

# Um Estudo Observacional sobre a Disciplina Introdutória de Programação

Elloá B. Guedes

<sup>1</sup>Escola Superior de Tecnologia  
Universidade do Estado do Amazonas  
Av. Darcy Vargas, 1200 – Manaus – Amazonas

elloa.uea@gmail.com

**Abstract.** *The particular characteristics of every context must be taken into account in programming teaching aiming to minimize problems such as the high rates of withdrawal which are common in these courses as described by the literature. Taking this into account, this work presents results from an observation study carried out in the introductory programming course in a public brazilian university. The results show that it is necessary to adopt other teaching strategies with students with previous failures in this courses, and that there are no evidences that the use of a programming language based on portuguese can favor the programming learning. Such study helps on identifying problems based on students' data as well as can lead to the development of strategies to overcome such problems.*

**Resumo.** *As características particulares de cada contexto devem ser levadas em consideração no ensino de programação com o intuito de minimizar os altos índices de abandono e reprovação nestas disciplinas, comumente registrados na literatura. Levando isto em consideração, este trabalho apresenta os resultados de um estudo observacional realizado na disciplina introdutória de programação em uma universidade pública brasileira. Os resultados mostram que é necessário adotar estratégias de ensino diferenciadas com estudantes que já sofreram reprovações prévias, e que não há evidências de que a utilização do português estruturado possa favorecer a aprendizagem de programação. Os resultados deste estudo auxiliam na identificação de problemas a partir de dados obtidos dos alunos e no desenvolvimento de estratégias para minimizá-los.*

## 1. Introdução

*Programação* é um assunto que se relaciona com muitas áreas da tecnologia e, em virtude de sua aplicação na resolução de diversos problemas, é estudada por alunos de diversas áreas. Recentemente a demanda por programadores cresceu e o número de estudantes de programação acompanhou esta tendência, fazendo com que os cursos introdutórios de programação se tornassem bastante populares [Robins et al. 2003].

Aprender a programar, entretanto, não é uma tarefa fácil. Programadores iniciantes sofrem uma grande variedade de dificuldades. Os cursos de programação são geralmente considerados difíceis e possuem altas taxas de reprovação e desistência. Isto acontece porque aprender a programar requer o desenvolvimento de conceitos abstratos, os grupos de estudantes costumam ser grandes e heterogêneos e também porque é difícil desenvolver estratégias gerais que capturem as necessidades de diversos perfis de alunos [Lahtinen et al. 2005].

A existência de dificuldades no ensino-aprendizagem de programação e a necessidade de contorná-las já são amplamente estudadas pela literatura [Soloway and Spohrer

2013]. Em diversos trabalhos há o relato de estratégias e práticas para favorecer o aprendizado de programação, bem como o estudo das vantagens e limitações destas práticas [Robins et al. 2003]. Embora estudos sobre estas práticas mostrem sua eficiência, nem todos os aspectos, dificuldades e particularidades de cada realidade são previamente conhecidos, permitindo a escolha de uma determinada prática em favor de outra. Assim, um passo que antecede a escolha de determinada estratégia é o conhecimento da realidade local, a fim de capturar suas particularidades.

Levando em consideração a necessidade de conhecer as particularidades de cada realidade local e de realizar diagnósticos, este trabalho ilustra os resultados de um estudo observacional conduzido na Escola Superior de Tecnologia da Universidade do Estado do Amazonas que teve dois objetivos principais: (i) avaliar o desempenho dos estudantes de Engenharias na disciplina de programação a fim de identificar possíveis explicações para os altos índices de reprovação e desistências reportados na referida instituição; e (ii) verificar se a utilização do português estruturado para aprendizagem de programação é um fator significativo para uma melhor aprendizagem, tal como argumentam alguns autores.

O estudo observacional em questão foi realizado em um semestre acadêmico com 147 estudantes divididos em quatro turmas, das quais uma era composta majoritariamente de alunos novatos e as outras três de alunos com reprovações prévias no currículo. Vários aspectos foram investigados, tais como, a origem destes estudantes (escola pública ou particular), a disponibilidade de computador e internet, os principais recursos utilizados para estudo, o número de horas semanais dedicados ao estudo da disciplina, dentre outros.

Os resultados deste estudo observacional mostram que alunos novatos, mesmo sem experiência prévia em programação, tendem a dedicar mais horas ao estudo da disciplina e, como consequência, tiveram maiores médias nas avaliações realizadas. Os alunos com reprovações prévias no currículo afirmaram estudar uma média de horas menor que a dos alunos novatos e afirmaram considerar a disciplina mais difícil que o outro grupo. Em relação a utilização do português estruturado, os dados mostram não haver evidências de que sua utilização facilite a aprendizagem de programação. Em relação ao desempenho dos alunos, os resultados obtidos mostram que é necessário utilizar outras práticas de ensino com os estudantes que sofreram reprovações prévias a fim de resgatar, principalmente, a motivação.

A realização e os resultados obtidos com este estudo observacional visam não somente fornecer informações para melhorar a disciplina na instituição em questão, mas também visam auxiliar e motivar outras instituições de ensino que estejam enfrentando problemas similares a analisar mais profundamente a realidade que possuem, minimizando problemas próprios e evitando conclusões baseadas apenas na intuição dos docentes.

Para apresentar os procedimentos realizados e os resultados da análise observacional em questão, este trabalho está organizado como segue. Na Seção 2 é ilustrada a caracterização do contexto em que o estudo observacional foi realizado, as características da disciplina de Programação e dos estudantes, bem como do português estruturado. Em seguida, na Seção 3, é apresentado o estudo observacional: seus objetivos, os procedimentos para coleta de dados, os resultados obtidos, dentre outros. Por fim, uma reflexão sobre os resultados obtidos e sugestões de trabalhos futuros são apresentados na Seção 4.

## 2. Contexto de Observação

A disciplina *Linguagem de Programação I*, ou Introdução à Programação, ministrada na Escola Superior de Tecnologia na Universidade do Estado do Amazonas é uma disciplina básica dos cursos de Engenharia, sendo ministrada no primeiro semestre letivo para todos

os alunos novatos. Nesta etapa, os alunos ainda não fizeram a escolha da graduação que irão cursar (Engenharia Civil, Química, Elétrica, etc.), o que será realizado apenas após o primeiro ano. Neste interim os alunos cursam disciplinas básicas da formação de Engenharia comum a todos os cursos, o chamado Ciclo Básico, que inclui disciplinas de Física, Cálculo, Álgebra, entre outras.

No primeiro semestre de cada ano são oferecidas, em média, 8 turmas desta disciplina para alunos novatos e 2 turmas para alunos com reprovações prévias. No segundo semestre, são oferecidas 4 turmas apenas para alunos com reprovações prévias, pois a entrada de alunos novatos é feita apenas no começo de cada ano. Em média, em cada turma são matriculados até 40 alunos. O objetivo da disciplina, conforme consta em sua ementa, é “*desenvolver a habilidade de construir algoritmos básicos, com a utilização de conceitos de programação estruturada, para a solução de problemas por meio de uma linguagem de programação*”.

Infelizmente, por questões de infraestrutura, não há laboratórios suficientes para ministrar tais aulas com computador. Assim, as aulas são majoritariamente expositivas, ministradas em salas de aula convencionais com a utilização de pincel e quadro e *data-show* para exibição de slides. Há resolução de problemas, cujos algoritmos são construídos no quadro pelo professor ou no papel pelos alunos. Alguns algoritmos são escolhidos para serem executados no computador do professor, para que os alunos possam visualizar a execução do mesmo no computador. Os professores desta disciplina possuem formação básica em Computação e possuem Mestrado na área. A carga-horária da disciplina é de 60 horas, com 4 horas-aula semanais. Há laboratórios com monitores disponíveis para auxílio dos alunos extra-classe e os professores disponibilizam pelo menos 4 horas semanais para atendimento aos estudantes além do horário das aulas.

Considerando a vantagem para o aprendizado de programação argumentada por alguns trabalhos na literatura [Dorneles et al. 2010], a linguagem adotada na disciplina é o português estruturado. O *Português Estruturado*, também conhecido como *Portugol*, é uma pseudolinguagem de programação utilizada no ensino de lógica e programação para iniciantes. Embora esteja presente em muitos materiais didáticos, não há uma documentação formal de sua sintaxe. Entretanto, para permitir a execução de programas construídos nesta linguagem, alguns autores propuseram a especificação da linguagem e construíram compiladores e interpretadores para as mesmas [Dorneles et al. 2010, de Souza 2009, Evaristo and Crespo 2010].

A maioria dos interpretadores de português estruturado contém apenas os elementos básicos de uma linguagem de programação que segue o paradigma imperativo, tais como: sistema de tipos, comandos de atribuição, desvio condicional e comandos de repetição. Algumas versões de interpretadores para português estruturado proveem recursos mais avançados, tais como a construção de variáveis compostas (registros), o uso de funções, dentre outros. A Tabela 1 sintetiza alguns recursos disponíveis no uso do português estruturado tomando como base a sintaxe do Visualizador de Algoritmos (VisuALG) [de Souza 2009], utilizada na disciplina.

**Tabela 1: Recursos disponíveis no VisuALG.**

Recursos	Exemplo
<b>Tipos de Dados</b>	Inteiro, Real, Caractere e Lógico
<b>Comando de Atribuição</b>	<variável>← <valor>
<b>Desvio Condicional</b>	se <valor-lógico>então <...> fimse
<b>Comandos de repetição</b>	para <variável> de <valor-inicial> ate <valor-final> faca <...> fimpara repita <...> ate (<valor-lógico>) enquanto (<valor-lógico>) faca <...> fimenquanto

Na maioria dos compiladores e interpretadores de português estruturado a grafia dos comandos e palavras reservadas não é *case-sensitive*, isto é, podem ser escritos em maiúscula ou minúscula sem alteração no resultado final. Em todos os casos observados, a definição do português estruturado é fortemente tipada, não requer indentação e a declaração de variáveis é feita de maneira estática, isto é, no momento da compilação.

No que diz respeito a avaliação da disciplina, são realizadas duas provas semestrais. As provas são realizadas em 100 minutos e são iguais para todas as turmas, sendo realizadas no mesmo dia e horário e elaboradas pela comissão de professores que está ministrando a disciplina naquele período. Caso o aluno alcance a média 8,0, está aprovado. Porém, em caso contrário, está apto a realizar a avaliação final. Nesta situação, deve alcançar média igual ou superior a 6,0 para lograr a aprovação. A presença na disciplina é obrigatória e deve ser de pelo menos 75%.

É interessante notar que a cada ano, 8 turmas de programação são oferecidas para alunos novatos e 6 turmas são oferecidas para alunos que já passaram por reprovações prévias, o que indica que há uma re-oferta anual de 75% desta matéria, um número considero alto, acima da média documentada na literatura [Bennedsen and Caspersen 2007]. Embora, a instituição oferte a disciplina em número alto de turmas, permitindo assim que alunos com reprovações prévias tenham nova chance de se recuperar e cumprir os créditos da disciplina, ano após ano há um alto índice de reprovações, reportados ao final de cada semestre pelos professores da disciplina, resultados demonstrados pelos diários de classe e pelas notas cadastradas no sistema acadêmico. Esse cenário motivou a realização de um estudo observacional, descrito na próxima seção, foi realizado com vistas a entender as causas deste problema para então sugerir práticas que colaborem com o aprendizado dos estudantes neste cenário.

### 3. Estudo Observacional

Na tentativa de investigar a utilização do português estruturado no contexto do seu uso, foi realizado um estudo observacional em quatro turmas de Introdução à Programação no primeiro semestre de 2014. O estudo observacional seguiu as recomendações dispostas em Wohlin et al. [Wohlin et al. 2012, Cap. 5] e os dados foram coletados dos alunos por meio de questionários e dos resultados das avaliações realizadas.

Considerando os objetivos deste estudo observacional mencionados na Seção 1, uma maneira de auxiliar na visualização de tais informações foi por meio da análise dos dados de forma comparativa, considerando alunos novatos e alunos com reprovações prévias.

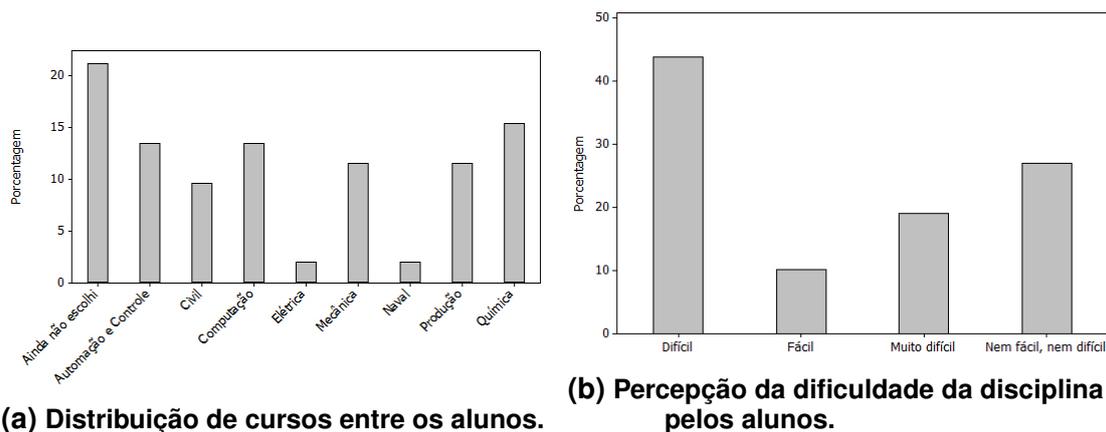


Figura 1: Gráficos contendo informações coletadas dos questionários dos alunos no início da disciplina.

### 3.1. Perfil dos Alunos

Os alunos responderam às perguntas por meio de questionários disponibilizados online. Nesses questionários não era possível identificar-se, deixando os alunos mais à vontade para emitirem as suas opiniões. A participação nos questionários também foi feita de forma voluntária. No questionário de sondagem, realizado no início da disciplina, 30% dos alunos participaram. Nos demais, realizados após as avaliações parciais, 27% e 60% dos alunos participaram, respectivamente. Considerando o número de respostas no questionário em que houve menor participação, é possível afirmar que os resultados obtidos são representativos para um intervalo de confiança de 95% com margem de erro de 13% [Jain 1991].

Do total de alunos que responderam ao questionário, tem-se que 63% são do sexo masculino, em contraste com 37% do sexo feminino. Os alunos oriundos majoritariamente de Escolas Públicas no Ensino Médio totalizaram 73,33%, em contraste com 26,67% de Escolas Particulares. Segundo o questionário, 71,11% afirmam não terem vínculos empregatícios, tendo seu tempo majoritariamente dedicado apenas para os estudos. A presença de computador em casa já é uma realidade para 93,3% dos alunos, bem como o acesso a internet, que já está disponível para 86,67% deles. Ou seja, mesmo havendo uma limitação de infraestrutura na universidade, os alunos dispunham do recurso para estudo em casa.

Os alunos distribuem-se por todos os cursos de Engenharia disponíveis na instituição, concentrando-se principalmente em Engenharia da Computação, Automação e Controle e Mecânica. Uma parcela significativa dos estudantes afirmou ainda não ter escolhido o curso, pois estes alunos integram o Ciclo Básico. A distribuição dos alunos por curso é ilustrada na Figura 1a. Apesar das diferentes escolhas de curso, 86% dos alunos consideram programação *totalmente relevante* ou *relevante* para a sua formação.

Em relação à percepção da dificuldade da disciplina, a maioria dos alunos afirmaram considerar esta difícil, conforme ilustrado na Figura 1b, resultado que está de acordo com outros relatos da literatura [Bennedsen and Caspersen 2007, Lahtinen et al. 2005]. Dos alunos que responderam ao questionário, 38% afirmaram não terem cursado a disciplina antes, ou seja, são novatos. Uma grande parte do total de alunos não possuía experiência prévia em programação (82%), os demais que possuem experiência prévia afirmaram serem oriundos de cursos técnicos.

Fazendo um contraste entre alunos com reprovações e alunos novatos, alguns resul-

**Tabela 2: Tabulação Quadrada: Possui reprovação versus Maneira de Estudar, por quantidade de alunos.**

Possui reprovação?	Apostilas	Exercícios Resolvidos	Listas de Exercícios	Livros	Slides	Vídeo Aulas
Não	13%	18%	32%	14%	9%	14%
Sim	0	56%	26%	8%	5%	5%

tados interessantes emergem. Em relação a maneira de estudar, os alunos que já sofreram reprovações preferem exercícios resolvidos. A relação entre ter cursado a disciplina anteriormente e o material de estudo preferido está explicitada na Tabela 2.

A maneira de estudar majoritariamente por exercícios resolvidos pode dar indícios de que não há muito prática, o aluno pode acabar apenas revisando as respostas, sem desenvolver uma atitude ativa de recriar soluções, estudar diferentes maneiras de solucionar o problema. Por consequência, isso compromete o desenvolvimento do raciocínio lógico, uma das metas a serem desenvolvidas na disciplina de programação. Este é, portanto, um dado relevante que fornece indícios para explicar a falta de sucesso de tais alunos em momentos anteriores. Com base nesta informação, todos os alunos foram alertados ao longo da disciplina para que procurassem estudar por outras fontes, principalmente pela resolução de exercícios.

Uma outra informação capturada nesta etapa foi o número de horas semanais que os alunos dedicam-se ao estudo da disciplina. Tanto os alunos novatos quanto os que possuem reprovações prévias estudam, em média, 5 horas semanais além do horário das aulas.

### 3.2. Desempenho dos Alunos nas Avaliações

As duas avaliações parciais foram realizadas após a 16ª e a 28ª aulas, respectivamente, de um total de 30 aulas da disciplina. Cada avaliação era composta de duas questões nas quais eram solicitadas a construção de algoritmos. Embora a solução fosse um algoritmo, a correção levava em consideração partes do mesmo. Essa estratégia foi utilizada para evitar correções estritamente rigorosas, resultados em resultados binários (questões totalmente certas ou totalmente erradas).

Após cada avaliação foi realizado um questionário. Em relação à primeira avaliação, os alunos foram questionados sobre o grau de dificuldade da avaliação e sobre a proximidade da mesma com as aulas e exercícios. Conforme mostram os gráficos das Figuras 2a e 2b, embora os alunos tenham relatado uma alta proximidade entre a avaliação e as aulas, o número de alunos que considerou a prova difícil ou muito difícil foi bastante significativo.

Perguntados sobre o número de horas que utilizaram além da sala de aula para estudarem a matéria, foi possível perceber que, quanto maior o número de reprovações prévias, menor também foi o número de horas estudadas. A partir do gráfico da Figura 2c, é possível perceber que o número de horas estudadas é inversamente proporcional ao número de vezes que o aluno cursou a disciplina anteriormente.

O desempenho dos alunos por turma é mostrado por meio de intervalos de confiança de 95% para a média no gráfico da Figura 2d. As turmas 1, 2 e 3 eram compostas majoritariamente por alunos que já haviam sofrido reprovações anteriores, enquanto a turma 4 consistia principalmente de alunos novatos. É interessante notar que a maior média reside nos alunos novatos. O desvio padrão é próximo para todos os intervalos, mostrando um desempenho homogêneo em cada turma.

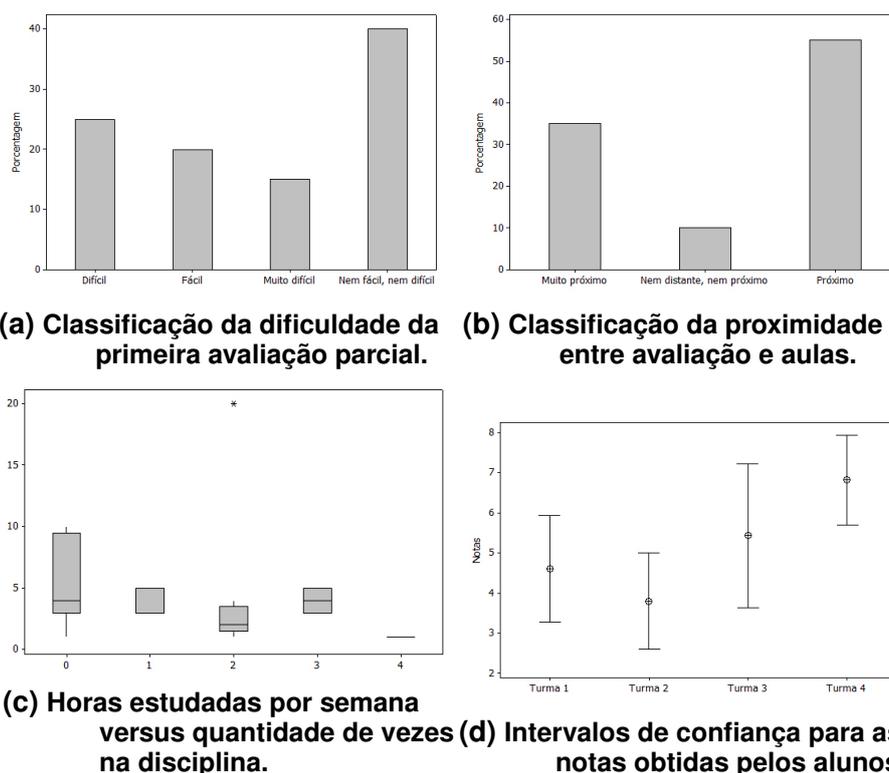


Figura 2: Gráficos contendo informações coletadas dos questionários dos alunos após a realização da primeira avaliação.

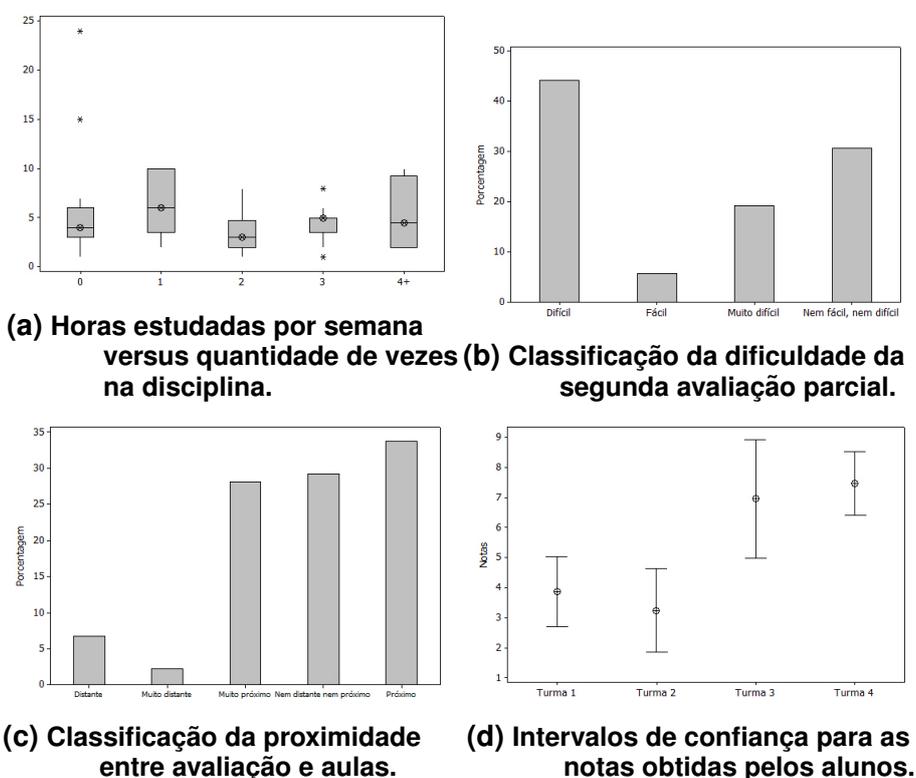


Figura 3: Gráficos contendo informações coletadas dos questionários dos alunos após a realização da segunda avaliação.

Em se tratando da segunda avaliação, o número de horas de estudo por semana para a segunda avaliação se manteve mais equilibrado entre as turmas, independentemente do seu histórico, como pode ser observado na Figura 3a. Porém, a percepção de dificuldade sobre a avaliação aumentou em relação à primeira avaliação, como mostrado na Figura 3b. Isto acontece porque os conteúdos didáticos são cumulativos da primeira para a segunda avaliação. Na média, a percepção de proximidade entre aulas e avaliação sofreu uma ligeira redução, mas a proximidade ainda é predominante, como mostra a Figura 3c. Os intervalos de confiança para as notas obtidas nesta avaliação são ilustradas na Figura 3d, na qual percebe-se que houve um aumento na média de todas as turmas em relação à primeira avaliação parcial.

O questionário realizado após a segunda avaliação também sondou os alunos sobre as impressões que tinham sobre seu próprio desempenho. A resposta do tipo “Não obteve sucesso e não alcançará pontuação para ir para a final” foi a menos frequente de todas (10%), seguido pelos demais tipos que obtiveram mais adeptos (“Não obteve sucesso, mas não irá para a final” – 15%; “Obteve sucesso e será aprovado por média” – 31%; e “Obteve sucesso, mas precisará ir para a final” – 44%). Aos alunos novatos foi perguntado se a disciplina os fez considerar o curso de Engenharia da Computação como uma opção de carreira e 62% afirmaram que sim, que a disciplina abriu-lhes essa percepção.

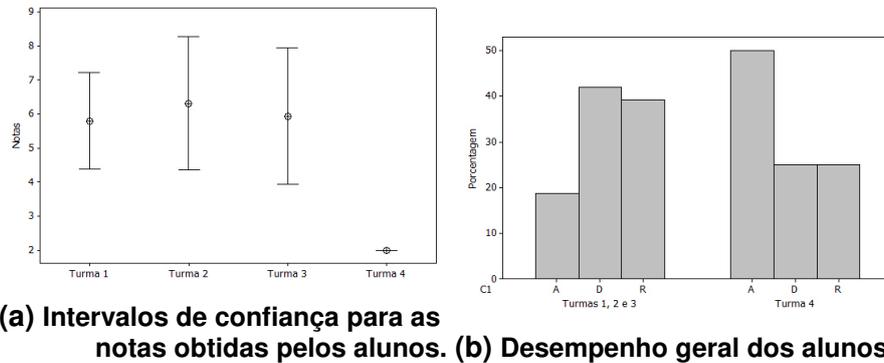
Em uma aula ao longo do curso foi mostrado aos alunos como eles poderiam construir aplicações multimídia, tais como animações, jogos, dentre outros, utilizando o Scratch. O Scratch utiliza uma linguagem de programação essencialmente visual, com comandos similares ao do português estruturado [Ford 2008]. O objetivo desta aula era motivar os alunos a utilizarem os conceitos de programação aprendidos na disciplina. No questionário realizado após a segunda avaliação foi também perguntando qual a motivação dos alunos para posteriormente usarem Scratch. Embora o Scratch possibilite utilizar os conceitos aprendidos na disciplina para criação de aplicações com outros propósitos (trabalhos para outras disciplinas, jogos para compartilhar na web, etc.), metade afirmou não se sentir motivado.

Apesar da utilização do Scratch não ter significativamente motivado os alunos, quando perguntados sobre a utilização de uma linguagem de programação real na disciplina (tais como C, Python, etc.), 68% afirmaram que gostariam que tais linguagens fossem utilizadas em substituição ao português estruturado. Considerando apenas os alunos novatos, o percentual de aceitação foi de 85%. Este é um bom indicador de que o português estruturado não foi tão significativo no aprendizado de programação, pois os alunos foram capazes de compreender a lógica e preferem uma linguagem real, passível de aplicação em contextos mais amplos. Os alunos novatos, cuja expressividade na utilização de uma linguagem real foi mais significativa (85%), também foram aqueles que tiveram o melhor desempenho na disciplina (Turma 4).

Em relação a utilização de português estruturado e outras linguagens de programação, o trabalho de Souza et al. [de Souza et al. 2013] relata uma comparação entre a linguagem de programação Pascal e o português estruturado para construção de algoritmos por iniciantes. Neste estudo os alunos foram motivados a construir algoritmos nessas duas linguagens de programação. Ao final, os estudantes foram perguntados sobre o idioma utilizado, se este dificultou ou facilitou a aprendizagem. Embora os estudantes tenham argumentado que houve uma maior dificuldade na utilização do Pascal em virtude do fato dos comandos serem escritos na língua inglesa, eles afirmaram preferir esta linguagem para construção de algoritmos. Os dados obtidos neste estudo observacional mostram um alinhamento com estes resultados.

No término da disciplina foi realizada a avaliação final para aqueles que não ti-

tenham sido aprovados e que alcançaram média parcial igual ou superior a 4,0. As notas obtidas pelos estudantes de acordo com a turma estão dispostas na Figura 4a. O gráfico do desempenho geral, considerando alunos novatos e que já tinham cursado a disciplina anteriormente, é ilustrado na Figura 4b.<sup>1</sup> Nesta etapa, é possível observar um aumento representativo na média das notas das Turmas 1, 2 e 3.



**Figura 4: Gráficos contendo informações dos alunos após a realização da avaliação final.**

Considerando o desempenho final das Turmas 1, 2 e 3, majoritariamente composta por alunos que já tinham cursado a disciplina anteriormente e não obtiveram sucesso, é interessante notar o alto índice de desistência e reprovação existente neste grupo. Durante o questionário também foi identificada uma menor motivação destes alunos, que alimenta um círculo vicioso. O oferecimento de vagas para cursar a disciplina não pode ser considerado um fator determinante para oportunizar a aprovação do aluno. Para turmas com perfil similar a este é essencial repensar as estratégias de ensino, de acompanhamento e de avaliação destes alunos. Embora no início da disciplina estes afirmaram estudarem uma média de horas igual ao dos alunos novatos, não há evidências de que este estudo esteja sendo efetivo para o aprendizado. Há indícios de que os alunos com este perfil não conseguem superar dificuldades antigas e se engajarem no aprendizado de forma significativa.

Com os recursos disponíveis atualmente, uma maneira de engajar alunos com o perfil mencionado em atividades que colaborem e favoreçam o aprendizado de programação é por meio da utilização dos laboratórios disponíveis no horário além da sala de aula com o acompanhamento de monitores para retirada de dúvidas; a adoção de estratégias para avaliação contínua; e a adoção de estratégias de ensino diferenciadas, tal como o currículo invertido [Bergmann and Sams 2012], visando um maior engajamento dos alunos. O engajamento dos professores na coleta de dados, identificação de problemas e implantação de estratégias que os minimizem também deve ser uma constante.

#### 4. Considerações Finais

Neste trabalho foram descritos os resultados de um estudo observacional realizado em uma universidade pública do norte do Brasil, tendo por objetivo avaliar o desempenho dos estudantes de Engenharias na disciplina de programação a fim de identificar possíveis explicações para os altos índices de reprovação e desistências reportados na referida instituição; e verificar se a utilização do português estruturado é um fator significativo para uma melhor aprendizagem de programação, tal como argumentam alguns autores.

Este estudo observacional foi realizado em um semestre acadêmico e o principal meio de coleta de dados foi a realização de questionários com os alunos. A partir dos dados, foi possível identificar que alunos que já tinham cursado a disciplina em um momento

<sup>1</sup> A letra “A” denota aprovado, “D” denota desistente e “R” denota reprovado.

anterior estudam praticamente a mesma média de horas que um aluno novato, porém utilizam estratégias que tendem a tornar este estudo menos eficaz no que diz respeito ao desenvolvimento das habilidades necessárias para a disciplina. Este estudo observacional aponta também para a necessidade de desenvolver estratégias diferenciadas de ensino, acompanhamento e avaliação dos estudantes que já sofreram reprovações com vistas a resgatar a motivação destes alunos e a engajá-los no aprendizado de programação. Além disso, não foi possível confirmar evidências de que a utilização do português estruturado (pseudolinguagem) possa melhorar a aprendizagem de programação.

Os resultados deste estudo observacional vão servir de diretrizes para implementar melhorias contínuas na disciplina a fim de minimizar os problemas diagnosticados. A utilização de novas estratégias de ensino e a escolha de Python como linguagem de programação serão os primeiros passos adotados na implantação de tais melhorias.

### **Agradecimentos**

A autora deste trabalho agradece aos professores e funcionários da Escola Superior de Tecnologia da Universidade do Estado do Amazonas. Este trabalho foi desenvolvido com o apoio da Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM).

### **Referências**

- Bennedsen, J. and Caspersen, M. E. (2007). Failure rates in introductory programming. *SIGCSE Bull.*, 39(2):32–36.
- Bergmann, J. and Sams, A. (2012). *Flip your classroom: Reach every student in every class every day*. International Society For Technology In Education.
- de Souza, C. M. (2009). VisuALG – Ferramenta de Apoio ao Ensino de Programação. *Revista TECCEN*, 2(2):1–9.
- de Souza, M. B., Moreira, J. L. G., Lobo, F. L., and dos Santos Alencar, M. A. (2013). Uma abordagem metodológica voltada para o ensino-aprendizagem de algoritmos. *Revista Novas Tecnologias na Educação*, 11(1):1–10.
- Dorneles, R. V., Picinin Jr., D., and Adami, A. G. (2010). Algoweb: A web-based environment for learning introductory programming. In *IEEE International Conference on Advanced Learning Technologies*.
- Evaristo, J. and Crespo, S. (2010). *Aprendendo a Programar – Programando Numa Linguagem Algorítmica Executável*. 2 edition.
- Ford, J. L. (2008). *Scratch Programming for Teens*, volume 1. Editora CENGAGE Learning.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley.
- Lahtinen, E., Ala-Mutka, K., and Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 14–18.
- Robins, A., Rountree, J., and Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2):137–172.
- Soloway, E. and Spohrer, J. C. (2013). *Studying the Novice Programmer*. Lawrence Erlbaum Associates.
- Wohlin, C., Runeson, P., Horst, M., Ohlsson, M. C., Regnell, B., and Wesslen, A. (2012). *Experimentation in Software Engineering*. Springer.