional para atrair os alunos e au-

mentar o interesse em computação. Os autores concluem que a oficina contribuiu para aumentar o interesse dos alunos do ensino médio pela área de Computação e que os participantes ficaram satisfeitos com a abordagem.

No Brasil, não existem muitos trabalhos relatando o uso de mídias (imagens e sons) para o ensino de programação na Educação Profissional. No entanto, podemos citar trabalhos desenvolvidos em outras modalidades de ensino, como o que os autores do presente artigo realizaram na educação básica [Araujo et al. 2017]. Trata-se de um estudo de caso com estudantes do nono ano do Ensino Fundamental II, iniciantes em programação. O estudo utiliza uma abordagem em espiral, dividida em dois blocos contextualizados, usando Figuras com a biblioteca Turtle e Imagens com o ambiente JES, ambos com a linguagem Python. Os resultados apontam alta motivação entre os estudantes, especialmente nas dimensões de atenção e confiança. O trabalho apresenta mudanças significativas quanto à percepção dos estudantes de como os programas funcionam, o que programadores fazem, e o caráter divertido da computação e da manipulação de imagens.

Mídias também são utilizadas em uma abordagem no ensino superior, em um curso de Engenharia Civil, na disciplina Introdução a Ciência da Computação [Santana et al. 2017]. Este estudo de caso investiga como uma abordagem com ambientes apropriados aos iniciantes e contextos próximos a eles podem influenciar na motivação. A abordagem é dividida em três unidades, utilizando Scratch, Turtle (Python) e JES, respectivamente, em cada uma.

Além disso, outros trabalhos, fora do Brasil, utilizam mídias no ensino de programação para estudantes tanto de cursos de graduação em computação [Simon et al. 2010] como de outros cursos [Guzdial 2003, Guzdial 2013]. Percebe-se que o uso da linguagem Python, além de oferecer suporte adequado aos iniciantes, possibilita a adoção de diversos contextos, como jogos e mídias, relevantes na vida dos estudantes.

### 3. Metodologia

Esta seção descreve as ferramentas e linguagem utilizadas, os participantes, o planejamento da abordagem e os procedimentos para coleta e análise de dados.

#### 3.1. Linguagem e Ferramentas

**Python** é uma linguagem adequada às necessidades de programadores iniciantes, em especial na educação profissional, pois, além de ser adequada ao ensino, é popular e utilizada comercialmente. Segundo Grandell et al. (2006), Python é uma linguagem de alto nível, originalmente criada para facilitar a aprendizagem e inicialmente sugerida como alternativa para o ensino de programação pelo seu criador, Guido Van Rossum. Python tem muitas características para ser uma linguagem adequada ao ensino de programação, como sintaxe clara e reduzida, tipos dinâmicos, semântica expressiva, *feedback* imediato, reforço de estrutura e projetos *open-source* relevantes.

**JES – Jython Environment for Students** é um ambiente desenvolvido pelo Georgia Tech para o ensino de programação através de manipulação de mídias [Guzdial 2003]. JES possui módulos para a manipulação de imagens, sons, vídeos, sites e um módulo do Turtle, todos em Python. Suas funções são simples, de modo a permitir a manipulação destas mídias com poucas linhas de código, o que possibilita aos estudantes iniciantes

na programação o desenvolvimento de atividades relevantes, e.g., efeitos em imagens, criação de figuras, criação e efeitos sonoros, edição de vídeo e criação de sites.

**PPlay** é um framework desenvolvido pela Universidade Federal Fluminense para o ensino de programação através do desenvolvimento de jogos, utilizando Python [IC-UFF 2017]. PPlay utiliza como base o PyGame. No entanto, diferentemente deste, PPlay tem um foco no ensino e, por este motivo, possui comandos mais simples, visando minimizar a sobrecarga cognitiva durante o processo de aprendizagem.

### 3.2. Participantes

O estudo aqui apresentado está sendo realizado em uma escola da Educação Profissional no município de Feira de Santana, Bahia, em uma turma do curso Técnico em Informática. A abordagem foi aplicada no terceiro ano, dentro da disciplina Linguagem de Programação 2, que propõe ensinar o paradigma de programação orientada a objetos (POO). Este estudo conta com 10 estudantes: 6 meninos e 4 meninas, com idade média de 16,5 anos (desvio padrão de 0,9) e com experiência inicial em programação. O professor da disciplina acompanhou as aulas, que foram ministradas pelo primeiro autor deste artigo e apoiadas por um monitor.

### 3.3. Planejamento da Abordagem

A abordagem proposta objetiva ensinar conceitos de programação, buscando facilitar a aprendizagem e aumentar a motivação dos estudantes em relação à programação. Por isso, utilizamos ferramentas e linguagens adequadas a estes estudantes, através de uma abordagem em espiral [Bruner 1966]. Planejamos o uso das seguintes ferramentas: JES, para a edição de Imagens, e PPlay, para a criação de jogos, ambos utilizando a linguagem Python. Dividimos o planejamento em três blocos. Atualmente o estudo encontra-se no terceiro bloco programado. A Tabela 1 apresenta uma visão geral sobre a abordagem.

**Tabela 1. Síntese do Planejamento**

<b>Local</b>	Colégio Estadual Doutor Jair Silva, Feira de Santana, Bahia
<b>Participantes</b>	10 Estudantes
<b>Blocos</b>	Imagens I – Efeitos em Imagens usando POO
	Imagens II – Efeitos em Imagens aprofundando POO
	Jogos – Desenvolvimento de Jogos
<b>Conteúdo</b>	Funções, Parâmetros, Estruturas de Repetição e Seleção, Variáveis, Contantes, Operadores Aritméticos, Operadores Relacionais, Matrizes, Classes, Objetos, Métodos, Herança, Composição, Sobrecarga e Sobrescrita.
<b>Ferramentas</b>	Linguagem Python, JES e PPlay
<b>Duração</b>	57 horas (3h por semana)
<b>Período</b>	11/04/2017 a 29/08/2017

Os três blocos que compõem este estudo apresentam conceitos de POO e de programação repetidas vezes, em níveis cada vez mais profundos e complexos. Por exemplo, uma Estrutura de Seleção simples (*if*) pode ser apresentada num primeiro momento; no segundo, uma Estrutura de Seleção encadeada (*if – else, if – elif – else*); num terceiro momento, uma Estrutura de Seleção aninhada e formas mais complexas utilizando combinação de operadores relacionais e lógicos.

O primeiro bloco apresentou conceitos relacionados à programação de modo geral (Matrizes, Repetição e Estrutura de Seleção) e conceitos de POO (Classe, Objeto,

Atributo e Método). No segundo bloco, buscou-se reapresentar os conceitos gerais sobre programação e possibilitar não só o uso de Classes, Métodos e Atributos, mas também a criação de Classes, assim como o uso de Herança, Composição, Construtores e Sobrecarga. As Tabelas 2, 3 e 4 apresentam os blocos com os objetivos de cada aula, o conteúdo e as atividades. Os materiais utilizados na abordagem podem ser consultados no web site de apoio <sup>1</sup>.

**Tabela 2. Planejamento das Aulas de LP2 - Bloco I**

<b>Dia</b>	<b>Objetivo</b>	<b>Conteúdo</b>	<b>Atividade</b>
<b>01</b>	Construir conhecimentos sobre imagens digitais, Objetos, Classe e aplicá-los no JES	Padrão RGB, Imagens Digitais, Classes, Objetos, Funções do JES (get/setRed, get/setGreen, get/setBlue, pickAFile, getPixel, setColor), Classes do JES (Color, Picture, FileChooser)	Criar figuras geométricas como pixels (Quadrado, Triângulo e Círculo).
<b>02</b>	Construir conhecimentos sobre Matriz e Estrutura de Repetição	Decomposição do código, Array, loop (for), funções in, range e função JES (write)	Criar figuras (Quadrado, Triângulo e Círculo) e efeito degradê com loops
<b>03</b>	Aplicar os conceitos de Matriz e Estrutura de Repetição à manipulação de imagens	Array, loop (for), Operadores aritméticos Funções do JES já apresentadas e getPixels.	Criar efeitos matiz (mudança de tom), Negativos e Escala de Cinza
<b>04</b>	Construir conhecimentos sobre Estruturas Condicionais e aplica-los à manipulação de imagens	Estrutura condicional (if) Operadores relacionais e Funções do JES já apresentadas e getWidth, getHeight	Criar efeitos em metade da imagem e efeito Preto e Branco
<b>05</b>	Aplicar conhecimentos construídos no bloco	Todos os conteúdos apresentados	Criar o efeito Chomakey com fundo verde
<b>06</b>	Aplicar conhecimentos construídos no bloco	Todos os conteúdos e conceitos apresentados	Realizar avaliação escrita

### 3.4. Coleta e Análise de Dados

Em paralelo a esta intervenção, uma pesquisa exploratória está sendo realizada. Seus resultados fogem ao escopo deste relato de experiência, mas os instrumentos de coleta e análise de dados qualitativos e quantitativos utilizados na pesquisa apoiam a construção deste relato.

Quanto aos procedimentos de coleta de dados quantitativos, estão sendo utilizados: i) questionário pré-intervenção; ii) questionários pós-bloco; iii) questionário pós-intervenção. Assim, ao final de cada bloco, aplicamos um questionário com o intuito de medir as mudanças de conceitos e a motivação dos estudantes. Após a realização de todos os blocos programados, utilizaremos um questionário pós-intervenção. Os resultados quantitativos serão analisados através de estatísticas descritivas e inferenciais. Serão utilizados ainda dados da prova escrita e os códigos desenvolvidos em sala. Todos os 10 estudantes responderam ao questionário pré-intervenção.

Para a coleta de dados qualitativos, utilizamos: i) entrevistas semiestruturadas com os estudantes; ii) entrevista semiestruturada com o professor; iii) diários de bordo e iv) observações realizadas durante a disciplina. Quanto à análise qualitativa, está sendo utilizada análise de conteúdo, iniciando com codificação aberta, seguida por codificação axial e, por fim, escrita de *memos* analíticos para facilitar a interpretação dos resultados.

<sup>1</sup>[http://luisaraujo.github.io/programacao\\_com\\_midias/poo/](http://luisaraujo.github.io/programacao_com_midias/poo/)

**Tabela 3. Planejamento das Aulas de LP2 - Bloco II**

<b>Dia</b>	<b>Objetivo</b>	<b>Conteúdo</b>	<b>Atividade</b>
<b>01</b>	Aprofundar conhecimentos Objetos e Classe	Conceito POO , Classes e Objetos	Criar aplicações matemáticas utilizando classes e objetos
<b>02</b>	Construir conhecimentos sobre Construtores, Parâmetros e aprofundar conhecimentos sobre Estruturas Condicionais	Construtores, Parâmetros, Método e Condicionais	Remodelar as Classes anteriores utilizando parâmetros e criar métodos de efeitos com condicionais
<b>03</b>	Construir conhecimentos sobre Composição e aplica-los à manipulação de imagens	Conceitos de POO (Composição), Repetição, Seleção e Funções JES já apresentadas	Utilizar composição em classes para criar efeitos em imagens
<b>04</b>	Construir conhecimentos sobre Herança e aplica-los à manipulação de imagens	Conceitos de POO (Herança) Repetição, Seleção Funções JES já apresentadas	Criar uma classe que herda métodos e atributos da Classe Picture
<b>05</b>	Construir conhecimentos sobre Estrada e Saída e aplica-los à manipulação de imagens	Herança, Funções JES (addText), Repetição, Instruções de Entrada e Saída	Criar uma classe para criação de Memes
<b>06</b>	Aplicar conhecimentos construídos no bloco	Todos os conteúdos e conceitos apresentados	Realizar avaliação escrita

#### 4. Resultados

No primeiro bloco contextualizado, foi apresentado o ambiente JES e a possibilidade de editar imagens através da programação. O professor da disciplina apresentou os principais itens da GUI do JES e iniciou com a apresentação das funções iniciais (para obter uma imagem e exibi-la). Este momento inicial é bastante expositivo e os estudantes repetiram os comandos utilizados pelo professor.

Embora imagens sejam algo próximo dos estudantes, fez-se necessário a apresentação de conceitos relacionados à representação de uma imagem no computador. Assim, conceitos sobre o padrão RGB e a imagem enquanto uma matriz de *pixels* foram abordados. Os estudantes apresentaram facilidade quanto a estes elementos e quanto às funções iniciais.

Como primeira atividade, foi pedido aos estudantes que eles modificassem alguns *pixels* de uma imagem. Em seguida, foi solicitado que eles criassem algumas figuras geométricas como quadrado, triângulo e círculo. Nesta aula, alguns estudantes, que já haviam utilizado Python, usaram estruturas de repetição para gerar as imagens, mesmo sendo este um assunto da aula seguinte.

No segundo dia, os estudantes utilizaram os códigos da primeira aula para decompor e adicionar estruturas de repetição, além de criarem figuras com o efeito de Degradê, usando fórmulas matemáticas. Os alunos não tiveram dificuldades em usar conceitos matemáticos simples como uma soma ou subtração dentro de uma estrutura de repetição. Esta atividade possibilitou aos alunos uma maior liberdade quanto a modificar os códigos, gerando vários efeitos de Degradê. Dois estudantes adicionaram uma função aleatória, gerando um degradê com pontos.

No terceiro e no quarto dias, os estudantes passaram de criar imagens em uma tela

**Tabela 4. Planejamento das Aulas de LP2 - Bloco III**

<b>Dia</b>	<b>Objetivo</b>	<b>Conteúdo</b>	<b>Atividade</b>
<b>01</b>	Aprofundar conhecimentos sobre Objetos, Classes e Métodos aplicando no PPlay	Objetos, Classe, Métodos e Funções do PPlay (Sprite, Window) e GameLoop com While	Iniciar a criação do Jogo Pong, com a criação da bola e barra
<b>02</b>	Aplicar conhecimentos sobre Condicionais e uso de Variável em Jogos	Estrutura de Seleção Simples e Aninhada, Operadores Relacionais e Variáveis	Cria score, condição de vitória e exibição do Placar.
<b>03</b>	Aprofundar conhecimentos sobre Repetição e uso de Estruturas de Dados Simples	Estrutura de Repetição, Array, Operadores Relacionais e Funções do PPlay (Animation)	Iniciar o Jogo T-Rex
<b>04</b>	Aplicar e aprofundar conhecimentos sobre POO	Criação de Classes com Atributos, Métodos e Construtor	Refatorar os comandos, criando classes, métodos e atributos do Jogo T-Rex
<b>05</b>	Refatorar código, utilizando Herança e Composição	Usando Herança e Composição	Organizar as classes visando reaproveitamento de código
<b>06</b>	Aplicar conhecimentos construídos em projetos de livre escolha	Todos os conteúdos e conceitos apresentados	Criar um jogo de livre escolha
<b>07</b>	Aplicar conhecimentos construídos no bloco	Todos os conteúdos e conceitos apresentados	Realizar avaliação escrita

inicialmente branca para manipular imagens previamente construídas. Foi utilizada uma imagem de Steve Jobs para a manipulação – o laboratório não possui acesso à internet, o que inviabilizou o uso de imagens escolhidas pelos próprios estudantes. Os estudantes apresentaram mais dificuldades na criação de efeitos com o uso de fórmulas matemáticas como média – para o efeito de Escala de Cinza – e quanto à ideia de subtrair um valor de 255 (máximo valor do RGB) para gerar o efeito Negativo. Como a apresentação dos efeitos se dava através da teoria e os alunos deveriam implementá-la, em alguns momentos, os estudantes se reuniam para encontrar uma forma de fazer o efeito.

Percebe-se que essas atividades despertaram mais satisfação nos estudantes do que as atividades mais fáceis. O desafio é algo que impulsionou os estudantes a resolverem as atividades em sala. As atividades também despertaram um ambiente de colaboração. Os estudantes que terminavam as atividades ajudaram, quando solicitados, aos colegas. O quinto dia foi utilizado para a criação do efeito *chroma key*. Houve inicialmente uma explicação do que era o efeito, sem abordar em detalhes a técnica utilizada para a realização. Em seguida, foi solicitado que eles o implementassem. A maioria dos estudantes conseguiu terminar a atividade até o término da aula.

No segundo bloco contextualizado, continuou-se com a manipulação de imagens. No entanto, procurou-se aprofundar conceitos de POO. Assim, no primeiro dia, os estudantes foram apresentados à possibilidade de criação de classes. Já que criar classes relacionadas com imagens requer conhecimento de outros conceitos de POO, iniciou-se com um exemplo de calculadora como classe e as quatro operações aritméticas como métodos. Após esta atividade, foi apresentada a possibilidade de criação de classes para cada um dos efeitos já vistos.

Na segunda aula, as classes foram agrupadas em uma e os estudantes usaram condicionais e parâmetros para definir qual efeito seria aplicado. Os estudantes também não tiveram dificuldades na implementação destas atividades iniciais. No entanto, o texto *case-sensitive* em Python foi um empecilho, gerando erros pelo uso de letras maiúsculas no lugar de minúsculas e vice-versa.

Na terceira aula, os estudantes aprenderam sobre composição, modificando a classe que tinham implementado e utilizando um atributo do tipo *Picture* do JES. Assim, eles utilizaram o atributo tanto nas operações dentro da classe como fora dela, através do objeto. Na quarta aula, os estudantes aprenderam sobre herança e utilizaram este conceito em uma nova classe que herdou atributos e métodos de *Picture*. Esta classe, chamada *MyPicture*, possuía métodos para aplicação de efeitos. Assim, os estudantes implementaram as devidas modificações para o funcionamento da herança. Os estudantes tiveram dificuldades em usar o operador *self*, mas conseguiram realizar a atividade, com ajuda do professor e do monitor.

Na quinta aula, os estudantes continuaram utilizando herança e criaram uma classe *Meme*, onde deveria ser adicionado um texto a uma imagem no momento da instanciação. Assim, eles precisaram criar um construtor, chamar o construtor da classe pai, utilizar o operador *self*, parâmetros e operações matemáticas para o posicionamento do nome. Esta atividade foi a mais complexa e a que houve menos instrução, assim como a do *chroma key*. Durante as atividades, alguns estudantes exploraram a ferramenta, encontrando funcionalidades do JES, ou adicionaram comandos que já conheciam, como foi o caso de *random*. Apesar de o material do JES estar predominantemente em inglês, alguns estudantes descobriram tutoriais na internet e implementaram suas soluções a partir deles.

Apesar do engajamento dos estudantes nas aulas, eles não costumam estudar ou realizar atividades em casa. No entanto, as atividades se mostraram satisfatórias quanto à relevância e quanto a prender a atenção dos alunos. A resolução dos desafios, facilitada pela linguagem e pelo JES, proporcionou satisfação e confiança aos estudantes. A abordagem em espiral tem se apresentado como um elemento que impacta positivamente na aprendizagem dos estudantes, pois não os frustra, introduzindo os conteúdos gradualmente e em níveis crescentes de complexidade. Isto possibilita a aprendizagem dos conceitos e realização dos desafios e atividades.

## 5. Lições Aprendidas

Apresentamos aqui algumas lições aprendidas durante a execução da nossa abordagem.

**Python é adequado aos estudantes iniciantes.** Percebe-se que, embora haja problemas iniciais como indentação e texto *case-sensitive*, a linguagem oferece facilidade quanto aos conceitos de programação e orientação a objetos. Isto se deve à sintaxe facilmente compreensível e a possibilidade de ter resultados com poucas linhas de código, o que implica em menos chances de erro. Isto também motiva os estudantes iniciantes que, em suas primeiras linhas de código, já conseguem construir aplicações reais.

**Atividades contextualizadas** são importantes para despertar a atenção e relevância. Estudantes jovens e adolescentes utilizam aplicativos de redes sociais, aplicam filtros em fotos, escutam música, assistem vídeos. Assim, editar imagens é um contexto próximo dos estudantes, o que torna a atividade relevante aos seus interesses, tornando-se um fator a mais para a motivação.

**Introdução de novos conteúdos e ambientes** é importante e deve ser realizada de forma que os estudantes sintam-se gradualmente à vontade para exercitar sua criatividade. No entanto, percebe-se que as primeiras atividades que visam apresentar os conceitos do ambiente são menos ativas, sendo necessário, portanto, buscar formas mais ativas de apresentação destes conceitos.



**Uso de contextos similares** é uma característica relevante, pois minimiza o problema de introdução ao ambiente, fazendo possível focalizar em conceitos relacionados à programação, paradigmas, padrões, dentre outros. Com isto, reduz-se a sobrecarga cognitiva e possibilita-se uma aprendizagem mais eficaz, revisitando e aprofundando os conceitos. Este fator também proporciona uma postura mais ativa dos estudantes, tendo em vista que eles já conhecem os conceitos básicos referentes ao contexto.

**Os estudantes não estudam em casa.** Esta é uma característica desafiadora. Percebe-se que deve-se estimular o estudo fora da escola, oferecendo tutoriais, links para a instalação do ambiente em casa e materiais de consulta (e.g., slides das aulas). Solicitar atividades para serem entregues nas aulas seguintes é uma alternativa que leva os estudantes a estudar em casa. No entanto, em especial pelo perfil dos estudantes da educação pública, nem todos os estudantes têm computadores em casa.

**Desafios são positivos.** Para os estudantes, os desafios são relevantes pois os motivam. A ideia de conseguir algo que o professor julgou difícil ou que o próprio estudante, através de tentativas, percebeu que é difícil é algo relevante que afeta a confiança e satisfação dos estudantes. Isto pode ser despertado pela abordagem do professor, ao propor um desafio, ou pela própria atividade que, dada sua complexidade, gera um estado desafiador. Por isto, deve-se planejar atividades além do que foi ensinado, para que os estudantes fiquem motivados, exercitem sua criatividade e explorem as ferramentas.

## 6. Conclusões

Este trabalho apresentou um relato de experiência de uma abordagem de ensino-aprendizagem na disciplina Linguagem de Programação 2 com estudantes do terceiro ano de um curso Técnico em Informática. Nossa abordagem combina o uso de ambientes (JES e PPlay) e linguagem (Python) adequados aos iniciantes em programação, contextos relevantes a estes estudantes (Jogos e Imagens) e apresentação de conteúdos em nível gradual de complexidade, baseado no ensino em espiral.

Percebe-se, sobre a formulação dos blocos didáticos, que contextos que utilizam novas ferramentas tendem a ter um caráter inicial mais expositivo e menos propício ao exercício da criatividade pelos estudantes. Deste modo, o uso de contextos, linguagens ou ambientes em blocos seguidos, possibilita uma postura mais ativa dos estudantes perante as atividades.

O presente estudo sugere que Python é uma linguagem adequada aos estudantes, pois possibilita a realização das atividades, minimizando problemas sintáticos. Além disso, nota-se que atividades que utilizam contextos presentes na vida dos estudantes são elementos motivacionais que despertam a atenção e a percepção de relevância. A combinação do JES e Python gera confiança nos estudantes e as atividades desafiadoras, quando concluídas, aumentam o grau de satisfação.

Pretende-se, em trabalhos futuros, fazer uma análise pormenorizada da abordagem aqui apresentada, visando compreender mais aprofundadamente, através de dados qualitativos e quantitativos, como a configuração da abordagem impacta na motivação e no aprendizado de estudantes na modalidade da educação profissional.

## Referências

- Araujo, L. G. J., Bittencourt, R. A., and Santos, D. M. B. (2017). Ensino de Programação na Educação Básica através da Manipulação de Mídias. In *COBENGE 2017 – XLV Congresso Brasileiro de Educação em Engenharia*.
- Bordini, A., Avila, C. M. O., Weissshahn, Y., da Cunha, M. M., Cavalheiro, S. A. C., Foss, L., Aguiar, M. S., and Reiser, R. H. S. (2016). Computação na Educação Básica no Brasil: O Estado da Arte. *Revista de Informática Teórica e Aplicada*, 23(2):210–238.
- Bruner, J. S. (1966). *Toward a Theory of Instruction*, volume 59. Harvard University Press.
- Cravo, A. C. (2012). Análise das Causas da Evasão Escolar do Curso Técnico de Informática em uma Faculdade de Tecnologia de Florianópolis. *Revista Gestão Universitária na América Latina – GUAL*, 5(2):238–250.
- Grandell, L., Peltomäki, M., Back, R.-J., and Salakoski, T. (2006). Why Complicate Things?: Introducing Programming in High School Using Python. In *Proceedings of the 8th Australasian Conference on Computing Education*, pages 71–80.
- Guzdial, M. (2003). A Media Computation Course for Non-Majors. *ACM SIGCSE Bulletin*, 35(3):104–108.
- Guzdial, M. (2013). Exploring Hypotheses about Media Computation. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, pages 19–26. ACM.
- IC-UFF (2017). PPLay: Aprenda a Programar Desenvolvendo Jogos. <http://www2.ic.uff.br/pplay/>. Acesso em 18 ago. 2017.
- Machado, M. R., Moreira, R., and Priscila, R. (2006). Educação Profissional no Brasil, Evasão Escolar e Transição para o Mundo do Trabalho. In *I Colóquio Internacional sobre Educação Profissional e Evasão Escolar*.
- Marques, D. L., Costa, L. F. S., de Azevedo Silva, M. A., and Rebouças, A. D. D. S. (2011). Atraindo Alunos do Ensino Médio para a Computação: Uma Experiência Prática de Introdução à Programação Utilizando Jogos e Python. In *Anais do Workshop de Informática na Escola*, pages 1138–1147.
- Observatório Softex (2010). *Formação e Capacitação para a Indústria Brasileira de Software e Serviços de TI*. Softex.
- Prensky, M. (2001). Digital Natives, Digital Immigrants Part 1. *On the Horizon*, 9(5):1–6.
- Rebouças, A. D. D. S., Marques, D. L., Costa, L. F. S., and de Azevedo Silva, M. A. (2010). Aprendendo a Ensinar Programação Combinando Jogos e Python. In *Simpósio Brasileiro de Informática na Educação*.
- Santana, B. L., Figuerêdo, J. S. L., and Bittencourt, R. A. (2017). Motivação de Estudantes Non-Majors em uma Disciplina de Programação. In *WEI 2017 – XXV Workshop sobre Educação em Computação*.
- Simon, B., Kinnunen, P., Porter, L., and Zazkis, D. (2010). Experience Report: CS1 for Majors with Media Computation. In *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, pages 214–218. ACM.