

Contribuições da Computação para as Tecnologias Educacionais

Gabriel Gerber Hornink (Org.)

1ª Edição

UNIFAL-MG

2018

when clicked

with 1 exec

repeat 10

ask: What's your name? end with

**Contribuições da
Computação para as
Tecnologias Educacionais**

Organizador: Gabriel Gerber Hornink

Autores: Augusto Márcio da Silva Júnior, Daniele de Almeida Dias, Gabriel Gerber Hornink, Lucas Faloni Ferreira, Luis Eduardo de Matos, Paulo Alexandre Bressan, Raphael Biavati Silva, Raquel Ribeiro Balbino

Editoração: Gabriel Gerber Hornink

Revisão: Erica Nicacio Hornink

Apoio à editoração: Marlom César da Silva

Capa e contra-capas: Eduardo de Almeida Rodrigues

Contribuições da Computação para as Tecnologias Educativas

1ª Edição

Alfenas-MG

UNIFAL-MG

2018

© 2018 Direitos reservados aos autores. Direito de reprodução do livro é de acordo com a lei de Lei nº 9.610, de 19 de fevereiro de 1998. Qualquer parte desta publicação pode ser reproduzida, desde que citada a fonte.

Contribuições da Computação para as tecnologias educacionais

Disponível em: <<http://www.unifal-mg.edu.br/bibliotecas/ebooks>>



Universidade Federal de Alfenas – UNIFAL-MG

Endereço: Rua Gabriel Monteiro da Silva, 700 Centro – Alfenas – Minas Gerais – Brasil – CEP: 37.130-001

Reitor: Sandro Amadeu Cerveira / Vice-reitor: Alessandro Antonio Costa Pereira

Sistema de Bibliotecas da UNIFAL-MG / SIBI/UNIFAL-MG

Organizador: Gabriel Gerber Hornink

Autoria de capítulos: Augusto Márcio da Silva Júnior, Daniele de Almeida Dias, Gabriel Gerber Hornink, Lucas Faloni Ferreira, Luis Eduardo de Matos, Paulo Alexandre Bressan, Raphael Biavati Silva, Raquel Ribeiro Balbino

Editoração: Gabriel Gerber Hornink

Revisão: Erica Nicacio Hornink

Apoio à editoração: Marlom César da Silva

Capa e contracapa: Eduardo de Almeida Rodrigues

Algumas das pesquisas deste *eBook* foram financiadas pelo Conselho Nacional de Desenvolvimento Científico e tecnológico (CNPq), pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG), pela Unifal-MG, por meio de Bolsas e equipamentos.

Dados Internacionais de Catalogação-na-Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal de Alfenas
Biblioteca Central – Campus Sede

C569

Contribuições da Computação para as tecnologias educacionais / Organizador: Gabriel Gerber Hornink -- Alfenas -- MG: Editora Universidade Federal de Alfenas 332f.:il --

ISBN: 978-85-63473-33-2 (E-book)

Disponível em: <http://www.unifal-mg.edu.br/bibliotecas/ebooks>

Inclui Bibliografia

Vários autores

1. Tecnologia Educacional. 2. Softwares. 3. Ciências - Estudo e ensino. 4. Mídia digital. 5. Dispositivos móveis. I. Hornink, Gabriel Gerber (org.). II. Título.

CDD:371.33
CDU: 371.334

Ficha Catalográfica elaborada por Marlom Cesar da Silva
Bibliotecário-Documentalista CRB6/2735

Sumário

Apresentação.....	7
--------------------------	----------

Capítulo 1 Histórico da informática e as implicações no ensino.....	15
--	-----------

1 Introdução.....	16
2 Histórico de utilização da informática na educação.....	20
3 Avanços tecnológicos e as novas metodologias de aprendizagem.....	27
4 Inovações na programação e as repercussões no processo educacional.....	43
5 Ambientes Virtuais de Aprendizagem (AVA).....	59
6 Considerações finais.....	73
Referências.....	75

Capítulo 2 DigiAtlas - Construção e avaliação de aplicativos para auxílio em trabalhos de campo.....	87
---	-----------

1 Introdução.....	88
2 Dispositivos móveis no ensino.....	93
3 Desenvolvimento - elementos principais.....	98
4 Resultados e discussões.....	128
5 Considerações finais.....	156
Referências.....	158

Capítulo 3 O uso do Scratch como proposta multidisciplinar educacional.....	163
--	------------

1 Introdução.....	164
1.1 O contexto educacional.....	166
1.2 Conhecendo o Scratch.....	173
1.3 Experiências no ensino.....	178
2 Procedimentos.....	180
3 Resultados.....	189

4 Considerações finais.....	210
Referências.....	212

Capítulo 4. InMapMoodle: ferramenta para moderação de fóruns.....215

1 Introdução.....	216
2 Ambientes virtuais de aprendizagem.....	218
3 Comparativo entre as plataformas.....	223
4 Fóruns.....	226
5 Grafos e suas aplicações computacionais.....	228
5 A Ferramenta inMapMoodle.....	232
6 Avaliação da ferramenta pelos tutores.....	243
7 Considerações finais.....	250
Referências.....	251

Capítulo 5 O uso de sistemas lúdicos de autoria no ensino-aprendizagem de programação.....254

.....	254
1 Introdução.....	255
2 Reflexões teóricas.....	260
3 Materiais e métodos.....	268
4 Resultados e Discussões.....	289
5 Considerações finais.....	303
Referências.....	305

Apêndices.....311

Apêndice 3.A Proposta de projeto final do curso de Scratch.	311
Apêndice 3.B Questionário Inicial (QI) do curso.....	312
Apêndice 3.C Questionário final (QF) do curso.....	314
Apêndice 5. A Dados pré e pós curso - 1ª oferta.....	317
Apêndice 5.B Levantamento prévio.....	319
Apêndice 5.C Respostas dos questionários pré e pós curso na 1ª oferta.....	323

Apêndice 5.D Resultados estatísticos - 1º oferta.....324

Apêndice 5.E Respostas dos questionários pré e pós curso
na 2ª oferta.....325

Apêndice 5.F Resultados estatísticos referentes a 2º oferta.
.....328

Autores.....329

Contato.....332

Apresentação

Gabriel Gerber Hornink

O desenvolvimento humano é algo que está em constante movimento, em todos aspectos possíveis, como os sociais, culturais e tecnológicos, sendo que há uma relação intrínseca e cíclica entre estes.

Não há como conceber o avanço tecnológico sem contemplar os demais aspectos, uma vez que para inovar nas tecnologias digitais se faz necessário inovar nas ideias, mas, para tanto, tem-se também influência das tecnologias digitais desenvolvidas e concepções culturais situadas historicamente, localmente, regionalmente e globalmente.

Iniciei meus trabalhos de pesquisa, desde a graduação, até a pós-graduação, focando nas tecnologias digitais educacionais para o ensino, inicialmente de Biologia e Ciências, e depois expandido para outras áreas.

Nesse percurso, percebi um pouco do avanço tecnológico que permitiu criarmos novos e melhores aplicativos para o ensino-aprendizagem, partindo de materiais simples e estáticos, para materiais dinâmicos,

interativos, gerando simulações de processos, chegando aos sistemas de autoria, nos quais os próprios estudantes passam a construir conhecimentos durante a ação de construir aplicativos para computadores ou dispositivos móveis.

Trouxe um pouco desta experiência para os projetos que participo como docente na Universidade Federal de Alfenas (Unifal-MG), onde estou a quase dez anos, desenvolvendo trabalhos na área de tecnologias educacionais.

No período de 2009 a 2018 pude trabalhar com diversos professores e estudantes da área de Ciência da Computação da Unifal-MG, desenvolvendo, principalmente, iniciações científicas, trabalhos de conclusão de curso e projetos de extensão, a partir dos quais desenvolvemos diversos softwares educacionais, além de pesquisas sobre as tecnologias aplicadas ao ensino.

A ideia deste *eBook* parte de compartilhar um pouco desta experiência, a partir das quais podemos compreender um pouco de como a área das Ciências da Computação pode contribuir para a melhoria da Educação e de como se faz importante a existência de projetos multi-interdisciplinares que conectem o ensino,

a pesquisa e a extensão, potencializando o uso desses instrumentos culturais digitais nos processos de ensino-aprendizagem.

Os capítulos que compõem este eBook são adaptações, com algumas atualizações, dos Trabalhos de Conclusão de Curso de cinco estudantes de Ciências da Computação que trabalharam em projetos resultantes da parceria do Laboratório de Mídias Educacionais (Depto. Bioquímica/ Instituto de Ciências Biomédicas/ Unifal-MG) com o Laboratório de Tecnologias Educacionais (Depto. Ciências da Computação / Instituto de Ciências Exatas).

No primeiro capítulo, a Raquel Ribeiro Balbino trabalhou, em 2011, um pouco do histórico da informática e as implicações dos avanços nas tecnologias digitais para a educação e para o ensino a distância. Abordou-se neste capítulo um pouco sobre o desenvolvimento das linguagens de programação e o que essas possibilitaram desenvolver que impactou em melhorias aplicações para uso no ensino.

O capítulo dois é resultado do trabalho de conclusão de curso do Lucas Faloni Ferreira, em 2015, com minha orientação e coorientação do prof. Paulo Alexandre Bressan, que se conectou com um projeto da Fapemig de desenvolvimento de um aplicativo para

dispositivos móveis (plataforma Android) para trabalhos de campo com dados ambientais do sul de Minas Gerais.

A partir deste capítulo vocês poderão compreender os fundamentos educacionais e computacionais que utilizamos na construção e avaliação do DigiAtlas, trazendo contribuições para outros pesquisadores desenvolverem suas aplicações para Android.

Destaco que este projeto teve continuidade com outro aluno da Ciência da Computação, como TCC, além de projeto de Extensão.

Apresentamos no Capítulo 3 uma de nossas primeiras experiências formalizadas abordando o uso do Scratch do ensino, a partir do projeto de TCC da Daniele de Almeida Dias, iniciado em 2014 e defendido em 2015. Iniciamos o uso do Scratch em 2012 no Laboratório de Mídias Educacionais e este trabalho da Daniele foi um dos motivadores para a criação do projeto de Extensão Pensando em Códigos o qual foi posto em prática posteriormente no contexto do mestrado do Augusto e TCC do Luis (Capítulo 5).

Destacamos que, além desses trabalhos, tivemos alunos de iniciação científica Júnior (BIC-JR) trabalhando com o uso do Scratch e App inventor (2012, 2013, 2014 e 2015).

A experiência prévia com esses alunos de BIC-Jr, entre 2012 e 2014, foram importantes para a estruturação do trabalho da Daniele e que se consolidaram com a criação do projeto de ensino, pesquisa e extensão “Pensando em Códigos” no fim de 2015, quando enviamos este para um edital da pró-reitoria de Extensão.

Destaca-se que no contexto do projeto Pensando em Códigos outros projetos foram e estão em desenvolvimento, incluindo o trabalho de TCC apresentado no Capítulo 5, além de um mestrado defendido e dois em andamento em Educação com previsão para término entre dezembro de 2018 e fevereiro de 2019.

No Capítulo 4 apresentamos o desenvolvimento do TCC abordando a construção do bloco inMapMoodle para o Moodle pelo estudante Raphael Biavati. Destaco que o texto em si foi foco do trabalho do Raphael, mas que o aplicativo foi desenvolvido por uma equipe, inicialmente instaurada no Centro de Educação Aberta e a Distância (CEAD da Unifal-MG).

O inMapMoodle é um bloco que possibilita a geração de grafos direcionais com os dados dos fóruns de discussão do Moodle, ou seja, ele cria mapa das

interações, quem escreveu para quem, usando setas direcionais, cores para indicar o número de mensagens e bordas para o tipo de usuários.

O aplicativo teve seu desenvolvimento iniciado em 2011 com o grupo de estagiários do CEAD, quando sentimos uma demanda de compreender melhor a dinâmica dos fóruns nos cursos de graduação e pós-graduação da Unifal-MG, no período em questão. A ideia era construir uma ferramenta que trouxesse indícios da dinâmica dos diálogos nos fóruns para que os tutores pudessem usar essas na moderação dos fóruns.

O software foi concluído em sua primeira versão em 2012 e teve ainda uma segunda versão em 2016, sendo que, no contexto do inMapMoodle, tivemos mais dois alunos de iniciação científica, um de TCC e um de mestrado em Educação (Eduardo de Almeida Rodrigues, defendido em 2018).

Finalizando o eBook, apresentamos o texto adaptado do TCC do Luis Eduardo Matos, trabalhando a questão do uso dos sistemas de autoria code.org e Scratch na educação, em curso online no Moodle.

O trabalho do Luis se desenvolveu no âmbito do projeto Pensando em Códigos, de forma colaborativa com o trabalho de mestrado em Educação, defendido em

2018, do Augusto Márcio da Silva Júnior. Destaco que o Luis trabalhou tanto a Iniciação Científica, quanto seu TCC nesta temática, contribuindo com o Augusto no desenvolvimento e aplicação do curso básico sobre Scratch, com minha colaboração.

Apresentamos no texto um pouco sobre a estruturação do curso e alguns dados que possibilitam algumas inferências sobre o curso e seu potencial, sendo que a temática foi aprofundada na dissertação do Augusto e continua em desenvolvimento nas dissertações da Eliana Alice Brochado e Moisa Aparecida Silva, as quais trabalham o uso do Scratch na construção de narrativas digitais.

Como citado anteriormente, o projeto Pensando em Códigos está em seu terceiro ano de atividades formais e com apoio da Extensão, incluso para bolsistas (Gabriel Ribeiro Paiva—2016 e Lucas da Silva Fazio – 2017 – 2018, ambos do curso de Ciência da Computação).

Tivemos outras experiências envolvendo diretamente estudantes do curso de Ciência da Computação, os quais trabalharam em temáticas como ensino de Ciências/ Educação Ambiental (projeto DigiAtlas), além de projetos desenvolvimento de

aplicações para área da saúde, como o GlicoControl (Aplicativo Android para Controle de Glicemia) e AedesMap (Aplicativo Android para criação de mapas de calor da dispersão do mosquito e doentes).

Esperamos compartilhar nestes cinco capítulos um pouco da experiência de orientação de estudantes do curso de Ciência da Computação e as contribuições que esta área pode trazer para a educação, principalmente quando se desenvolvem projetos em parceria e com visão multi/ interdisciplinar, fundamentando os projetos, não apenas nas teorias educacionais, mas também nas teorias da Ciência da Computação, criando um sinergismo que potencializa de forma significativa a construção de conhecimentos para a área das tecnologias educacionais, seja no desenvolvimento de novas tecnologias digitais ou na compreensão de uso destas no contexto educacional.

Boa leitura para todos!

Prof. Gabriel Gerber Hornink

Capítulo 1 Histórico da informática e as implicações no ensino

*Raquel Ribeiro Balbino
Gabriel Gerber Hornink*

Diversos acontecimentos importantes marcaram a história da informática na educação, como a criação da linguagem Logo, em 1967, conferências, seminários e projetos, como o I CONTECE, EDUCOM, PRONINFE, ProInfo e UCA. Com as mudanças pedagógicas e socioculturais ao longo da história, houve uma preocupação em desenvolver *softwares* e ferramentas educacionais que acompanhassem a realidade dos usuários, que estavam cada vez mais utilizando a tecnologia para diversos fins. Esta demanda, acompanhada pelo avanço tecnológico das linguagens de programação e do hardware, permitiram a criação de ferramentas *online* que primeiramente ficaram conhecidas como web 1.0, marcada pela grande quantidade de informações disponíveis, mas com conteúdos pouco interativos, e mais tarde foram chamadas de web 2.0, quando ocorreu a mudança para uma internet como plataforma, permitindo ao usuário não apenas acessar, mas também editar e criar conteúdos coletivamente. Na década de 90 começaram a surgir os primeiros ambientes virtuais de aprendizagem (AVAs), como o WebCT em 1996, e o Moodle e o TelEduc em 2001. Apresenta-se neste capítulo um pouco dessa história e evolução cíclica das tecnologias e alguns de seus impactos para a educação.

1 Introdução

Desde a década de 50 que se realizam experimentos visando a inserção do computador na educação, mesmo que de forma primordial e simples, uma vez que o hardware tinha diversas limitações, quando comparado com os existentes, destacando a ausência de sistemas operacionais gráficos.

Um dos pontos críticos na história da informática no ensino se dá com o surgimento da internet, quando a utilização de computadores e a sua influência na sociedade cresce exponencialmente, fazendo com que as tecnologias digitais passassem também a serem utilizadas para construir ambientes de aprendizagem, possibilitando que os estudantes participassem ativamente do processo de construção do conhecimento.

Segundo Casas et al., (1996), o uso das tecnologias da informação e comunicação, no âmbito das digitais, que contam com recursos como multimídia, hipertexto, hipermídia, realidade virtual e telemática, podem oferecer flexibilidade, personalização, interatividade e qualidade no ensino, pois a educação e os processos de comunicação não são mais unidirecionais, a informação circula agora de forma

bidirecional, colaborativa e interdisciplinar (CAMPOS et al., 1998).

De toda forma, essa potencialização depende do modo que essas tecnologias digitais são utilizadas no contexto educacional.

Logo, é necessária a mudança na concepção da utilização do computador e demais tecnologias digitais para acompanhar as transformações nos processos de ensino-aprendizagem, além dos mais diversos fins que estes se inserem.

A evolução tecnológica, principalmente a criação de ferramentas da chamada web 2.0, como blogs, wikis, salas de bate-papo e fóruns, permitem a criação de ambientes que podem ser, em parte, semelhantes às atividades que acontecem em sala de aula e estas inovações repercutiram em mudanças no ensino a distância (EaD), que teve seu campo de abrangência largamente ampliado.

O avanço nas tecnologias digitais nas últimas décadas (2000-2018), somada a expansão do acesso à internet, assim como maior velocidade de banda, resultou no maior uso pela população que incorporou estas tecnologias nas mais variadas esferas de suas vidas. Destaca-se que o acesso à internet, seja em

domicílios, trabalho, escolas, *lan houses* ou outros locais atingiu 77,8 milhões de pessoas no segundo trimestre de 2011 segundo o Instituto Brasileiro de Opinião Pública e Estatística (IBOPE)(2011).

Esses valores aumentaram muito em 2016, sendo que 64,7% das pessoas com mais de 10 anos acessaram pelo menos uma vez a internet no primeiro trimestre do ano, cerca de 112,5 milhões de pessoas (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE, 2018).

Esta crescente interação da população com as tecnologias digitais da informação e comunicação (TIDCs), aliada às mudanças sociais e na forma de aprender, gera demandas de transformação dos tradicionais métodos de ensino, a fim de inserir a informática na educação de modo a potencializar o processo de ensino-aprendizagem e acompanhar a realidade dos estudantes que estão diariamente conectados, principalmente nas redes sociais, residindo aqui um aspecto relacionado com a inclusão digital.

Essa inserção das TIDCs na educação ocorre simultaneamente ao avanço tecnológico para o desenvolvimento de ferramentas que permitem que o estudante participe ativamente, interativamente e

colaborativamente no processo de aprendizagem, fazendo com que o computador e outras tecnologias digitais assumam um papel importante como instrumento sociocultural educacional para complementação e aperfeiçoamento do ensino (VALENTE, 1993).

Dentre as TDICs, os ambientes virtuais de aprendizagem (AVAs) são importantes tecnologias que permitem o armazenamento, administração, disponibilização de conteúdos no formato Web, comunicação, interação em um curso, seja ele a distância ou híbrido.

O Moodle e TelEduc são exemplos de AVAs, neste caso *softwares* livres, que também impactaram o ensino-aprendizagem a distância ou híbrido, facilitando a administração, disponibilização de conteúdos e potencializando as interações entre os participantes (ex. estudante-tutores-professores), usando ferramentas para aulas virtuais, fóruns de discussão *on-line*, *chats* (bate-papos), wiki (texto colaborativo), entre outros.

Todas estas ferramentas tecnológicas foram desenvolvidas à medida que as linguagens de programação e *hardwares* evoluíram, facilitando a programação de *softwares* e recursos web. Assim, torna-

se necessária a identificação e compreensão do histórico do desenvolvimento das tecnologias digitais da informação e comunicação e seus impactos na educação.

2 Histórico de utilização da informática na educação

Valente (1999) ressalta que desde a década de 50, quando surgiram os primeiros microcomputadores, que experimentos são realizados utilizando a informática na educação, apesar de o objetivo nesta época ser apenas armazenar e disponibilizar informações, assim, faz-se importante ressaltar que a realidade deste período difere-se muito da realidade atual dos dispositivos portáteis com tela interativa, internet móvel e maior capacidade de processamento e memória.

Contudo, é na década de 70 que as universidades no Brasil e em outros países começaram a realizar estudos e promover seminários e conferências sobre o assunto, como na Universidade de São Carlos em 1971, onde foi realizado um seminário sobre o uso da informática no ensino de Física e no Rio de Janeiro, onde aconteceu em 1971 a Primeira Conferência Nacional de Tecnologia em Educação Aplicada ao Ensino Superior – I CONTECE.

Em 1975, os professores Seymour Papert e Marvin Minsky, do laboratório Logo do *Massachusetts Institute of Technology* – MIT, visitaram a Unicamp e, a partir deste encontro, nasceu o projeto para a utilização no Brasil de computadores na educação utilizando a linguagem Logo (CHAVES, 1983).

Vários estudantes de pós-graduação se interessaram pelo projeto, como Maria Cecília Calani (1981), e nesse contexto, em 1983, foi criado o Núcleo de Informática Aplicada à Educação (NIED).

Também na Universidade Federal do Rio Grande do Sul (UFRGS), no início da década de 80, criou-se o Laboratório de Estudos Cognitivos (LEC), do Instituto de Psicologia, onde atuou a profa. Dra. Léa da Cruz Fugundes, o qual utilizou a linguagem Logo com crianças em escolas públicas que apresentavam estudantes com dificuldades de aprendizagem, buscando promover a aprendizagem autônoma dessas crianças (BONILLA; PRETTO, 2000; BUSTAMANTE, 2013).

A linguagem Logo foi desenvolvida por Seymour Papert e Wally Feurzeig no final dos anos 60, no MIT, em Cambridge. A equipe do NIED da Unicamp traduziu para o português do Brasil uma versão da linguagem chamada SuperLogo (VALENTE; ALMEIDA, 1997).

Trata-se de uma linguagem que foi desenvolvida para ser usada por crianças, com a filosofia da educação de inspiração piagetiana, em que a criança aprende explorando e criando o ambiente com regras que ela mesma impõe. Papert havia estudado com Jean Piaget entre 1958 e 1963 e incorporou à linguagem conceitos do construtivismo. A linguagem Logo geralmente é apresentada por uma tartaruga que se move no espaço ou na tela como resposta aos comandos que o usuário fornece por meio do computador (VALENTE; ALMEIDA, 1997).

A resposta imediata torna essa linguagem divertida e mais fácil de aprender, tornando este processo explícito e refletindo sobre o mesmo a fim de compreendê-lo e depurá-lo, o que é muito difícil de se conseguir por meio dos meios tradicionais de ensino (VALENTE, 1993).

A linguagem Logo exige que o usuário, por meio dos comandos para movimentar a tartaruga, explore conceitos de diferentes domínios, como matemática, física, etc e habilidades de resolução de problemas, planejamento e programação, além de noções de direção e orientação pelo fato de a tartaruga estar em algum lugar, ou seja, em uma posição (GREGOLIN, 2008).

Assim, possibilita-se no LOGO um ambiente de aprendizado onde o conhecimento não é passado para a criança, mas onde a criança interagindo com os objetos desse ambiente, possa desenvolver outros conceitos, por exemplo, conceitos geométricos e matemáticos (VALENTE, 1993).

Atualmente, a maioria das linguagens de programação não exige que o usuário tenha total domínio do processo de geração de código dos programas. Isso se deve às grandes facilidades de programação introduzidas nas linguagens, o que permite programar sem se preocupar com a maneira e todos os passos necessários para atingir os objetivos.

Destacam-se entre as linguagens lúdicas, aquelas que utilizam o *blockly*, um sistema de blocos de encaixe coloridos que possibilita a programação de forma intuitiva e divertida. Esta é usada em aplicativos como Scratch e App Inventor, ambos do MIT, sendo possível sua utilização em diversos contextos educacionais como sistemas de autoria pelos próprios estudantes, possibilitando, paralelamente ao desenvolvimento do conteúdo específico, o desenvolvimento do raciocínio lógico-matemático (SILVA; HORNINK, 2017).

Outro importante acontecimento da informática na

educação foi em 1981 na Universidade de Brasília, onde aconteceu o I Seminário Nacional de Informática em Educação, promovido pela SEI, MEC e CNPq. Este seminário teve como objetivo discutir as possibilidades de utilização do computador no processo de ensino-aprendizagem, enfocando aspectos teóricos e a aplicabilidade em todos os níveis de ensino. E em 1982 na Universidade Federal da Bahia aconteceu o II Seminário Nacional de Informática em Educação tendo como objetivo coletar subsídios para a criação de centros-piloto, a partir de reflexões de especialistas das áreas de educação, psicologia, informática e sociologia. (BUSTAMANTE, 2013; VALENTE, 1997).

Com estes seminários, o projeto EDUCOM (CHAVES, 1983) foi criado com os objetivos de implantar o Núcleo de Pesquisa e Desenvolvimento de Informática na Educação, implantar e acompanhar as experiências de centros-pilotos, além de permitir a formação de pesquisadores das universidades e de profissionais das escolas públicas com a realização de diversas ações iniciadas pelo MEC na década de 80, como:

- realização do Concurso Nacional de *Software* Educacional;
- implementação do FORMAR – Curso de

Especialização em Informática na Educação, visando a formação de professores das diversas secretarias estaduais de educação e das escolas técnicas federais;

- implantação nos estados do CIEd – Centros de Informática em Educação.

Mais ações do MEC na área de informática na educação podem ser encontradas no artigo de Maria Cândida Moraes (1997) e nos livros Projeto EDUCOM e Projeto EDUCOM: realizações e produtos (ANDRADE, 1993; ANDRADE, LIMA, 1993).

O Plano Nacional de Informática Educativa (PRONINFE) foi criado em 1989, pelo MEC, visando desenvolver a informática educativa no Brasil por meio de projetos e atividades, assegurando a unidade política, técnica e científica.

No fim da década de 90, com a iniciativa da Secretaria de Educação a Distância (SEED/MEC), foi criado o Programa Nacional de Informática na Educação (ProInfo) para introduzir a tecnologia na rede pública, melhorando a qualidade do processo de ensino e aprendizagem (BUSTAMANTE, 2013).

Vários estudos mostram os resultados do ProInfo no Brasil, entre eles, Marcelino (2003) mostra que, do ponto de vista de infra-estrutura física e tecnológica, os

índices de aprovação variaram de 90 a 93%. Porém, somente 22,5% técnicos e 10% dos coordenadores consideraram boas ou excelentes as condições de trabalho para suporte técnico às escolas, exatamente uma das funções críticas do NTE.

Em Goiânia, apesar de o ProInfo ter sido o programa de informática educativa mais importante do município, houve uma falha técnica e institucional, sendo que os computadores não foram atualizados e se tornaram equipamentos defasados, pouco aproveitados para a utilização de determinados *softwares* ou para o acesso à internet, e a assistência prestada às escolas mostrou-se precária e lenta (BARRA; MORAES, 2006).

Em 2007 foi criado com a iniciativa do Governo Federal o programa Um Computador por Aluno - (UCA), que visava distribuir a cada estudante da rede pública um laptop destinado à educação. Durante o ano de 2007 foram selecionadas 5 escolas, em cinco estados, como experimentos iniciais: São Paulo-SP, Porto Alegre-RS, Palmas-TO, Piraí-RJ e Brasília-DF. Posteriormente, centenas de escolas e milhares de estudantes foram beneficiados em todo o país.

Assim, os programas criados para inserção da tecnologia na educação foram surgindo à medida que

mudanças pedagógicas foram propostas, acompanhadas de mudanças socioculturais, visando a utilização do computador na educação não apenas como auxílio à memorização da informação pelo estudante, mas na construção do conhecimento realizada de maneira significativa, sendo o professor o facilitador desse processo (VALENTE, 1999).

Apresentaremos no próximo tópico uma linha do tempo com os principais acontecimentos da informática na educação, mostrando o impacto que o surgimento das linguagens de programação teve no desenvolvimento de novas tecnologias, que possibilitaram a criação dos ambientes virtuais de aprendizagem.

3 Avanços tecnológicos e as novas metodologias de aprendizagem

As inovações tecnológicas, como o surgimento da internet e a criação das linguagens de programação de alto nível¹, facilitaram o desenvolvimento de aplicativos e recursos cada vez mais acessíveis à população.

Isso fez com que a tecnologia ficasse mais presente na vida das pessoas, que hoje a utilizam no trabalho, no lazer, nas compras online e na educação,

1 Linguagem cujas instruções estão próximas do nível de compreensão humana (XAVIER, 2007).

colaborando para o surgimento do ensino online e dos ambientes virtuais de aprendizagem.

Para compreender um pouco mais a repercussão que as transformações tecnológicas tiveram nos meios de comunicação e interação e conseqüentemente no processo educacional, faz-se importante a visualização simultânea da evolução histórica das linguagens de programação, do surgimento de novas tecnologias e dos principais acontecimentos da informática na educação. Para tanto, construiu-se uma linha do tempo com os principais acontecimentos (Figura 1.1), sendo que cada cor de bandeira na figura representa algo diferente:

- bandeiras azuis: representam o surgimento das principais linguagens de programação;
- bandeiras laranjas: mostram as principais ferramentas de interação e comunicação;
- bandeiras vermelhas: são os principais ambientes virtuais de aprendizagem;
- bandeiras verdes: as redes sociais mais utilizadas no Brasil; e
- bandeiras amarelas: os principais acontecimentos que influenciaram e potencializaram a inserção da tecnologia na educação;

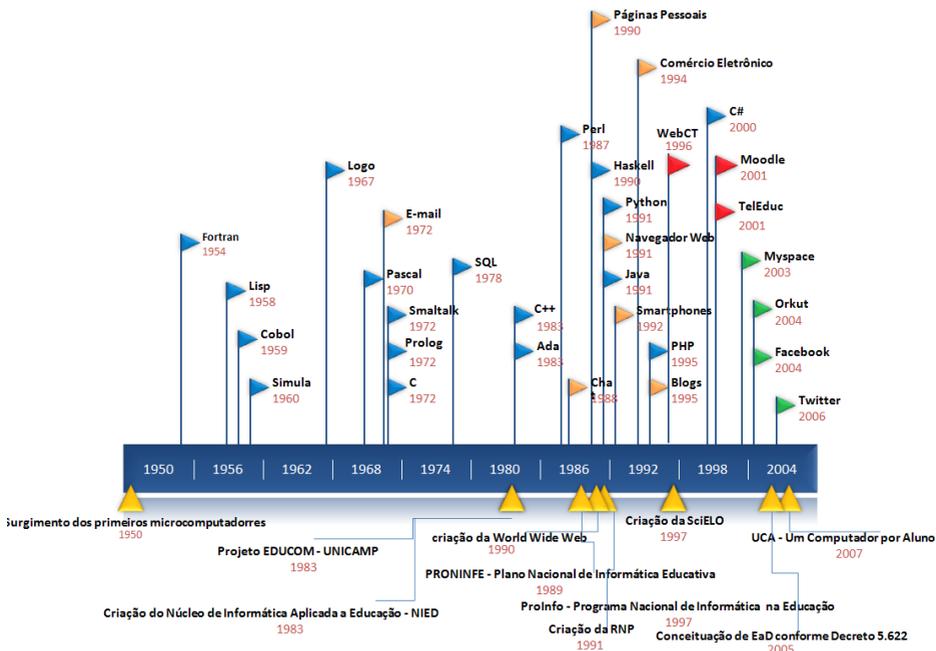


Figura 1.1 - Linha do Tempo.

Fonte: Os autores

A internet surgiu durante a guerra fria com objetivos militares e nas décadas de 1970 e 1980 também foi um importante meio de comunicação entre estudantes e professores universitários, principalmente dos EUA. Mas foi em 1990 que a Internet começou a se popularizar por meio do desenvolvimento do World Wide Web pelo físico e pesquisador Tim Berners-Lee do MIT (CNPQ, 2018).

Tinha início a primeira geração da web, conhecida como Web 1.0, marcada pela grande quantidade de informações disponíveis, mas com conteúdos pouco

interativos. No Brasil a internet surgiu em 1991 quando a Rede Nacional de Ensino e Pesquisa (RNP) passou a oferecer o acesso comercial à internet.

Desde a Segunda Guerra Mundial os avanços da tecnologia, principalmente a energia nuclear, despertaram os países para a importância da pesquisa científica. Vários avanços aconteceram nas décadas de 50 até a década de 80 quando foi criada a RNP, como o surgimento do CNPq, Conselho Nacional de Pesquisa e Desenvolvimento Científico e Tecnológico, em 1951 e no mesmo ano a criação da Campanha Nacional de Aperfeiçoamento de Pessoal de Nível Superior, atualmente a Capes (CNPQ, 2018).

Na década de 90, o CNPq estava transferindo algumas das suas funções para o Ministério da Ciência e Tecnologia (MCT) e, com a inserção da função social na produção tecnológica e científica, a missão do CNPq passou a ser, em 1995, "Promover o desenvolvimento científico e tecnológico e executar pesquisas necessárias ao progresso social, econômico e cultural do País", e isso resultou em ações que fizeram com que a RNP passasse a oferecer o acesso comercial à internet² (CNPQ, 2018).

2 Extraído do Regimento Interno do CNPq – Portaria nº 816, de 17 de dezembro de 2002 – Título I, Capítulo I, Artigo 2º.

A RNP conecta as principais universidades e institutos de pesquisa do país, sendo esta rede uma das mais avançadas do mundo, possuindo conexão com as principais redes acadêmicas internacionais (CNPQ, 2018).

A internet é utilizada em vários contextos como para fins científicos, onde o Núcleo de Informática Biomédica (NIB) da Universidade Estadual de Campinas (UNICAMP) criou em 1994 o projeto de publicação científica Hospital Virtual Brasileiro visando o intercâmbio de informações e interatividade entre profissionais e estudantes da área de saúde.

Logo após, o Grupo de Publicações Eletrônicas em Medicina e Biologia (e*pub), divisão do NIB, desenvolveu a primeira publicação científica brasileira, o *Online Journal of Plastic and Reconstructive Surgery* e, a partir dessa, aumentou-se o número de revistas científicas disponíveis na internet, sendo hoje um risco para uma revista científica não ter uma versão na web (AGUIAR, 1997; SOUZA, 2006).

No âmbito do uso da rede para ampliar e potencializar as produções científicas, facilitando o acesso a estas, surge em 1997 a Scientific Electronic Library Online (SciELO), como produto da cooperação

entre a FAPESP³ e BIREME⁴, sendo a primeira nacional e a segunda internacional. A Scielo consiste em um modelo de publicação eletrônica gerada pela cooperação de periódicos científicos digitais.

3.1 Web 2.0

Observa-se na Figura 1.1 que o uso da internet na década de 90 teve novas finalidades, para os fins pessoais e grupais, sendo muito utilizada para a criação de páginas com informações gerais, e-mail, mensagens, fotos, livro de visitas, entre outros elementos que mais tarde foram substituídos, recontextualizados e incrementados por meio dos blogs. Os sites pessoais, por exemplo, foram utilizados inicialmente para publicar o cotidiano da vida das pessoas, uma espécie de diário, e com o tempo passou a se tornar um canal de comunicação pessoal e empresarial (FERREIRA, 2008).

Esta interação do usuário com a web, permitindo o mesmo não apenas acessar, mas também editar e criar conteúdos coletivamente, aperfeiçoando-os com a colaboração dos usuários, é um dos fatores que marcaram a geração da chamada Web 2.0 (ARAYA; VIDOTTI, 2010).

3Website: <http://www.fapesp.br>

4Website: <http://www.bireme.br>

Este nome foi utilizado pela primeira vez em 2004 pela O'Reilly Media, referindo-se às novas formas de utilização da internet, onde o conteúdo deixa de ser estático e passa a ser dinâmico, interativo e flexível, podendo ser criado e editado facilmente pelos usuários, sendo as características comuns entre as empresas que conseguiram se manter com a crise da internet em meados do ano 2000 (ARAYA; VIDOTTI, 2010).

De acordo com Tim O'Reilly (2006) "Web 2.0 é a mudança para uma internet como plataforma e um entendimento das regras para obter sucesso nesta nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a inteligência coletiva (O'Reilly, 2006)" e esta coletividade permite a troca de informações e principalmente a colaboração, que é uma das palavras-chave da web 2.0, dos internautas com sites e serviços virtuais (ARAYA; VIDOTTI, 2010).

Segundo D'Ávila (2003), a colaboração, vista a partir de uma perspectiva sócio-histórico-cultural de Vygotsky, é o espaço definido, por excelência, para as interações cognitivas e também sociais.

Para Corrêa (2004), alinhado com as ideias de

Vygotsky, para que o conhecimento seja criado é necessário que haja o compartilhamento de saberes, de conhecimento e de opiniões. Nesse sentido, a Web 2.0 apresenta o reflexo de uma mudança de paradigma sobre o processo de construção do conhecimento.

Uma característica marcante da Web 2.0, de acordo com Alexander (2006), tanto ideologicamente quanto tecnologicamente é a abertura, que teve início com o movimento do software livre comandado por Richard Stallman em 1983 (PEREIRA, 2004).

Nesta época, começaram a surgir os conteúdos de código aberto, permitindo que os programas e serviços fossem reutilizados, corrigidos, alterados e melhorados (PEREIRA, 2004).

Alguns exemplos de aplicações que marcam a transição da web 1.0 para a web 2.0 são os blogs, wikis, salas de bate-papo, fóruns, sites de compartilhamento de arquivos e redes sociais.

A web 3.0 terá como foco principal a colaboração, onde será possível criar sistemas de conhecimentos coletivos na qual os usuários poderão compartilhar a informação, utilizando diferentes dispositivos que tenham acesso à internet como computador, celular ou qualquer outro dispositivo móvel (ISOTANI et al., 2008).

Dentre essas ferramentas, o fórum é um elemento de troca de informações e conhecimentos que se tornou comum na internet. Nesta época haviam tecnologias como a linguagem PHP, JavaScript e MySQL, que também possibilitaram mais tarde o surgimento dos softwares colaborativos, também chamados de wikis.

O fórum possui características como:

- defasagem de tempo bastante variada, podendo o usuário postar e responder mensagens com segundos entre uma mensagem e outra, ou então dias, semanas ou até meses, dependendo das regras do fórum;
- armazenamento das mensagens do fórum na Web, organizadas por linhas de discussão;
- organização das mensagens por tópicos de assuntos.

Estas características fazem do fórum uma ferramenta apropriada para a socialização ou discussão mais aprofundada de temas (ARAÚJO, 2002).

Por ser assíncrono, permite aos usuários controlar o tempo, o lugar, o ritmo e a natureza da interação, com a opção de responderem imediatamente ou refletir e pesquisar antes de se pronunciarem (HARASIM, 1998).

As salas de bate-papo e os serviços de e-mail

foram os que mais trouxeram popularidade à internet. As salas de bate-papo, também conhecidas como chats, permitem conversas em tempo real entre pessoas interessadas em um mesmo assunto e hoje.

Um dos maiores bate-papos do mundo é da empresa UOL, lançado em 1996, com mais de 7800 salas e altíssimo acesso. Por exemplo, no período de 09 a 15 de fevereiro de 2018, a Google Analytics registrou mais de 4.087.054 visitas, sendo que este total de visitas registrou 2.470.415 horas de uso da plataforma, com pico de registro de mais de 79.791 usuário simultâneos (UOL, 2018).

As principais diferenças entre o chat e o fórum é que o chat proporciona a comunicação em tempo real (síncrona), as mensagens são mais curtas e a interação geralmente não exige muito formalismo. Nos fóruns as mensagens costumam ser mais trabalhadas, elaboradas e objetivas, podendo ocorrer de forma não simultânea (assíncrona).

O correio eletrônico é anterior ao surgimento da internet e surgiu a partir de troca de mensagens entre computadores em rede. Porém, apenas em meados de 1996 que surgiram os primeiros serviços de e-mail.

Nas redes sociais, o conhecimento é aberto e

colaborativo e para a EaD, isto significa que o estudante, além de leitor, passa também a ser autor e produtor de material para a educação, e inclusive editor e colaborador (MATTAR, 2011).

Pereira (2010) mostra um caso de sucesso do uso de redes sociais na educação, onde um grupo de brasileiros e portugueses se reúnem no *Twitter* continuamente para discutir EAD, utilizando a *hashtag* #eadsunday.

O *Twitter* é uma rede social, criada em 2006 por Jack Dorsey, que permite aos usuários enviar e receber atualizações pessoais conhecidos como *tweets*. O *twitter*, no processo de ensino e aprendizagem, pode melhorar a integração dos estudantes e incentivar a autonomia de pesquisa na rede e o compartilhamento de soluções (SILVEIRA, 2009).

Além disso, acaba por estimular a habilidade de síntese dos textos, uma vez que só é possível escrever frases com até 140 caracteres.

3.2 Relação entre a evolução nas linguagens de programação, o desenvolvimento de novas tecnologias e o uso da informática na educação

Existe uma relação cíclica entre a evolução nas linguagens de programação (tecnologias digitais em um

modo geral), o desenvolvimento de novas tecnologias e o uso da informática na educação (HORNINK, 2010).

A interação entre as pessoas teve uma mudança significativa com o advento da eletricidade e das redes fixas de comunicação, como telégrafo e com o telefone e também em meados do século XX com a disseminação do rádio e da televisão, sendo o início da indústria e das teorias de comunicação de massa (CYSNEIROS, 2010).

Com a crescente interação entre os usuários e os meios de comunicação, assim como com a criação da *world wide web* (www) em 1990, novas tecnologias digitais surgiram, quando observou-se um aumento exponencial no desenvolvimento destas, assim como na facilidade e disponibilidade de recursos para o desenvolvimento.

Neste contexto, criaram-se várias linguagens de programação e ferramentas, incluindo as voltadas para a educação. Nesta época, discutia-se a importância da interação e do computador no processo educacional, que precisava ser repensado para acompanhar a realidade dos estudantes.

Segundo Cysneiros (2010), muitas das colocações sobre dialogismo se aplicam à interação mediada por tecnologias onde o diálogo é um princípio que permeia as

grandes teorias educacionais.

Cysneiros (1998) descreveu o ciclo entre interação, tecnologia e educação começando com a chegada de uma nova tecnologia, que gerava elevadas expectativas sobre sua importância na Educação, acompanhada de um discurso de obsolescência da escola, havendo necessidade de mudanças; estabeleceram-se políticas públicas de introdução dos objetos técnicos, fechando-se o ciclo com o uso limitado da tecnologia, já não tão nova.

Essa interação do estudante na construção do conhecimento, onde ele deixa de ser apenas um espectador da aula e passa a assumir um papel onde ele pode interagir com o conteúdo, buscando e podendo modificar e incrementar informações é extremamente importante na realidade vigente, onde os processos ocorrem de maneira muito rápida e imperceptível e as competências adquiridas por uma pessoa no começo do seu percurso profissional, serão obsoletas no fim de sua carreira (LÉVY, 2000).

Assim, o computador passa a ser uma ferramenta educacional, de complementação e aperfeiçoamento do ensino (VALENTE, 1993).

3.3 Novas metodologias de aprendizagem

O avanço tecnológico permitiu mudanças tanto no ensino presencial, como também no EAD ou mesmo ensino híbrido, que teve suas origens no final do século XIX com a criação, em diferentes países, de instituições que ofereciam cursos por correspondência, enviando material ao estudante, seja na forma escrita, em vídeos ou CD-ROMs e visava, principalmente, atingir um setor da população que não tinha outra forma de acesso à educação por razões geográficas, por falta de escolas próximas, ou por outras dificuldades.

Além do tempo que se gastava até a chegada do material do EAD ao estudante, o envio das atividades ao professor e novamente a correção para o estudante, o custo de postagem e fabricação dos materiais impressos ou CD-ROMs também era muito alto. Depois do advento da Internet, do e-mail e de todos os recursos tecnológicos disponíveis, ampliou-se largamente o campo de abrangência da EaD.

Atualmente os cursos possuem várias opções gratuitas de ambientes virtuais de aprendizagem (AVAs) para auxiliar na construção e gerenciamento de cursos a distância.

Ressalta-se que concepção do EAD se alterou a

medida que as tecnologias digitais e, simultaneamente, as metodologias, evoluíram, saindo de uma visão unidirecional, com foco na entrega da informação, para uma visão colaborativa, e com foco na comunicação mediada pelo computador ou outras tecnologias (HORNINK, 2010).

Os recursos tecnológicos viabilizaram a criação de ambientes que podem ser semelhantes às atividades que acontecem em sala de aula, entretanto, sendo instrumentos de mediação diferentes, resultaram em processos de comunicação e aprendizagem diferenciados, seja nas relações sociais, afetivas ou cognitivas (HORNINK, 2010).

Por exemplo, as web-conferências possibilitam o encontro de várias pessoas localizadas em lugares diferentes, compartilhando áudio, imagem e arquivos. O chat permite a interação de um grupo de pessoas discutindo sobre determinado tema e com a presença do professor como mediador.

Há também os fóruns onde os participantes se manifestam expressando suas opiniões ou tirando dúvidas com o professor. Por fim, o e-mail que pode ser utilizado também para a troca de mensagens e materiais entre os participantes.

Porém, para que ocorra aprendizado, não basta apenas o desenvolvimento das ferramentas e dos ambientes virtuais de aprendizagem, é preciso que os professores tenham uma preparação e treinamento para utilizar os recursos educacionais (PRADO; VALENTE, 2003).

Para Valente (2009), esta preparação é muito mais profunda do que simplesmente montar laboratórios de computadores nas escolas e formar professores para utilizá-los. É preciso auxiliá-los para desenvolverem conhecimentos sobre o próprio conteúdo e sobre como o computador pode ser integrado no desenvolvimento desse conteúdo. Sem um projeto pedagógico atuando em conjunto com as tecnologias, os AVAs podem, por exemplo, serem apenas repositório de arquivos, sem atividades que promovam a interação e colaboração entre os usuários.

Assim como a wiki, se não utilizada corretamente, passa a ser apenas um espaço de postagem de respostas, sem a colaboração que é o principal objetivo da ferramenta.

4 Inovações na programação e as repercussões no processo educacional

A partir da visão do uso das TICs, abordado anteriormente, focaremos nos aspectos constitutivos (do ponto de vista técnico) para uma análise sobre os ambientes virtuais de aprendizagem e outras TICs, sendo o primeiro elemento de grande importância a linguagem de programação.

De acordo com Lorenzi e Lopes (2000), uma linguagem de programação é uma abstração destinada ao uso e à exploração dos recursos de um computador, constituindo, de acordo com Walker (2008), o meio de comunicação entre o usuário e o computador.

Apresentaremos algumas linguagens e paradigmas de programação, com uma breve descrição e o contexto histórico de cada uma, assim como mostraremos a repercussão que algumas linguagens tiveram no processo educacional, como o desenvolvimento de *softwares* educativos e ambientes virtuais de aprendizagem.

Atualmente, segundo o site Tiobe (2018), a linguagem de programação Java é a mais utilizada no mundo, seguida por C e C++ (Tabela 1.1).

Tabela 1.1 - Ranking das principais linguagens de programação.

Jul 2018	Jul 2017	Variação	Programming Language	Ranking %	Variação %
1	1	=	Java	16,139	2,37
2	2	=	C	14,662	7,34
3	3	=	C++	7,615	2,04
4	4	=	Python	6,361	2,82
5	7	↑	Visual Basic .NET	4,247	1,20
6	5	↓	C#	3,795	0,28
7	6	↓	PHP	2,832	-0,26
8	8	=	JavaScript	2,831	0,22
9	-	↑↑	SQL	2,334	2,33
10	18	↑↑	Objective-C	1,452	-0,44
11	12	↑	Swift	1,412	-0,84
12	13	↑	Ruby	1,293	-1,05
13	14	↑	Assembly language	1,154	-1,09
14	15	↑	R	1,150	-0,95
15	17	↑	MATLAB	1,130	-0,88
16	9	↓↓	Delphi/Object Pascal	1,109	-1,38
17	11	↓↓	Perl	1,101	-1,23
18	10	↓↓	Go	0,969	-1,39
19	16	↓	Visual Basic	0,885	-1,21
20	20	=	PL/SQL	0,704	-0,84

Fonte: Tiobe, 2018

Na Tabela 1.1, pode-se observar, no período de julho de 2017 e julho de 2018 as colocações e as variações no uso das linguagens, indicando se houve

manutenção, aumento ou queda no *ranking*.

Apresenta-se na Figura 1.2 o gráfico da evolução histórica (2002 – 2018) no ranking das dez principais linguagens de programação. Observa-se o predomínio do Java e C, apesar de, historicamente, terem sido muito mais utilizadas, com um grande aumento após 2017, quando houve uma queda em seus usos. Além disso, uma queda significativa no uso do C++ e PHP, apesar de um pequeno aumento do C++, este ainda está bem abaixo quando comparado ao período de 2002 à 2005.

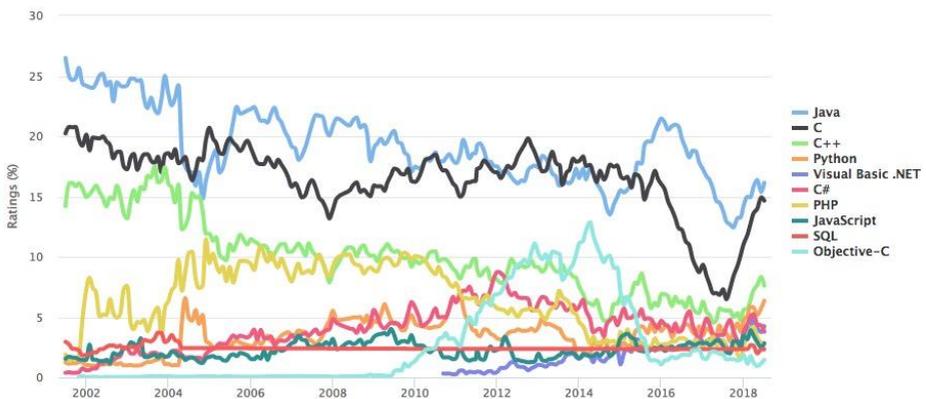


Figura 1.2 - Evolução histórica dos rankings das principais linguagens de programação.

Fonte: TIOBE, 2018.

Olhando-se um pouco mais para o histórico dessas linguagens, tendo em consideração as dez mais usadas em 2018, tem-se na Tabela 1.2 um comparativo do ranking de uso a partir de 1988 até 2018.

Tabela 1.2 - Ranking das principais linguagens entre 1988 e 2018.

Linguagem	2018	2013	2008	2003	1998	1993	1988
Java	1	2	1	1	16	-	-
C	2	1	2	2	1	1	1
C++	3	4	3	3	2	2	5
Python	4	7	6	11	23	18	-
C#	5	5	7	8	-	-	-
Visual Basic.NET	6	12	-	-	-	-	-
JavaScript	7	10	8	7	20	-	-
PHP	8	6	4	5	-	-	-
Ruby	9	9	9	18	-	-	-
R	10	23	46	-	-	-	-
Perl	12	8	5	4	3	11	-
Objective-C	17	3	40	53	-	-	-
Ada	27	18	18	14	9	5	3
Fortran	30	25	20	12	6	3	15
Lisp	31	11	15	13	7	6	2

Fonte: TIOBE, 2018.

As linguagens de programação são anteriores ao advento dos primeiros microcomputadores e alguns autores, como Sebesta (2002), as classificam em quatro paradigmas: imperativo, funcional, lógico e orientado a objetos.

Apresenta-se no Quadro 1.1 algumas características das linguagens que serão discutidas a seguir, indicando-se o paradigma, vantagem e desvantagem.

Quadro 1.1 - Características de algumas linguagens de programação.

Ling.	Paradigma	Vantagens	Desvantagens
Ada	Imperativo	Programação em tempo real, sincronização primitiva para processos e programação concorrente	Rigidez sintática
C	Imperativo	Portabilidade, velocidade e tamanho dos programas	Maior tempo de desenvolvimento, sujeito a erros
Cobol	Imperativo	Capacidade de autodocumentação e bons métodos de manuseio de arquivos	Lentidão, decorrente da autodocumentação
Fortran	Imperativo	Comandos curtos para operações matemáticas com matrizes e disposições de tipos (versão > 95)	Sintaxe defasada, sujeito a erros
Pascal	Imperativo	Fortemente tipada, o que auxilia os iniciantes	Dificuldade para modelar
Haskell	Funcional	Códigos curtos, claros e fáceis de manter; maior confiabilidade	Maior distância do hardware
Prolog	Lógico	Versátil, eficiente, base para muitos dos mecanismos de inferência da IA	Limitação de recursos para estruturação de objetos complexos e programação de larga escala
Java	Orientado a objetos	Portabilidade, segurança, simplicidade na especificação	Pouca integração com o sistema operacional
PHP	Orientado a objetos	Fácil aprendizado, menor complexidade para páginas web com acesso a banco de dados, multiplataforma e robustez	Compatibilidade entre versões, documentação incompleta, alguns problemas de segurança (depende

			da versão)
C++	Orientado a objetos	Portabilidade, brevidade no código, programação modular, velocidade (código eficiente), adequado para grandes projetos	Tempo para aprendizagem, maior necessidade de padrões, falta na biblioteca padrão para <i>threads</i> , conexões TCP/IP, e outras, problemas com alguns compiladores.
C#	Orientado a objetos	Alto poder de abstração, boa capacidade de virtualização, multiplataforma, simplicidade, mais fácil de aprender (comparado com C e C++), fortemente tipada, flexibilidade	Programas e componente antigos precisam ser reescritos.

Fonte: Os autores.

4.1 Paradigma Imperativo

As linguagens imperativas organizam o programa sob a forma de uma série de instruções, agrupadas por blocos e permitindo saltos condicionais que permitem retornar a um bloco de instruções se a condição for realizada.

Foram projetadas em torno da arquitetura conhecida como arquitetura de Von Neumann, onde tanto os dados como os programas são armazenados na mesma memória e a unidade central de processamento (CPU), que executa realmente as instruções, é separada

da memória.

As principais características das linguagens imperativas são:

- variáveis;
- comandos de atribuição;
- execução sequencial de instruções;

São exemplos de linguagens imperativas: C, Fortran, Cobol, Basic, Modula-2, Ada e Pascal.

4.1.1 Linguagem C

A linguagem C tem sido utilizada em muitos problemas de programação como sistemas operacionais, planilhas de texto e hoje estão disponíveis compiladores eficientes para máquinas de todos os tipos de capacidade de processamento (COCIAN, 2004).

Grande parte dos programas Linux e quase todo o Kernel foram escritos em C (MAUERER, 2008).

C foi desenvolvida no AT&T Bell Labs entre 1969 e 1973 para a implementação do sistema Unix⁵, que originalmente foi escrito em um PDP-7 em Assembly⁶,

5 Sistema operacional descrito em 1974 originalmente por Ken Thompson e Dennis Ritchie (BACH, 1986).

6 Linguagem de baixo nível, cujas instruções estão mais próximas ou correspondem quase diretamente ao código de máquina (Polloni e Fedeli, 1899).

por Dennis Ritchie e Ken Thompson.

Em 1973, com a adição do tipo `struct`⁷, a maioria das partes do núcleo do Unix foram reescritas em C.

As principais características da linguagem incluem:

- portabilidade entre máquinas e sistemas operacionais;
- dados compostos em forma estruturada;
- programas extremamente rápidos e pequenos;
- “alto nível” e “baixo nível” ao mesmo tempo, permitindo o controle total da máquina por parte do programador e permitindo efetuar ações sem depender do sistema operacional utilizado;
- fácil acesso ao hardware quando necessário.

4.1.2 FORTRAN

A Linguagem Fortran, *Formula Translator*, foi desenvolvida em 1955 por uma equipe liderada por John Backus e foi uma das primeiras linguagens de alto nível da história, projetada especificamente para aplicações científicas (COELHO, 2007).

Melhorias na linguagem foram surgindo com novas versões como em 1958 com o Fortran II e logo depois o

⁷ Estrutura que agrupa várias variáveis em uma só, podendo ser utilizada para agrupar um conjunto de características que descrevem um objeto, formando um novo tipo de dados (Cocian, 2004).

Fortran III que permitia a inclusão de instruções em linguagem Assembly. Em 1961 apareceu o Fortran IV que foi uma grande revolução na programação de computadores de grande porte e em 1966, o Fortran 66 padronizando a linguagem e permitindo que ela fosse utilizada em qualquer computador.

Com isso, Fortran passou a ser a linguagem mais usada na indústria e nas universidades e em 1977, com o advento dos microcomputadores, surgiu o Fortran 77, seguido do Fortran 90 e do Fortran 95, que de acordo com Leite (2006), traziam novidades como:

- aparecimento de ponteiros;
- edição de texto em software para windows;
- instruções ou expressões poderiam começar em qualquer linha;
- estrutura de repetição "do while" e de seleção "select case";
- possibilidade de alocação de memória.

4.1.3 Pascal

A linguagem Pascal, nome do filósofo francês Blaise Pascal, foi criada em 1970 pelo Dr. Niklaus Wirth para servir de ferramenta no ensino de programação e pela visão de que a maior parte dos erros de

programação poderiam ser evitados a partir da utilização de uma linguagem estruturada em blocos e dotada de um severo controle de tipos.

Pascal se tornou tão popular que acabou por penetrar também nas áreas científica e cultural (LORENZI; LOPES, 2000).

Trata-se de uma linguagem de alto nível, bem estruturada, mas com regras rígidas, o que a torna difícil de modelar.

É muito usada para iniciar os cursos de Programação por ser fortemente tipada, e existem vários ambientes de desenvolvimento (IDE) para a linguagem como o FreePascal, o Kylix e o Delphi.

4.1.4 Ada

Ada é uma Linguagem de programação estruturada, originada de Pascal e outras linguagens. Foi originalmente produzida por uma equipe liderada por Jean Ichbiah da CII Honeywell Bull, contratados pelo Departamento de Defesa dos Estados Unidos durante a década de 70, com o intuito de substituir as centenas de linguagem de programação usadas pelo DoD.

Ada é uma aplicação com compiladores validados para uso confiável em missões críticas, tais como

softwares de aviação. Outras vantagens dessa linguagem referem-se a suporte para programação em tempo real, incluindo primitivas para sincronização de processos e programação concorrente (POLLONI; FEDELI, 1999).

O nome Ada é uma homenagem a condessa de Lovelace, Augusta Ada Byron, filha de Lord Byron que trabalhou com Charles Babbage e é considerada como a primeira programadora de computadores da história. De concepção moderna e avançada, Ada contém conceitos de reusabilidade, portabilidade, modularidade e confiabilidade, princípios originários da Engenharia de Software (POLLONI; FEDELI, 1999).

4.1.5 COBOL

A linguagem Cobol, Common Business Oriented Language, foi desenvolvida no final da década de 50 por um grupo pertencente ao Comitê CODASYL, com o objetivo de ser uma plataforma de desenvolvimento para aplicações bancárias e financeiras em geral. Algumas vantagens da linguagem são:

- excelentes capacidade de auto-documentação;
- bons métodos de manuseio de arquivos;
- boa modelagem de dados para a época em que a linguagem foi desenvolvida, graças ao uso da cláusula

PICTURE para especificações detalhadas de campos.

Inicialmente, o COBOL foi proposto para solução de problemas do governo e militares dos EUA, entretanto, teve amplo uso comercial e, ainda hoje, é utilizado por empresas, como instituições financeiras, ressaltando que é pouco utilizado para construção e novos softwares. Ao fim de 1990, o Gartner group (empresa de pesquisa na área de processamento de dados), estimou que 80% dos códigos fontes construídos no mundo (240 bilhões de linhas) foram desenvolvidos em Cobol (VELLOSO, 2017).

4.2 Paradigma Funcional

O Paradigma funcional consiste em desenvolver funções que solucionam um determinado problema. Ele visualiza o processo de desenvolvimento de programas como a conexão de expressões, funções e declarações predefinidas, em que cada uma recebe entradas e produz saídas de forma a se produzir o relacionamento desejado de entrada-saída (BROOKSHEAR, 2003).

Este paradigma é bastante interessante quando aplicado em ambientes onde funções matemáticas são muito utilizadas, como cálculos recursivos (COSTA, 2010). Como exemplos de linguagens funcionais, tem-se: Lisp e Haskell.

4.2.1 Haskell

Haskell é uma linguagem de programação puramente funcional, de propósito geral, criada em 1990 e nomeada em homenagem ao lógico Haskell Curry. A linguagem possui características como códigos mais curtos, claros e fáceis de manter; menor quantidade de erros; maior confiabilidade e uma distância semântica menor entre o programador e a linguagem, o que melhora a produtividade do programador (FRANCISCO, 2006).

Existem vários compiladores disponíveis livremente para diferentes plataformas.

4.3 Paradigma Lógico

A ideia básica da programação em lógica é que um algoritmo é constituído de dois elementos distintos: a lógica, que define a solução do problema e o controle, que define como a solução do problema é alcançada.

Assim, um programa especificado no paradigma lógico é fundamentalmente um conjunto finito de sentenças lógicas, denominadas cláusulas, que expressam o problema a ser solucionado. O paradigma lógico foi muito utilizado nos anos 80 em projetos de computadores de quinta geração e teve um aumento

significativo em sua utilização a partir das pesquisas de Inteligência Artificial.

Na educação, em 1978, o prof Kowalski fez uma experiência usando o Prolog nos computadores do Imperial College do Park House Middle School, em Wimbledon (KREPSKY, 1999).

Na Inglaterra, em 1980, o governo desenvolveu um programa de incentivo para o ensino da programação em lógica, conhecido como “Lógica como Linguagem de Programação para crianças”, e teve muito sucesso no desenvolvimento da capacidade de aprendizado de informática e também no desenvolvimento do pensamento lógico, por meio da linguagem lógica (PALAZZO, 1997).

4.3.1 Prolog

Prolog é uma linguagem que nasceu na década de 70 para usar lógica com uma linguagem de programação e os desenvolvedores desta idéia foram Robert Kowals, Maarten Van Emdem e Alain Comerauer (ABE; SCALZITTI; SILVA Filho, 2002).

Trata-se de uma linguagem declarativa e pode ser usada em (CLOCKSIN; MELLISH, 2003):

- bancos de dados relacionais;

- lógica matemática;
- solução problemas abstratos;
- muitas áreas da inteligência artificial;

Estas e outras características constituem uma estrutura poderosa e flexível de programação.

4.4 Paradigma orientado a objetos

A orientação a objetos é um paradigma baseado na composição e interação entre diversas unidades de software chamadas de objetos (COSTA, 2010).

Segundo Ferrante e Rodrigues (2000), este paradigma foi introduzido em 1960 com a implementação da linguagem de programação Simula e é constituído de classes, que representam os objetos e possuem comportamentos (métodos), estados (atributos) e relacionamentos.

As linguagens C++, C#, Java são exemplos de linguagens de orientadas a objetos muito utilizadas atualmente, principalmente na educação, sendo Java a linguagem mais utilizada no mundo.

4.4.1 Java

A linguagem de programação Java é concorrente, baseada em classes, orientada a objetos e de propósito geral. Foi implementada em 1990 pela *Sun Microsystems*

e sua principal característica era a portabilidade já que seria executada em qualquer plataforma, independente do sistema operacional utilizado (MATTOS, 2007).

Java é uma linguagem compilada e interpretada, ambas parcialmente, gerando o compilador de arquivos objetos chamados de byte-codes.

A Máquina Virtual Java (JVM) é um interpretador desenvolvido para cada plataforma que transforma os *byte-codes* em linguagem de máquina.

A linguagem Java tornou-se uma importante ferramenta para a educação, principalmente no ensino a distância. Na área das ciências, ela está por detrás dos "Laboratórios Virtuais", ambientes que simulam determinado fenômeno físico e rodam em pequenos programas, conhecidos como Applets (FIGUEIRA, 2005).

4.4.2 PHP

A linguagem *Hypertext Preprocessor* - (PHP⁸) foi criado em 1994 por Rasmus Lerdorf e escrita em Perl, sendo reescrita para C para incluir acesso a banco de dados e a primeira versão foi disponibilizada em 1995, conhecida como "Personal Home Page Tools" (MORAZ, 2005).

8 A linguagem possui recursos de orientação a objetos a partir da versão PHP 3.

Em julho de 2018, a linguagem se encontrava na versão 7.3.0, de código aberto e desenvolvida para a web. A versão 8 tem previsão de ser lançada em 2021.

O PHP reduz a complexidade do processo de desenvolvimento de páginas dinâmicas, além de simplificar o acesso aos diversos bancos de dados existentes (ESTROZI et al., 2010).

Estas e outras vantagens permitem que a linguagem tenha uma significativa importância para a educação, principalmente para o ensino a distância, uma vez que os principais ambientes virtuais utilizados são escritos em PHP, como o Moodle e o Teleduc.

5 Ambientes Virtuais de Aprendizagem (AVA)

Os Ambientes Virtuais de Aprendizagem (AVAs) estão sendo cada vez mais utilizados como uma opção tecnológica educacional, sendo classificados pelo Ministério da Educação (2007, p. 11) como:

Os ambientes virtuais de aprendizagem são programas que permitem o armazenamento, a administração e a disponibilização de conteúdos no formato Web. Dentre esses, destacam-se: aulas virtuais, objetos de aprendizagem, simuladores, fóruns, salas de bate-papo, conexões a materiais externos, atividades interativas, tarefas virtuais (webquest), modeladores, animações, textos colaborativos (wiki).

Vale destacar que o objetivo dos AVAs não é diminuir a distância física entre os participantes do EAD e sim promover um ambiente que auxilie no ensino, independente da modalidade (presencial, a distância ou híbrido).

Moore (1993) considera a distância educacional não do ponto de vista físico, mas do ponto de vista comunicativo.

Após a criação do primeiro navegador para a web no final de 1990, a internet deixou de ser uma rede acadêmica e passou a incorporar vários tipos de atividades, onde surgiram os primeiros projetos de construção de ambientes virtuais de aprendizagem.

As primeiras versões dos AVAs para educação foram modeladas com base em quatro estratégias, com relação às suas funcionalidades (FRANCO et al., 2003):

- Incorporar ferramentas já existentes na web;
- Agregar elementos para atividades específicas de informática;
- Criar elementos específicos para a atividade educacional;
- Adicionar elementos de administração acadêmica sobre cursos, estudantes, avaliações e relatórios.

Com a criação dos ambientes virtuais de aprendizagem, a web passou a ser bastante utilizada para auxiliar os cursos de graduação e pós-graduação e também auxiliar as empresas em treinamentos e cursos a distância e também presenciais.

Um dos primeiros AVAs foi o *Web Course Tools* (WebCT), atualmente denominado *Blackboard*. Foi desenvolvido em 1996 pelo Departamento de Ciência da Computação da Universidade de British Columbia e possui características como (HAGUENAUER et al., 2003):

- interface amigável;
- ferramentas que facilitam a aprendizagem e a colaboração;
- ferramentas de comunicação: chat, discussions, e-mail e whiteboard;
- ferramentas de avaliação: tarefas, notas, testes;
- ferramentas de estudo: *bookmarks*, *my progress*, busca, dicas, *home pages* dos estudantes, apresentação do estudante, anotações;
- ferramentas de conteúdo: calendário, CD-rom, compilação do material, mapa do curso, glossário, banco de imagens, index, syllabus (visualização do conteúdo *offline*).

As páginas do sistema foram desenvolvidas em HTML⁹ e JavaScript, sendo os dados armazenados em arquivos e manipulados por CGI usando a linguagem de programação Perl.

Apresenta-se na Figura 1.3 uma das telas da interface do WebCT, destacando as principais funcionalidades da ferramenta, como: calendário, materiais do curso, ferramentas de comunicação, chat, glossário entre outras.

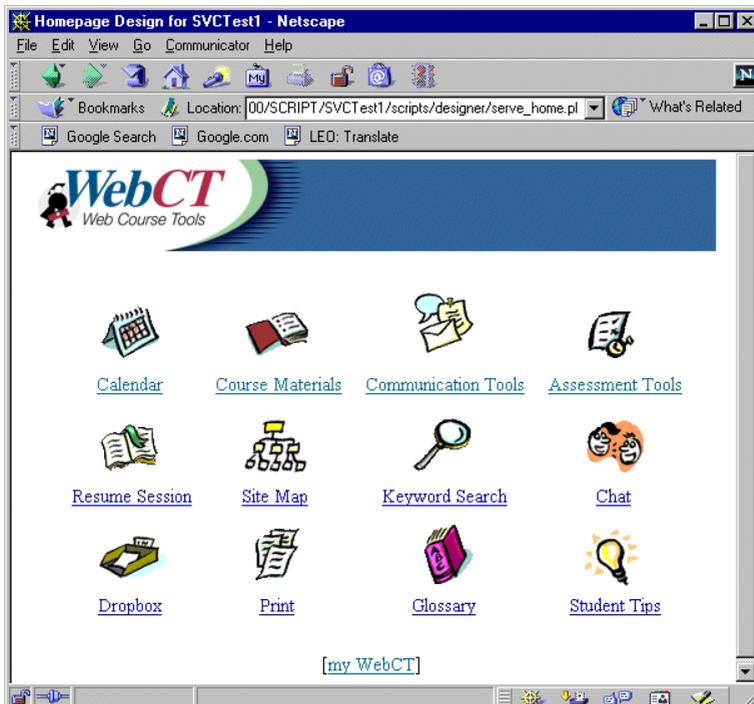


Figura 1.3 - Tela principal da Interface do AVA WebCT.
Fonte: BRUGGER, 2000

Outra AVA de importância nacional e internacional foi o Teleduc, o qual teve sua produção iniciada em 1997, como parte da dissertação de mestrado da Alessandra de Dutra e Cerceau, a qual foi orientada pela profa. Dra. Heloísa Vieira da Rocha do NIED-Unicamp. Este foi desenvolvido inicialmente em php, JavaScript, Perl e MySQL, para servidores Linux (TELEDUC, 2018).

O TelEduc é um ambiente para a criação, participação e administração de cursos na Web, visando desenvolver e depurar ferramentas de acordo com as necessidades dos usuários. Entre 1998 e 2001 o Teleduc teve seu uso intensificado na Unicamp e, gradativamente, a ferramenta foi aprimorada até sua abertura pública em fevereiro de 2001 (TELEDUC, 2018).

O TelEduc foi aberto para a comunidade e transformado em uma licença de software livre, ocorrendo um grande aumento do seu uso na área acadêmica. Sua distribuição é livre e as linguagens utilizadas para a sua criação são PHP para praticamente todas as páginas geradas no ambiente, Perl utilizado pela ferramenta bate-papo para exibição de mensagens, criptografia de senhas e *upload* de arquivos e Java usada na ferramenta Intermap (SIMÕES; LINDEMANN, 2007).

Apresenta-se nas Figuras 1.4 e 1.5 a visualização

inicial do ambiente, iniciando pela agenda, das versões 3.x e 4.x do *software*.

Projeto ensino FAPESP/PETROBRAS
Agenda - 2010 Busca Atualizar

Agendas Anteriores Editar Agendas

Ribeirão Anhumas na Escola

Apoio: **FAPESP** **CNPq** Patrocínio: **PETROBRAS AMBIENTAL** **IBR** **PETROBRAS** **IB** **IB** **IB**

Hoje novembro de 2009 Imprimir Semana Mês

seg	ter	qua	qui	sex	sáb	dom
26	27	28	29	30	31	
		07:50 4ª Aula Form.	19:00 Reunião grup			
2	3	4	5	6	7	
		07:00 4º Campo do	19:00 Reunião grup			

Figura 1.4 - Página de abertura do ambiente TelEduc (versão 3.3.8), com visualização da agenda.
 Fonte: Tela capturada pelo autor.

TELEDUC 1.1

Treinamento para Desenvolvedores Teleduc

Agenda - Agenda Atual

Agendas Anteriores

Título
 Agenda Parte 3 - Desenvolvimento de uma Ferramenta Teleduc

Texto

O objetivo desta parte do treinamento é desenvolver uma ferramenta Teleduc. Para tal, siga os seguintes passos:

1. Leia a documentação na web, <http://www.nied.unicamp.br/wiki/teleduc>, item **Documentação** - **Treinamento**
2. Desenvolva uma ferramenta. A descrição da ferramenta está na **Atividade Ferramenta Já Já Já Combat**
3. Monte a ferramenta funcionando para o seu formador.
4. Faça a **Atividade Documentação para Desenvolvedores** e poste as respostas no seu **Portfólio** individual.
5. Faça o **Atividade Avaliação Final** e poste as respostas no seu **Portfólio** individual.

Figura 1.5 - Página de abertura do ambiente Teleduc (versão 4.x), com visualização da agenda.
 Fonte: TELEDUC, 2018.

O Teleduc tem diversos recursos e cada disciplina é estruturada conforme a prática pedagógica usada, incluindo ou não ferramentas (BACKES et al., 2007).

As principais ferramentas são:

- Atividades: atividades propostas realizadas no decorrer do curso;
- Material de apoio: informações relacionadas à temática do curso;
- Leituras: publicação de artigos e referências de revistas, jornais, sites, etc;
- Mural: espaço para postagens diversas;
- Fóruns de Discussão: onde os participantes discutem sobre determinado assunto;
- Bate-papo: para conversa síncrona (tempo real);

Uma ferramenta bastante utilizada no TelEduc é o Intermap (FIGURA 1.6), que foi criado para auxiliar o participante a apreender a estrutura dos diálogos e o histórico das discussões, bem como as relações entre os participantes de um curso (ROCHA et al., 2002).

O mapa da interação, representado por meio de um grafo, é gerado a partir de uma *Applet Java* que pode ser executada em um navegador. O grafo gerado na Figura 1.6 está organizado na estrutura polar (circular), sendo que há também o grafo simples, por meio do qual se organiza as relações de forma mais dispersa, parecendo diagramas de árvore.

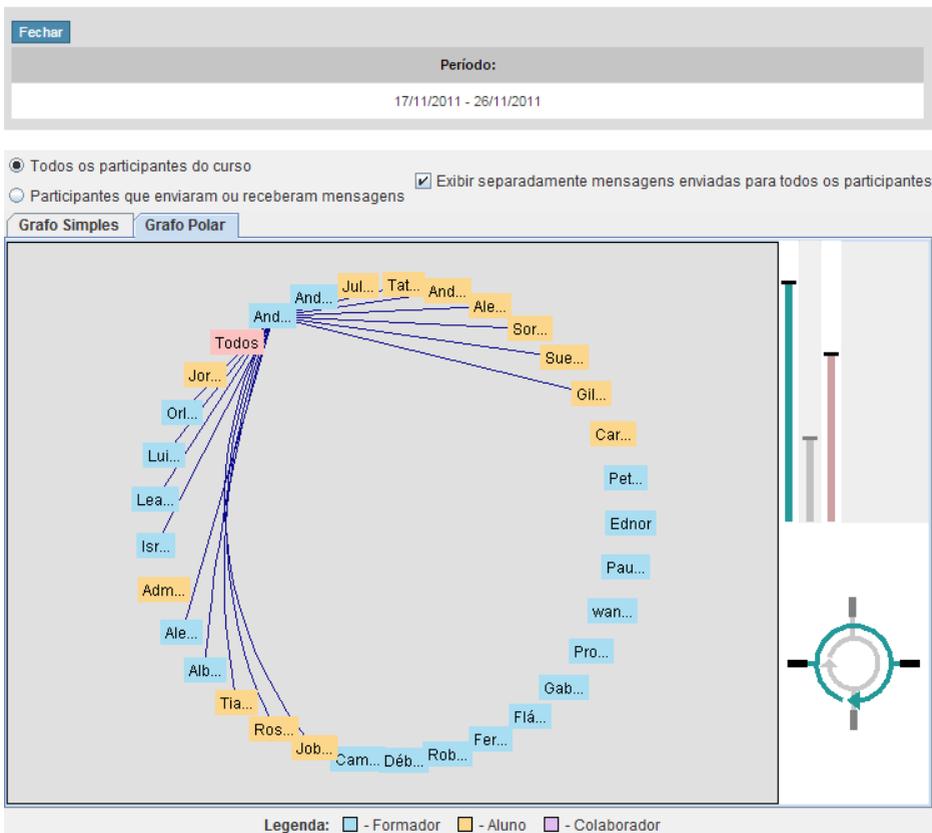


Figura 1.6 - Ferramenta da ferramenta Intermap no modo Polar.
 Fonte: TELEDUC, 2018.

Em 2002, quando o TelEduc foi atualizado e transformado em software livre, a Unicamp passou a adotá-lo como ambiente institucional para o ensino presencial e para o ensino à distância, sendo que em 2003, o Centro de Computação da Unicamp - (CCUEC) criou o projeto Ensino Aberto para universalizar o uso do TelEduc na instituição, vinculando este com o sistema

acadêmico, facilitando a geração de salas e a inscrição dos matriculados nas disciplinas nas salas do TelEduc.

Nesta época, outro ambiente que estava ganhando popularidade era o Moodle¹⁰, um *software* livre de apoio à aprendizagem, caracterizando-se como Sistema de Gerenciamento de Cursos – *Course Management System* (CMS), *Learning Management System* (LMS) ou um Ambiente Virtual de Aprendizagem (AVA).

Segundo Teodoro e Rocha (2007), o Moodle é um sistema de gerenciamento de cursos com finalidade de auxiliar educadores na construção e implementação de cursos em um ambiente virtual de aprendizagem.

Foi criado em 2001 pelo educador e cientista computacional Martin Dougiamas na Curtin University of Technology, em Perth, na Austrália, e tornou-se muito popular entre os educadores de todo o mundo.

Em julho de 2018 havia 232 países usando o sistema, com 102.135 registros de sites (servidores com a instalação), totalizando 14.830.097 cursos criados e 129.672.065 usuários cadastrados. Neste mesmo período, o país com maior número de registro era os EUA (9.790), ficando o Brasil em quarto com 5.190 registro de sites (MOODLE, 2018).

¹⁰Acrônimo de *Modular Object-Oriented Dynamic Learning Environment*.

O termo “orientado a objetos” em seu nome está, na verdade, relacionado à maneira como o sistema foi construído, pois trata-se de um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos (NAKAMURA, 2009).

O ambiente Moodle é distribuído gratuitamente, sob licença GNU-GPL¹¹ e foi escrito usando-se a linguagem PHP, podendo ser utilizados os seguintes bancos de dados: MySQL, PostgreSQL, Oracle, Access e Interbase (ou outra base com acesso via ODBC¹²).

Apresenta-se na Figura 1.7 uma página de um curso exemplo, para possibilitar uma visão geral do ambiente. Observa-se a área principal as abas do curso, onde se acessam as informações e atividades do curso. Na lateral esquerda, um menu retrátil, no qual pode-se navegar pelas semanas do curso ou nas funções de administração. Uma demonstração do funcionamento do Moodle pode ser acessada em <http://demo.moodle.net>.

11 GNU-GPL é um termo utilizado inicialmente por Richard Stallman, em 1984, para designar software livre. GNU é o nome do sistema operacional livre idealizado por Stallman, enquanto GPL (General License Public) é traduzido como Licença Pública Geral. Atualmente, a GPL é a licença com maior utilização em de projetos de software livre.

12 Acrônimo de *Open Database Connectivity*.

The image shows a Moodle course interface. On the left is a sidebar with a blue header 's crt_18_1' and a list of menu items: Participantes, Competências, Notas, 1ª Semana, 2ª Semana, 3ª Semana, 4ª Semana, 5ª Semana, Projeto Final, Painel, Página inicial do site, Calendário, Arquivos privados, Learner Dashboard, Instructor Dashboard, and Competency Dashboard. The main content area has a title 'Pensando em Códigos: Scratch 2018' and a breadcrumb 'Painel / Meus cursos / s crt_18_1'. Below the title is a 'Boas vindas' message and a progress indicator 'Seu progresso'. A navigation bar shows 'Boas vindas', '1ª Semana' (selected), '2ª Semana', '3ª Semana', '4ª Semana', '5ª Semana', and 'Projeto Final'. The '1ª Semana' section is titled '1ª Semana' and '02 a 8 de Abril - Scratch'. It features a Scratch logo and text explaining that Scratch programs are called scripts, created by dragging and dropping blocks. Below this is an 'Introdução' section with a list of activities: 'Leia e siga as orientações (1S)' (checked), 'AULA 01 - Apresentação do Scratch' (checked), '1.1 - Apostila de Scratch' (unchecked), and '1.2 Avaliação - Questionário sobre o vídeo 01' (unchecked). A 'Restrito' status is shown for 'AULA 01' with the text: 'Disponível se: A atividade AULA 01 - Apresentação do Scratch esteja marcada como concluída'.

Figura 1.7 - Página do ambiente Moodle, versão 3.5.1+, de um curso exemplo.

Fonte: Tela capturada pelos autores.

De acordo com Valente, Moreira e Dias (2009), o Moodle integra muitas das características esperadas de uma plataforma de e-learning, dentre elas:

- fóruns de discussão;
- gestão de conteúdos;
- criação de questionários;
- Chats;
- Wikis;

Dougiamas e Taylor (2003) examinaram seus trabalhos de ensino com o Moodle utilizando um curso à distância e o resultado provou ser muito bem-sucedido, focando elementos do ambiente de aprendizagem como:

- o site da Web como uma ferramenta para navegar o curso;
- o conteúdo do site da Web, atividades e recursos;
- a participação e apoio do tutor online;
- a participação e apoio dos estudantes.

Outra pesquisa feita por Dougiamas (2000) apresentou os resultados de um estudo com uma classe de professores do ensino médio aprendendo sobre a Internet, visando identificar pontos de melhoria para o Moodle, como uma ferramenta para construir e executar cursos na internet, obtendo resultados satisfatórios.

O Moodle apresenta-se como um projeto de apoio ao construtivismo social na educação, de acordo com Alves, Barros e Okada (2009), possibilitando a interação entre os participantes e permitindo a atuação da proposta pedagógica da utilização de AVAs, que leva em conta as experiências vividas dos usuários e seus conhecimentos na solução de problemas na medida em que se constrói o conhecimento.

Moodle é um pacote de software para a produção de cursos e web sites em internet. É um projeto de desenvolvimento contínuo concebido para apoiar a Filosofia do Moodle, dentro de um quadro construcionista social de educação. (MOODLE, 2012, online).

Baseando-se no trabalho de Valadares (2001), pode-se indicar algumas características dos ambientes construtivistas:

- Põem ênfase na construção do conhecimento e não na sua reprodução de memória, isto é, possibilita que o estudante atue ativamente na sua aprendizagem;
- Propiciam múltiplas representações na mesma realidade e não só uma. O Moodle e o TelEduc, por exemplo, possuem várias ferramentas distintas que permitem a comunicação entre os participantes, o envio de atividades, os ambientes de discussão como os fóruns e chats. Isto faz com que eles atendam diferentes estudantes, com maneiras distintas de construir seus conhecimentos, sendo que:
 - Encorajam a reflexão crítica constante dos estudantes durante suas atividades;
 - Privilegiam a avaliação formadora que deve estar voltada não só para a regulação da aprendizagem de cada estudante como também para a reflexão, auto-avaliação e autocorreção

da própria aprendizagem.

O Moodle possui várias opções de relatórios por meio dos quais os estudantes podem ver as atividades realizadas por eles, assim como o TelEduc por meio do intermap, onde o estudante visualiza graficamente sua interação nas atividades.

O construtivismo na sua forma mais trivial afirma que “o conhecimento é ativamente construído pelo estudante, não passivamente recebido do meio ambiente”. (VON GLASERSFELD, 1990).

Segundo Jonassen (1999), o conceito de construtivismo enfatiza o estudante como sendo o aprendiz ativo, desempenhando um papel central na mediação e controle de aprendizagem.

Para Vygotsky (1982), o aprendizado é resultante de um processo interativo ou coletivo e atua como promotora do desenvolvimento cognitivo do indivíduo.

Para a construção de ambientes educativos, aspectos como usabilidade, interação, afetividade, entre outros devem ser considerados. Projetos de softwares com perspectivas construtivistas, apoiam as práticas pedagógicas com relação à autonomia do sujeito e à forma de trabalhar o erro (COSTA, 2005).

Segundo Reategui (2007), a facilidade de

aprendizagem está relacionada aos mecanismos de orientação e navegação empregados na ferramenta; a eficiência de uso e frequência de erros estão relacionadas principalmente aos mecanismos de interação utilizados, enquanto a memorização pode estar vinculada à densidade de informações existentes nas telas (*layout/ design*).

Vale ressaltar que o fato de algumas plataformas possuírem uma abordagem construtivista não garante que toda atividade realizada por meio delas seja construtivista, sendo indispensável a formação e intervenção do professor que precisa ser um aprendiz e um investigador, a lutar por uma maior consciência dos ambientes e os participantes em uma situação de ensino dado, a fim de ajustar continuamente suas ações para envolver os estudantes na aprendizagem, utilizando o construtivismo como um referente (DOUGIAMAS, 1998).

6 Considerações finais

Buscou-se neste capítulo apresentar uma síntese do histórico do desenvolvimento das tecnologias da informação e comunicação e seus impactos na educação, ressaltando a relação cíclica entre a evolução nas linguagens de programação, o desenvolvimento de novas

tecnologias e o uso da informática na educação.

Quanto mais instrumentos tecnológicos para desenvolvimento de novas tecnologias, mais novos instrumentos são desenvolvidos, ressaltando que as mudanças sociais e históricas ocorrem de modo simultâneo, ou seja, com novas tecnologias digitais criam-se novas demandas, para novas ferramentas e assim por diante segue a evolução dessas.

Concomitante ao desenvolvimento dessas TIDCs, tem-se também alterações nos modos de comunicação e de construção de conhecimentos e, em muitos casos, acabam não sendo muito sincronizados. Isso quer dizer que, em muitas realidades, pode-se ter tecnologias digitais de ponta e metodologias ultrapassada, ou seja, não há uma relação determinística entre o uso de novos instrumentos com o uso de novas metodologias, apesar de ser essa a utopia educacional que esperaríamos.

Percebe-se com o histórico das linguagens de programação como estas impactaram na construção das ferramentas, criando novos instrumentos que facilitaram a criação dos recursos.

Nesse avanço tecnológico, pensando nos processos de ensino-aprendizado, avançou-se ao ponto dos próprios estudantes desenvolverem seus aplicativos

educacionais para dispositivos móveis ou computadores, usando linguagens lúdicas e simplificadas, baseando estas nas linguagens clássicas de programação, mas de um modo mais intuitivo e motivador para o estudante que não trabalha na área de informática.

Há muitos outros exemplos de como as tecnologias estão alterando os modos de se pensar e construir conhecimentos, assim como em como se relacionam e comunicam, esse será um processo que não terá fim dentro da história humana, constantemente evoluindo em um processo conectado das ciências, tecnologias e sociedades.

Referências

ABE, J. M.; SCALZITTI, A.; SILVA FILHO, J. I. *Introdução a lógica para a ciência da computação*. São Paulo: Editora Arte & Ciência, 2002.

AGUIAR, S. *Desatando os nós da rede*. Rio de Janeiro: Senac Nacional, 1997.

ALEXANDER, B. Web 2.0: A new wave of innovation for teaching and learning? *EDUCAUSE Review*, Louisville, v. 41, n. 2, p. 32–44, 2006.

ALVES, L.; BARROS, D.; OKADA, A. *Moodle: Estratégias Pedagógicas e Estudo de Caso*. Salvador: EDUNEB, 2009.

ANDRADE, P. F. *Projeto EDUCOM: realizações e produtos*. Brasília: MEC/OEA, 1993.

ANDRADE, P. F.; LIMA, M. C. M. A. *Projeto EDUCOM*. Brasília:

MEC/OEA, 1993.

ARAÚJO, J. P. Novas Tecnologias na Educação Especial: algumas considerações técnicas e pedagógicas. *Revista Conect@*, Rio de Janeiro, n. 4, 2002.

ARAYA, E. R. M.; VIDOTTI, S. A. B. G. *Criação, proteção e uso legal de informação em ambientes da World Wide Web [online]*. São Paulo: Editora UNESP; São Paulo: Cultura Acadêmica, 2010. 144 p. ISBN 978-85-7983-115-7. Available from SciELO Books <<http://books.scielo.org>>.

BACH, M. J. *The design of the Unix operating system*. New Jersey: Prentice-Hall, 1986.

BACKES, L.; MENEGOTTO, D. B.; SCHLEMMER, E. Ambiente virtual de aprendizagem: formação de comunidades virtuais? *Revista Filosofia Capital*, Rio Grande do Sul, v. 2, n. 4, p. 10-24, 2007.

BARRA, A. S. B.; MORAES, R. A. Proinfo, formação de professores e a informática em Goiânia. *Revista Educação Profissional: Ciência e Tecnologia*, Brasília: v. 1, n. 1, 2006.

BONILLA, M. H.; PRETTO, N. *Políticas brasileiras de educação e informática*. 2000. Disponível em: <<http://www.faced.ufba.br/~bonilla/politicas.htm>>. Acesso em: 02 out. 2011.

BRASIL. Ministério da Educação e Cultura - MEC. Referenciais para elaboração de material didático para ead no ensino profissional e tecnológico. 2007. Disponível em: <<https://pt.scribd.com/document/77111666/Referenciais-para-elaboracao-de-material-didatico-para-EaD-no-ensino-prof-e-tec>>. Acesso em: 06 set. 2018.

BROOKSHEAR, J. G. *Ciência da computação: uma visão abrangente*. 7. ed. Porto Alegre: Bookman, 2003.

BROWN, G. D. *COBOL: The failure that wasn't*. COBOLReport.com.

BRUGGER, R. *Web Based Course Environments: an Overview*. Universität Fribourg, 2000. Disponível em: <http://diuf.unifr.ch/people/brugger/papers/00_flashinfo/wbc-environmentsEN.html>. Acesso em: 01 nov. 2011.

BUSTAMANTE, S. B. V. Fundamentos, conceitos e importância do Núcleo de Informática Aplicada à Educação da UNICAMP na consolidação das atividades do núcleo de educação a distância da Universidade Católica de Petrópolis. *Tecnologias, Sociedade e conhecimento*, Campinas, v. 1, n. 1, p. 6-33, 2013.

CALANI, M. C. *Conceitos Geométricos através da linguagem Logo*. Campinas: Departamento de Ciência da Computação, IMECC, Unicamp, 1981.

CAMPOS, F. C. A.; ROCHA, A. R. C.; CAMPOS, G. H. B. Design Instrucional e Construtivismo: em Busca de Modelos para o Desenvolvimento de Software. In: CONGRESSO RIBIE, 4., Brasília, 1998. *Actas...* Brasília: UFRGS, 1994. p. 1-18. Disponível em: <http://www.ufrgs.br/niee/eventos/RIBIE/1998/pdf/com_pos_dem/250M.pdf>. Acesso em 05 set. 2018.

CASAS, L. A. A. et al. Construção de Conhecimento por Imersão em Ambientes de Realidade Virtual. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 7., 1996, Belo Horizonte, *Anais...* Belo Horizonte: PUCRS, 1996.

CHAVES, E. O. C. *Projeto Educom*. Campinas: NIED, 1983. Disponível em: <<http://www.chaves.com.br/TEXTSELF/EDTECH/EDUCOM.htm>>. Acesso em: 26 set. 2011.

CLOCKSIN, W. F.; MELLISH, C. S. *Programming in Prolog*. Editora Springer, 2003. p. 1.

CONSELHO NACIONAL DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO - (CNPQ). *Centro de memória: história do CNPq*. Disponível em: <<http://centrodememoria.cnpq.br/Missao2.html>>. Acesso em:

10 ago. 2018.

COCIAN, L. F. E. *Manual Da Linguagem C*. Canoas: Editora da ULBRA, 2004. p. 31–378

COELHO, F. C. *Computação Científica com Python*. Petrópolis: Editora Flávio Codeço Coelho, 2007.

CORRÊA, C. H. W. Comunidades Virtuais gerando identidades na sociedade em rede. *Revista eletrônica Ciberlegenda*, Niterói: n. 13, 2004.

COSTA, L. A. C.; FRANCO, S. R. K. Ambientes Virtuais De Aprendizagem e suas Possibilidades Construtivistas. *Revista Novas Tecnologias na Educação*, Porto Alegre: v. 3, n. 1, 2005.

COSTA, D. G. *Administração de redes com scripts: bash script, python e vbscript*. Rio de Janeiro: Editora Brasport, 2010. p. 23-24.

CYSNEIROS, P. G. *Professores e Máquinas: uma concepção de informática na educação*. Campinas/SP, Edutecnet, 1998. Disponível em: <http://edutec.net/Textos/Alia/proinfo/prf_txtie08.htm>. Acesso em: 05 nov. 2011.

CYSNEIROS, P. G. *Interação, tecnologias e Educação*. Recife: UFPE. 2010. Disponível em: <<http://www.diaadia.pr.gov.br/nre/ibaiti/arquivos/File/Cysneiros.pdf>>. Acesso em: 07 nov. 2011.

D'ÁVILA, C. M. *Pedagogia Cooperativa e Educação a Distância: uma aliança possível*. *Educação & Contemporaneidade*, São Paulo, v. 12, n. 20, p. 273-297, 2003.

MATTOS, E.T. *Programação de Softwares em Java*. São Paulo: Digerati Books, 2007

DOUGIAMAS, M. *A journey into Constructivism*. 1998. Disponível em:

<<http://dougiamas.com/writing/constructivism.html>>. Acesso em: 17 out. 2011.

DOUGIAMAS, M. *Improving the effectiveness of tools for Internet based education*. Proceedings of the 9th Annual Teaching Learning Forum, 2000. Disponível em: <<https://dougiamas.com/writing/tlf2000/>>. Acesso em: 10 ago. 2018.

DOUGIAMAS, M.; TAYLOR, P. *Moodle: usando comunidades de aprendizes para criar um sistema de fonte aberta de gerenciamento de curso*. Papel arbitrado, apresentado no EDMEDIA 2003. Disponível em: <<http://dougiamas.com/writing/edmedia2003/>>. Acesso em: 25 out. 2011.

ESTROZI, L. F.; BATISTA NETO, J. E. S.; MARTIN, O. *Programando para a Internet com PHP*. São Paulo: Editora Brasport, 2010.

FERRANTE, A. J.; RODRIGUEZ, M. V. R. *Tecnologia de informação e gestão empresarial*. Rio de Janeiro: Editora E-papers, 2000.

FERREIRA, A.; VIEIRA, J. A moda dos blogs e sua influência na cibercultura: do diário virtual aos posts comerciais. *Revista da Associação Nacional dos Programas de Pós-Graduação em Comunicação*, Belo Horizonte: p. 1-14, jun. 2008. Disponível em: <<http://www.compos.org.br/seer/index.php/e-compos/article/viewFile/205/206>>. Acesso em: 15 out. 2011.

FIGUEIRA, J. S. Easy Java simulations - modelagem computacional para o ensino de Física. *Revista Brasileira de Ensino de Física*, São Paulo, v. 27, n.4, 2005.

FRANCO M. A.; CORDEIRO, L. M.; CASTILLO, R. A. F. *O ambiente virtual de aprendizagem e sua incorporação na Unicamp*. [online]: Educação e Pesquisa, 2003. Disponível em: <<http://dx.doi.org/10.1590>>. Acesso em 05 set. 2018.

FRANCISCO, A. *75.31 Teoría de Lenguajes Haskell* λ . Facultad

de Ingeniería, Universidad de Buenos Aires, Buenos Aires, 2006. Disponível em: <<http://web.fi.uba.ar/~falbani/7531/Haskell.pdf>>. Acesso em: 09 out. 2011.

GREGOLIN, V. R. LOGO: explorando conceitos matemáticos. *Revista Tecnologias da Educação*, Belo Horizonte, v. 1, n. 1, 2008.

HAGUENAUER, C. J.; LOPEZ, F. B.; MARTINS, F. N. Estudo Comparativo de Ambientes Virtuais de Aprendizagem. Colabor@ - *Revista Digital da CVA - Ricesu*, Curitiba, v. 2, n. 5, 2003.

HARASIM, L. *On-Line Education: A New Domain*. International Centre for Distance Learning, London: The Open University, Oxford: Pergamon Press, 1998. Disponível em <<http://www.bdp.it/rete/im/harasim1.htm>>. Acesso em: 12 nov. 2011.

HORNINK, G. G. *Cartografando online: caminhos da informática na escola com professores que elaboram conhecimentos em formação contínua*. 2010. 296f. Tese (Doutorado em Ciências) - Instituto de Geociências, Universidade Estadual de Campinas (Unicamp), Campinas/SP, 2010. Disponível em: <<http://repositorio.unicamp.br/jspui/handle/REPOSIP/287024>>. Acesso em: 06 set. 2018.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - (IBGE). *PNAD Contínua TIC 2016: 94,2% das pessoas que utilizaram a Internet o fizeram para trocar mensagens*. [Agência IBGE]: Rio de Janeiro/RJ. 2018. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2013-agencia-de-noticias/releases/20073-pnad-continua-tic-2016-94-2-das-pessoas-que-utilizaram-a-internet-o-fizeram-para-trocar-mensagens.html>>. Acesso em: 06 set. 2018.

INSTITUTO BRASILEIRO DE OPINIÃO PÚBLICA E ESTATÍSTICA

- (IBOPE), Nielsen Online. Total de pessoas com acesso à internet atinge 77,8 milhões. 2011. Disponível em: <http://www.ibope.com.br/pt-br/noticias/Paginas/Total%20de%20pessoas%20com%20acesso%20%C3%A0%20internet%20atinge%2077,8%20milh%C3%B5es.aspx>>. Acesso em: 06 set. 2018.

ISOTANI, S. et al. Web 3.0 - Os Rumos da Web Semântica e da Web 2.0 nos Ambientes Educacionais. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 19., 2008, Fortaleza. *Anais ...* Fortaleza: SBC, 2008. p. 785 - 795. Disponível em: <<http://dx.doi.org/10.5753/cbie.sbie.2008.785-795>>. Acesso em 06 set. 2018.

JONASSEN, D. Designing constructivist learning environments. In C. Ed. *Reigeluth, Instructional design theories and models: a new paradigm of instructional theory*, vol. II. Mahwah, NJ: Lawrence Erlbaum Associates, 1999. p. 215-239

KREPSKY, W. *Protótipo de um interpretador para um ambiente de programação lógica*. 1999. 70 f. Trabalho de Conclusão de Curso (Bacharelado) - Universidade Regional de Blumenau Centro de Ciências Exatas e Naturais Curso de Ciências da Computação, Blumenau-SC, 1999 Disponível em: <<http://campeche.inf.furb.br/tccs/1999-I/1999-1wendykrepskyvf.pdf>>. Acesso em: 11 out. 2011.

LEITE, M. *Técnicas de Programação - Uma Abordagem Moderna*. São Paulo: Brasport, 2006. p. 182.

LÉVY, P. *Cibercultura*. 2 ed. São Paulo: Ed. 34, 2000.

LORENZI, F.; LOPES, A. V. *Linguagem de Programação Pascal*. Canoas: Editora ULBRA, 2000, p. 07.

MARCELINO, G. F. Avaliação de políticas públicas: os resultados da avaliação do ProInfo. In. CONGRESO INTERNACIONAL DEL CLAD SOBRE LA REFORMA DEL ESTADO Y DE LA ADMINISTRACIÓN PÚBLICA, 8., 2003, Panamá. *Anais ...* Panamá: CLAD, 2003. p. 28-31.

- MATTAR, J. Web 2.0 e redes sociais na educação a distância: Cases no Brasil. *Revista La Educ@cion, Local*, n. 145, 2011.
- MAUERER, W. *Professional Linux Kernel Architecture*. Local: John Wiley & Sons, 2008.p. 1175.
- MOODLE. Moodle stats. Moodle.org, 2008. Disponível em: <<https://moodle.net/stats/>>. Acesso em: 06 set. 2018.
- MOODLE. Sobre o Moodle. Moodle.org, 2012. Disponível em: <>. Acesso em: 06 set. 2018.
- MOORE, M. G. Theory of transaction distance. In: KEEGAN, Ed. *Desmond, Theoretical principles of distance education*. London: Routledge, 1993.
- MORAES, M. C. Informática Educativa no Brasil: uma história vivida, algumas lições aprendidas. *Revista Brasileira de Informática na Educação*, Porto Alegre, n. 1, p.19-44, 1997
- MORAZ, E. *Treinamento prático em PHP*. Universo dos Livros Editora LTDA, 2005. p. 14.
- NAKAMURA, R. *Moodle: como criar um curso usando a plataforma de Ensino a Distância*. São Paulo: Farol do Forte, 2009.
- O'REILLY, T. *Web 2.0 Compact Definition: Trying Again*. 2006. Disponível em: <<http://radar.oreilly.com/2006/12/web-20-compact-definition-tryi.html>>. Acesso em 10 out. 2011.
- PALAZZO, L. A. M. *Introdução à Programação Prolog*. Editora da Universidade Católica de Pelotas/UCPEL – Pelotas, p. 06, 1997.
- PEREIRA, I. O movimento do software livre. In: VIII CONGRESSO AFRO-BRASILEIRO DE CIÊNCIAS SOCIAIS, 8., 2004, Coimbra. *Anais...* Coimbra: CES, 2004. p. 1-17. Disponível <<https://www.ces.uc.pt/lab2004/pdfs/InesPereira.pdf>>. Acesso: 06 set. 2018. p. 83.

PEREIRA, A. A. #eadsunday: educação a distância em discussão aos domingos (estudo de caso de uma experiência bem sucedida no Twitter). São Paulo: Universidade de São Paulo, 2010. Disponível em: <http://issuu.com/antoniacj/docs/monografia_eadsunday_final>. Acesso em: 09 nov. 2011.

POLLONI, E. G. F.; Fedeli, R. D. *Introdução à Ciência da Computação*. Cengage Learning Editores, 1999. p. 62–66

REATEGUI, E. Interfaces para Softwares Educativos. In: CICLO DE PALESTRAS NOVAS TECNOLOGIAS NA EDUCAÇÃO, 9., 2007, Porto Alegre. *Tópico temático...* Porto Alegre: CINTED, 2007. p. 01-10. Disponível em: <<http://www.cinted.ufrgs.br/ciclo9/artigos/1bEliseo.pdf>>. Acesso em: 03 nov. 2011.

ROCHA, H. V. et al. Projeto TelEduc: Pesquisa e Desenvolvimento de Tecnologia para Educação a Distância. In: CONGRESSO INTERNACIONAL DE EDUCAÇÃO A DISTÂNCIA DA ABED, 9., 2002, São Paulo. Anais... São Paulo: ABED, 2002. p. 01-72. Disponível em: <http://www.teleduc.org.br/sites/default/files/publications/pr_emio_abed2002.pdf>. Acesso em 06 set. 2018.

SEBESTA, R. W. *Conceitos de linguagem de programação*. Porto Alegre : Editora Bookman, 2002.p. 36,

SILVA JÚNIOR, A. A.; HORNINK, G. G. Desenvolvendo o raciocínio lógico matemático a partir da aprendizagem de princípios de programação em blocos. In: ENCONTRO VIRTUAL DE DOCUMENTAÇÃO EM SOFTWARE LIVRE E CONGRESSO INTERNACIONAL DE LINGUAGEM E TECNOLOGIA ONLINE. 2017. [S.l.]. *Anais ... Texto Livre*, [S.l.] v. 6, n. 1, jun. 2017. ISSN 2317-0239. Disponível em: <http://www.periodicos.letras.ufmg.br/index.php/anais_linguagem_tecnologia/article/view/12127>. Acesso em: 18 jul. 2018.

SILVEIRA, S. A. Twitter na escola ajuda? *Revista ARede*. São Paulo. n. 51, 2009. Disponível em:

<<http://www.arede.inf.br/inclusao/edicoes-anteriores/150-edicao-no51-setembro2009/2268-twitter-na-escola-ajuda>>. Acesso em 15 out. 2011.

SIMÕES, J. A. A.; LINDEMANN, V. Editor de Textos Colaborativo Integrado ao Teleduc. *RENOTE - Revista Novas Tecnologias na Educação*, Porto Alegre. v. 5, n. 1, 2007. Disponível em: <<http://seer.ufrgs.br/renote/article/view/14215>>. Acesso em 06 set/ 2018.

SOUZA, P. S. Publicação de revistas científicas na Internet. *Revista Brasileira de Cirurgia Cardiovascular*, São José do Rio Preto, v. 21, n. 1, p. 24-28, Mar. 2006. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-76382006000100006&lng=en&nrm=iso>. Acesso em 18 jul. 2018.

TELEDUC, 2018 <http://www.teleduc.org.br>

TEODORO, G. L. M.; ROCHA, L. C. D. *Moodle - Manual do Professor*. Belo Horizonte: UFMG, 2007.

TIOBE, Software BV. TIOBE Programming Community Index for July 2018. Disponível em: <<https://www.tiobe.com/tiobe-index>>. Acesso em: 19 jul. 2018.

UOL. Bate-papo UOL bate recorde histórico de audiência no carnaval. *Uol Entretenimento*, São Paulo, Fev. 2018. Disponível em: <<https://entretenimento.uol.com.br/noticias/redacao/2018/02/20/bate-papo-uol-bate-recorde-historico-de-audiencia-no-carnaval.htm>>. Acesso em 18 jun. 2018.

VALADARES, J. Estratégias Construtivistas e Investigativas no Ensino das Ciências. Conferência proferida no Encontro «O ensino de Ciências no âmbito dos Novos Programas», Universidade do Porto, Porto, 2001.

VALENTE, J. A. *Diferentes usos do Computador na Educação*.

Em Aberto, Brasília, v. 12, n. 57, 1993. Disponível em: <http://emaberto.inep.gov.br/index.php/emaberto/article/view/1876>>. Acesso em: 06 set. 2018. p.85

VALENTE, J. A. *O Computador na Sociedade do Conhecimento*. Campinas: UNICAMP/NIED, 1999. Disponível em: <<https://www.nied.unicamp.br/biblioteca/o-computador-na-sociedade-do-conhecimento>>. Acesso: 06 set. 2018.

PRADO, M. E. B. B.; VALENTE, J. A. A formação na ação do Professor: uma abordagem na e para uma nova prática pedagógica. In: VALENTE, J. A. (org.) *Formação de educadores para uso da informática no ensino*. Campinas: NIED-Unicamp, 2003.

VALENTE, J. A.; ALMEIDA, F. J. Visão analítica da informática na educação no Brasil: a questão da formação do professor. *Revista Brasileira de Informática na Educação*. Porto Alegre, n. 1, 1997. p. 1-28

VALENTE, L.; MOREIRA, P.; DIAS, P. Moodle: moda, mania ou inovação na formação? In: CONFERÊNCIA INTERNACIONAL DE TECNOLOGIAS DE INFORMAÇÃO E COMUNICAÇÃO NA EDUCAÇÃO, 5., 2009, Braga. *Anais ... Braga: Universidade do Minho*, 2009. p. 35-54. Disponível em <http://www.valente.org.pt/downloads/artigos/moodle_estrategias_ped.pdf>. Acesso em 06 set. 2018.

VELLOSO, F. C. *Informática: conceitos básicos*. 10. ed. Rio de Janeiro: Elsevier, 2017.

VIANA, L. C. A. et al. Proposta de uma Metodologia de Avaliação de Ambientes Educacionais a Distância baseado em Concepções Pedagógicas. In: CONGRESSO DE COMPUTAÇÃO DO SUL DE MATO GROSSO, 1, 2005 Rondonópolis, *Anais...* Rondonópolis: UFMT, 2005. p. 235 - 250. Disponível em: <<http://nlx.di.fc.ul.pt/~guelpeli/Arquivos/Artigo10.pdf>>. Acesso em 06 set. 2018.

VON GLASERSFELD, E. An exposition of constructivism: Why some like it radical. In: Davis R. B.; Maher C. A.; Noddings N.

(eds.) Monographs of the Journal for Research in Mathematics Education, 4, Reston: National Council of Teachers of Mathematics, v.4, p. 19–29, 1990. Disponível em: <<http://www.vonglasersfeld.com/127>>. Acesso em: 06 set. 2018.

VYGOTSKY, L. S. *Obras Escogidas II (Pensamiento Y Lenguaje)*. Moscú: Editorial Pedagógica, 1982.

WALKER, D. *C++ Programming*. [S.I]: Global Media Media. 2008.

XAVIER, G. F. C. *Lógica de Programação*. São Paulo: Editora Senac, 2007. p. 21

Capítulo 2 DigiAtlas - Construção e avaliação de aplicativos para auxílio em trabalhos de campo

Lucas Faloni Ferreira
Paulo Alexandre Bressan
Gabriel Gerber Hornink

A problemática ambiental faz parte de nosso dia a dia e pode fazer parte, de modo integrado, do ensino, em uma visão multi/ interdisciplinar. Os trabalhos de campo são importantes para abordar tal temática e, para tanto, desenvolveu-se o aplicativo DigiAtlas Mobile, um aplicativo para a plataforma Android que visa auxiliar o ensino-aprendizagem, sendo possível ser utilizada em campo ou em sala de aula para o gerenciamento de dados ambientais que são apresentados em um mapa e realização de atividades didáticas. Os dados iniciais disponíveis no DigiAtlas são provenientes de pesquisas realizadas por professores do Ecofrag/ Unifal-MG, abordando a região de Alfenas e arredores. Além destes, é possível a inclusão de novos dados que poderão ser utilizados localmente e, se enviados para o banco de dados, poderão ser disponibilizados a todos usuários em futuras atualizações após serem verificados por pesquisadores. Para a verificação de novos dados foi desenvolvida a aplicação DigiAtlas Web. Como uma ferramenta de ensino-aprendizagem, é crucial que o DigiAtlas seja avaliado quanto a sua usabilidade, visando que seja fácil e não dificulte o aprendizado e o ensino. Para isso, foram aplicadas avaliações de usabilidade no *software* sendo a primeira realizada com professores da área e uma segunda avaliação com estudantes de graduação. O uso do DigiAtlas levará para escolas dados ambientais locais, com a possibilidade de gestão e inclusão de dados pelos próprios estudantes, potencializando o ensino da temática ambiental.

1 Introdução

A temática ambiental é apresentada constantemente em sala aula devido a importância em saber de nossa existência como parte de um sistema natural, e assim, como manipuladores e exploradores, temos a responsabilidade de preservar e manter o equilíbrio nesse sistema. Essa temática envolve várias disciplinas em seu conteúdo como Biologia, Geografia, Química, Física, sociologia, entre outras, sendo que cada uma contribui para um olhar diferente a esse assunto de forma integrada.

A conscientização ambiental é de grande importância, contudo, muitas vezes, é difícil chamar a atenção e trabalhar esses conceitos com os estudantes, estando fora do espaço real, em sala de aula. Dessa forma o desenvolvimento de instrumentos e métodos dinâmicos que unifiquem tecnologia e educação é uma opção para criar um maior interesse e melhorar a compreensão por parte dos educandos.

Nesse contexto, surgiu um aplicativo gerenciador de dados ambientais para dispositivos móveis, o DigiAtlas¹³ (FERREIRA, et al., 2015), com sua segunda

¹³Website: <http://www.unifal-mg.edu.br/lme/projetos/digiatlas>

versão desenvolvida por Silva et al., (2018).

O DigiAtlas foi desenvolvido para dispositivos móveis, plataforma Android, disponibilizado no *Google Play*¹⁴, visando dar mobilidade aos pesquisadores no campo e ainda despertar interesse nos estudantes, que geralmente têm muito contato com essas de tecnologias.

O principal modo de interação é por meio de um mapa digital que apresenta informações ambientais georreferenciadas. Além da descrição de cada informação, estas podem conter áudios, imagens e vídeos de fácil acesso, deixando a aplicação mais interessante. Com isso, o DigiAtlas pode ser utilizado como auxílio para o ensino-aprendizado em atividades fora ou dentro de sala, quebrando a rotina e trazendo maior interesse do estudante.

O DigiAtlas tem basicamente dois objetivos: O primeiro deles é ser uma ferramenta de ensino-aprendizagem, auxiliando o professor em atividades dinâmicas em campo ou em sala de aula, buscando criar maior interesse dos estudantes no estudo da temática ambiental, e assim melhorar a consciência da sua e de outras influências no meio em que habita, e a necessidade de preservá-lo; o segundo objetivo é ser

14 <https://play.google.com/store/apps/details?id=com.digi atlas>

uma ferramenta para auxílio em pesquisas em laboratório ou no campo, com a padronização da forma como os dados são estruturados, unificação estes em um único local e o compartilhamento de dados individuais.

Além da versão Android para dispositivos móveis, construiu-se também a versão Web¹⁵ (SILVA, 2018), na qual pode-se visualizar os dados, além disso, validar dados na base para inserção no aplicativo.

A cada nova pesquisa, dados sobre desmatamentos sempre são alarmantes devido, historicamente, à forma exploratória da colonização, as características atuais de manejo do solo pela agricultura que não utilizam práticas conservacionistas, a degradação das áreas de preservação permanente e reserva legal, entre outros motivos. Ainda hoje, a Mata Atlântica tem perdido extensas áreas, com aproximadamente 13% de remanescentes florestais (ATLÂNTICA; INPE, 2014).

O estado de Minas Gerais se encontra como primeiro colocado no *ranking* de áreas florestais desmatadas pelo quinto ano consecutivo, apresentando atualmente somente 10,4% de área de vegetação nativa. No período de 2012 a 2013 foram desmatados 8.437 hectares de Mata Atlântica somente neste estado,

15Website: <http://www.desenvolvimentovirtual.com/digiatlas>

que apesar de diminuir na proporção desmatada, não cessa essa prática de alto impacto ambiental (ATLÂNTICA; INPE, 2014).

O DigiAtlas possui uma abordagem multidisciplinar, que é essencial para a junção dos conhecimentos gerados pelas disciplinas e o entendimento dos mesmos, o que muitas vezes é complexo e difícil. Tal abordagem pode trazer o interesse dos estudantes nas multidisciplinas, permitindo esse olhar multifocal e dinâmico à aprendizagem e ao ensino (DUSO; BORGES, 2010).

Nesse sentido, soma-se ao uso do DigiAtlas o fato de que as atividades de campo acabam por viabilizar ou mesmo potencializar a visão multidisciplinar, fazendo com que os estudantes saia da rotina da sala de aula, estimulando a problematização de situações nas quais o estudante está inserido, trazendo a realidade deste para o contexto escolar, viabilizando o desenvolvimento da criatividade do estudante e professor no decorrer das atividades (PINTO, 2003).

Com relação aos trabalhos de campo tradicionais, tem-se a característica na forma como dados são coletados em visitas ao campo pelos pesquisadores. Nessas visitas, os dados são anotados de acordo com a

preferência do pesquisador, sem uma padronização, o que dificulta a junção das informações coletadas de pesquisadores distintos, ou muitas vezes confunde o próprio pesquisador ao olhar as anotações tempos depois. Assim, o DigiAtlas padroniza a forma de realizar essa coleta/anotações de dados em visitas ao campo, e possibilita o compartilhamento dessas anotações entre os usuários da aplicação.

Devido o DigiAtlas ser uma ferramenta de ensino-aprendizagem, há também a preocupação quanto a sua usabilidade visto que se ele for complexo ou difícil de usar, pode tornar-se desestimulante para o estudante e professor, interferindo assim no aprendizado por parte do estudante e no ensino pelo professor.

Nesse sentido, diversas questões técnicas e didáticas foram demandadas para seu desenvolvimento, sendo sua construção realizada em algumas etapas (levantamento de requisitos, desenvolvimento no Android Studio e avaliação da usabilidade) que serão apresentadas a seguir, fornecendo importantes informações para compreensão do DigiAtlas, assim como para auxiliar desenvolvedores de sistemas educacionais similares.

2 Dispositivos móveis no ensino

Algumas aplicações semelhantes ao DigiAtlas Mobile foram encontradas, como o MySoil¹⁶ e o Geomóvel. Dentre essas aplicações, o Geomóvel (MARÇAL, et al., 2013) mais se aproxima, pois, é um aplicativo para auxílio a aulas de campo de geologia, que se assemelha a possibilidade de utilização do DigiAtlas para aulas de campo com foco na temática ambiental.

A seguir tem-se a descrição dos dois aplicativos, assim como também um tópico sobre o uso de dispositivos móveis no ensino.

2.1 Aplicativos pré-existent

Desenvolvido pela *British Geological Survey* o aplicativo MySoil está disponível para as plataformas iOS e Android. O MySoil dá acesso às propriedades do solo europeu como profundidade, textura, pH, temperatura, teor de matéria orgânica e habitats dominantes, possuindo dados mais detalhados do Reino Unido.

É dedicado para pessoas com interesse no solo na Europa, incluindo jardineiros e produtores de hortaliças, agricultores e técnicos agrícolas, escolas e faculdades, ambientalistas e planejadores de uso do solo.

¹⁶<https://play.google.com/store/apps/details?id=uk.ac.bgs.mysoil&hl=pt>

MySoil combina informações sobre o solo a partir do *British Geological Survey*, o Centro de Ecologia e Hidrologia, o Centro Comum de Investigação da União Europeia (*EU Joint Research Centre (JRC)*) e do *Met Office*.

O aplicativo tem a navegação por meio de um mapa podendo utilizar-se do GPS para acessar sua localização e descobrir os detalhes do solo nessa localização.

Oferece a opção para envio de uma descrição ou análise de qualquer solo, incluindo as fotos, ajudando a comunidade com interesse nessas informações. Contudo, como dito, os dados mais detalhados são da região do Reino Unido, e o restante apenas da Europa.

O Geomóvel (MARÇAL, et al., 2013) é resultado de um projeto realizado por pesquisadores da Universidade Federal do Ceará (UFC) e tem como principal objetivo auxiliar nas aulas de campo de Geologia. As atividades de campo que foram escolhidas para serem reproduzidas com a ajuda do aplicativo foram: medição de rota e ângulo do mergulho, coleta de coordenadas geográficas, fotografias e anotações (via texto e/ou áudio).

Em atividades de campo realizadas por estudantes de Graduação em Geologia, a todo momento é

necessário realizar anotações. Tais anotações futuramente serão a base para um relatório que deve ser entregue para consolidar as informações coletadas. No Geomóvel é possível agregar as anotações aos respectivos dados coletados e exportar, consolidando assim a atividade de campo.

Todas as informações coletadas são associadas a coordenadas geográficas por meio do GPS. Já o acelerômetro e manômetro, assim como o GPS disponíveis em *smartphones* e *tablets*, são utilizados para medição de ângulo e de rumo de mergulho, simulando uma bússola.

Outros recursos presente nos dispositivos móveis que também são utilizados pelo Geomóvel são microfone e câmera, que servem para a captura de áudio e fotos respectivamente como anotações.

A exportação resulta em um arquivo no formato padrão KMZ, que é muito utilizado e pode ser lido por meio de ferramentas gratuitas como *Google Earth*, *GPS Map Viewer* e *FileViewPro*.

2.2 Uso de dispositivos móveis no ensino

Nos últimos anos a disseminação dos dispositivos móveis pelo mundo é massiva, principalmente entre os

jovens, adolescentes e crianças que se encantam com maior facilidade pelos recursos disponíveis.

Estes dispositivos permitem acesso a informação em qualquer local, inclusive em ambiente escolar. Mesmo que exista aplicativos que desviem a atenção do estudante, diminuindo o interesse no aprendizado, os recursos disponíveis dos dispositivos móveis (áudio, foto, vídeo, localizador geográfico, dentre outros), possibilitam um novo meio de elaborar atividades didáticas em sala de aula.

Com isso, os professores podem usar aplicativos pré-instalados no próprio dispositivo, ou por meio de aplicativos desenvolvidos especificamente para a abordagem de assuntos de uma ou mais disciplinas, utilizando os recursos disponíveis nos dispositivos móveis.

Segundo a empresa Distimo (2013), que faz análise de lojas de aplicativos, em setembro de 2013, na loja virtual para aplicativos Android, *Google Play*, apenas 2% eram relacionados com educação. Além disso, muitos aplicativos educacionais, abordam o conteúdo de forma superficial e sem preocupações pedagógicas, ou ainda, não abordam os assuntos específicos da grade curricular (FERREIRA et al., 2015).

De acordo com Ferreira et al., (2015, p. 695)

[...] o desenvolvimento de aplicativos que exerçam papel eficiente na educação e/ou na transferência de conhecimento pode favorecer o processo ensino-aprendizagem. Segundo Ferreira e Tomé (2010), os estudantes tiveram maior interesse e facilidade com os currículos escolares, e assim alcançam maior fixação do conteúdo.

Entretanto, o desenvolvimento de aplicativos educacionais não é algo exato, deve-se levar em conta o conteúdo a ser abordado e as atividades que podem ser desenvolvidas com o auxílio do aplicativo. Mesmo que o aplicativo seja feito atentando-se a todos os detalhes, de nada adiantará se o mesmo não for bem utilizado, é o professor quem pensará na sequência didática. O aplicativo é uma nova ferramenta para a abordagem do mesmo assunto, assim ainda existe todo o esforço de preparar as aulas para que o ensino seja efetivo.

Também a familiaridade e criatividade do professor ao usar a aplicação pode ser um ponto chave no bom aproveitamento do aplicativo para o ensino-aprendizado. Por fim, a utilização e aplicação por meio da tecnologia possibilita trabalhos fora da sala de aula que é algo que pode criar maior interesse do estudante, além disso, o ensino multidisciplinar torna a abordagem da temática ambiental mais atraente (FERREIRA et al., 2015).

3 Desenvolvimento - elementos principais

Para a construção das aplicações, definiram-se requisitos em reuniões periódicas com os professores envolvidos no projeto, nas quais se discutiram os principais elementos desenvolvidos no DigiAtlas.

No desenvolvimento, quando necessário, fez-se estudos para a utilização de recursos disponíveis nos dispositivos móveis, e em seguida, implementaram-se os elementos.

Durante a avaliação de usabilidade, avaliaram-se os elementos essenciais do ponto de vista técnico da arquitetura da informação e do *design*. Refez-se o levantamento de requisitos, desenvolvimento e avaliação de usabilidade diversas vezes a cada acréscimo de novas funcionalidades na ferramenta ou mesmo quando houve mudanças em algum requisito pronto.

No tópico a seguir é descrito a origem e a padronização dos dados presentes na aplicação. Em seguida tem-se o método de avaliação do aplicativo. Também nesta seção, serão colocadas as plataformas utilizadas para a construção do DigiAtlas, tanto Mobile (aplicativo), quanto Web (site).

É discutida a plataforma Android, utilização da API *Google Maps*, a metodologia Scrum para

desenvolvimento ágil de software e a persistência de dados com SQLite em aplicações Android.

Após isto, são descritos o *framework Codeigniter* para auxílio da programação web, o qual utiliza da linguagem PHP, CSS e a Linguagem de Marcação de Hiper Texto (HTML). Também é apresentado o padrão MVC utilizado pelo *Codeigniter*.

O banco de dados utilizado foi o mesmo (SQLite), a fim de facilitar a integração das duas aplicações.

3.1 Dados ambientais

Os dados iniciais do DigiAtlas são provenientes dos professores de Biologia, Ecologia e Geologia da Universidade Federal de Alfenas, principalmente os envolvidos no Laboratório de Ecologia de Fragmentos Florestais (Ecofrag), oriundos de coletas realizadas na região do sul de Minas Gerais, entorno de Alfenas-MG.

Tais dados foram organizados em uma tabela com os seguintes atributos: Fonte que coletou e disponibilizou o dado; Latitude/Longitude que especifica a localização; Local em que foi realizada a coleta; Tipo de dado (Aves, Mamíferos, Invertebrados, Anfíbios, Plantas e Abióticos); Família, Gênero e Espécie; Nome Popular (se existir); Ocorrência especificando o número indivíduos

visualizados no local; Data da coleta; Cidade em que foi realizada a coleta; Comentário ou alguma informação extra sobre o dado (FERREIRA et al., 2015).

Como descrito, o atributo tipo é limitado a seis categorias e alguns atributos não são existentes em todos os tipos de dados, por exemplo, em dados abióticos que especificam regiões com assoreamentos ou depósitos de lixos, não existe o atributo família.

Grande parte dos dados possuem áudios e/ou imagens, sendo imagens a grande maioria. Muitos dados possuem mais de uma imagem. Ao total são cerca de 580 imagens e 120 áudios das mais variadas espécies.

3.2 Avaliação

O DigiAtlas, como ferramenta de ensino-aprendizagem, deve ser avaliado quanto a sua usabilidade, para que seu uso seja estimulante, intuitivo e realmente ajude seus usuários. Realizaram-se as avaliações seguindo os princípios de usabilidade de Nielsen e Budiu (2014).

Na ISO 9241-11, usabilidade é definida como:

“uma medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso”
(ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002, p. 5)

De acordo com Nielsen e Budiu (2014), um teste de usabilidade é bem aproveitado se realizado na forma qualitativa e não estatística. A ideia dos autores não é excluir a avaliação quantitativa, pois esta também é importante, mas indicar que quando uma dificuldade é encontrada, que também pode vir a ser uma dificuldade para a maioria dos usuários, deve-se sanar essa dificuldade sem se preocupar em saber qual o número exato de usuários do teste que tiveram a mesma dificuldade.

Utilizou-se na avaliação o método do percurso cognitivo para a inspeção de usabilidade, com elementos de *checklist*. Esse método baseia-se em deixar que o usuário explore a interface computacional a ser avaliada, e aprenda sem a necessidade de treinamento ou consultas a manuais. Isso, devido a estudos apresentarem que a maioria dos usuários prefere aprender dessa forma. Assim, avalia-se o quão fácil foi para o usuário aprender a utilizar a aplicação explorando a interface da aplicação (DIX et al., 2004; LEWS; WHARTON, 1997).

3.3 Desenvolvimento do aplicativo

3.3.1 Android

Android é um sistema operacional projetado pela empresa *Google*, para dispositivos móveis (*tablets* e *smartphones*), sendo o sistema operacional móvel mais utilizado no mundo. Em 2015 estava presente em pouco mais de 50% de todos aparelhos comprados no mundo (NET MARKET SHARE, 2015), o que já era um ponto forte para o uso desse sistema.

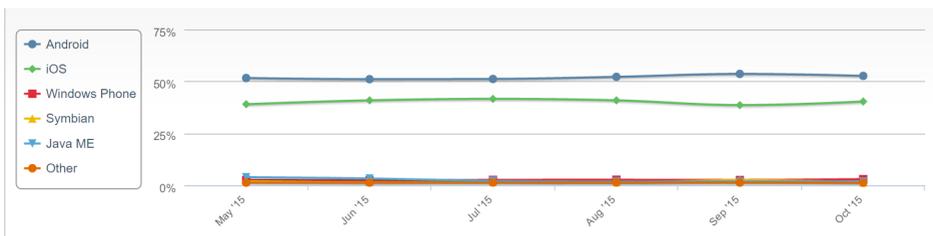


Figura 2.1 - Sistemas Operacionais móveis mais utilizados
Fonte: NET MARKET SHARE, 2015

Entre junho de 2017 e junho de 2018 o Android passou a dominar o cenário dos dispositivos móveis, alcançando 70,02% dos dispositivos, em segundo lugar o IOS com 28,63% e os demais sistemas todos abaixo de 1% (NET MARKET SHARE, 2018).

Para maior facilidade no desenvolvimento para sua plataforma, a Google disponibiliza o SDK (*Software Development Kit*) Android, que permite utilizar recursos

de hardware do aparelho como serviço de localização, Wi-Fi, acelerômetro e outros. No próprio SDK, existe a *API Google Maps*, que é utilizada no DigiAtlas para a exibição e interação do usuário com o mapa onde estão os dados disponíveis.

Quando se desenvolve para Android, existem várias formas disponíveis pelo SDK de persistir dados, por exemplo, para salvar as preferências de configuração em um *aplicativo*, é recomendado utilizar *Shared Preferences*. Como no DigiAtlas a quantidade de dados é grande, o recurso utilizado para o armazenamento destes é o banco de dados *SQLite*, também já disponível na biblioteca do Android.

A linguagem utilizada para a programação de aplicativos Android é Java, a qual é uma linguagem orientada a objetos e com documentação completa, o que torna sua utilização mais fácil. Esses aplicativos escritos em Java são compilados pela máquina virtual nativa nos dispositivos Android, *Dalvik*. A compilação gera *bytecodes Dalvik*, e assim os aplicativos podem ser executados todos dispositivos Android independente do processador utilizado.

3.3.1 API Google Maps

O Android SDK disponibiliza a API Google Maps que pode ser utilizada para a exibição do mapa produzido pela Google. Esse mapa passa por atualizações periódicas, o que é bom, pois mantém as aplicações que o utilizam sempre com fotos de satélites ou mesmo em outros modos de exibição atualizados.

No mapa também existe por padrão a opção de personalização como adicionar marcadores, traçar linhas, polígonos entre outras. Estas personalizações serão úteis a fim de proporcionar uma boa interação com o usuário sem dificultar o trabalho de desenvolvimento. A API Google Maps também disponibiliza do recurso de localização, que pode vir a ser útil em uma visita ao campo com o DigiAtlas.

3.3.2 Scrum

Para o desenvolvimento do DigiAtlas foi adotada a metodologia ágil Scrum. Essa, é uma metodologia iterativa e incremental que permite controlar o trabalho de uma equipe que trabalha para alcançar um objetivo comum. Basicamente na Scrum, tendo em mãos os requisitos para o sistema (o que deve ser feito), divide-se cada requisito em tarefas menores e estima-se o

tempo necessário para o cumprimento dessas tarefas, que quando unidas, se tornaram o requisito pronto. As tarefas foram planejadas com tempo de duração de uma semana, o que é definido como uma split na Scrum (CRUZ, 2013).

Como dito por SOARES (2004) na Scrum diariamente existem reuniões de curta duração (aproximadamente 15 minutos) onde o que foi feito e os próximos passos são discutidos. Além disso, também são apresentadas as dificuldades e os fatores de impedimentos são identificados e resolvidos.

Ao término de uma split, uma reunião era realizada a fim de conferir se o requisito estava de fato finalizado ou se eram necessárias correções para melhorar a interatividade com a aplicação.

Como no DigiAtlas não existiu uma equipe de programadores, a metodologia não foi seguida a risca e foi adaptada a fim de auxiliar o controle do que estava sendo feito e o que ainda devia ser feito, assim como também agilizar o trabalho do desenvolvedor.

3.3.3 SQLite

A utilização do SQLite e não outro banco de dados deve-se ao fato de ser o banco com suporte nativo no

Android, assim não é necessário nenhuma configuração ou instalação e possui uma documentação completa. Além disso, possui facilidades como, por exemplo, para atualizar uma linha de uma tabela, ou realizar uma pesquisa, não é necessário montar a cláusula SQL inteira, basta chamar os métodos correspondentes e passar argumentos necessários que o próprio SDK montará a cláusula e executá-la.

No caso da pesquisa, o retorno é um cursor, que referencia uma tabela onde existem somente os dados que atendem as restrições dos argumentos passados, e com esse cursor é possível percorrer essa tabela, o que facilita a programação.

3.3.4 Relacionamento entre os itens descritos na aplicação mobile

Apresenta-se na Figura 2.2 como os itens descritos nas seções acima se relacionam com o DigiAtlas Mobile e entre si. O Android SDK será a parte central, que fará comunicação com a aplicação e com a base de dados por meio da biblioteca SQLite.

Além disso é responsável por meio da API Google Maps, pela exibição e interações com o mapa, o qual possui marcações nas localidades onde existem dados já coletados. A forma como os dados serão mostrados

detalhadamente ao usuário será responsabilidade da interface do DigiAtlas.

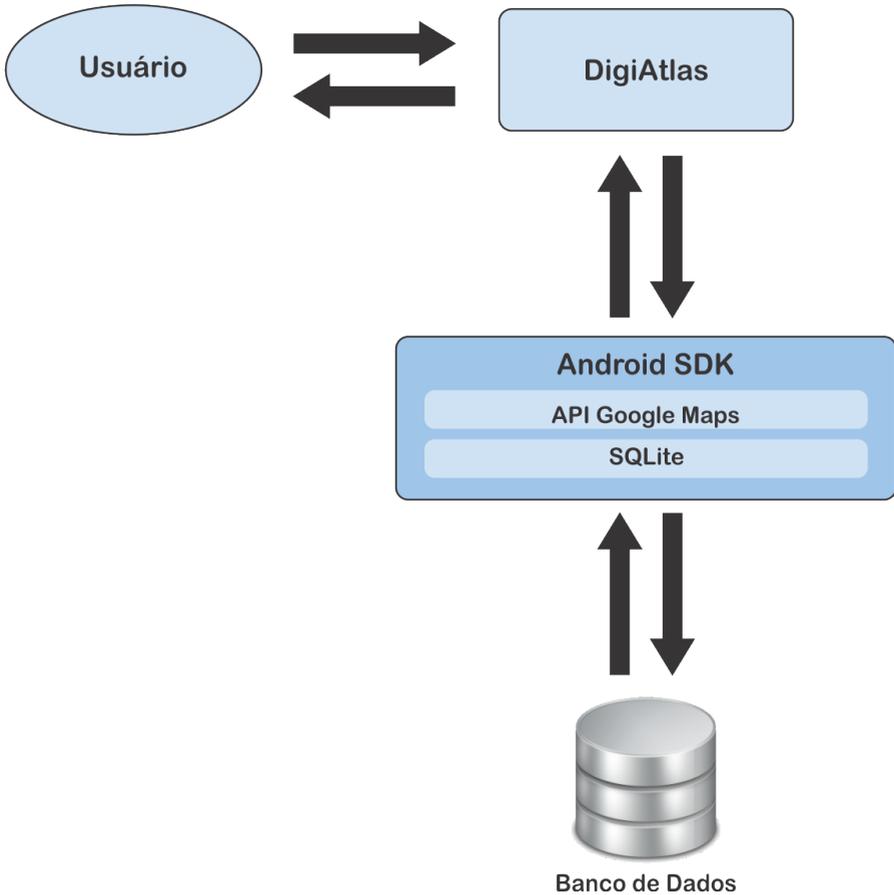


Figura 2.2 - Relacionamentos de interações entre Usuário, DigiAtlas Mobile, Android SDK e Banco de Dados.

Fonte: Os autores

3.3.5 O framework CodeIgniter

O *CodeIgniter* é um *framework* de código aberto para PHP, que tem como objetivo otimizar a produtividade e não sobrecarregá-la. Esse *framework*

utiliza o padrão *Model, View, Controller* (MVC), o que garante melhor organização do código facilitando a implementação de novas funcionalidades e a manutenção do sistema. Como a maioria dos *frameworks*, o *CodeIgniter* possui uma documentação completa com funções, estruturas, convenções e utilização. Como uma das principais características, possui a simplicidade, e assim, uma curva de aprendizagem mais rápida quando comparada a outros *frameworks*. (BLANCO; UPTON, 2009).

O *CodeIgniter* suporta vários tipos de banco de dados, como MySQL, MS SQL, Oracle, SQLite entre outros. A configuração para a utilização de cada um dos bancos compatíveis é facilmente encontrada na documentação e fóruns de dúvidas. Isso também foi um facilitador, pois como dito anteriormente, o banco de dados utilizado na aplicação servidor também foi o SQLite, a fim de facilitar a integração da base de dados dos dispositivos móveis no servidor.

O *framework* também possui bibliotecas integradas que podem ser utilizadas, como para o envio de e-mail e compressão de arquivos, o que é um facilitador pois tira a responsabilidade do desenvolvedor criá-las e reduz o tempo de programação.

3.3.6 O padrão MVC

O padrão *Model, View, Controller* (MVC) é um padrão de arquitetura de software, ou seja, uma maneira de estruturar a aplicação. Basicamente, no MVC existe as três camadas, que se relacionam (Figura 2.3).

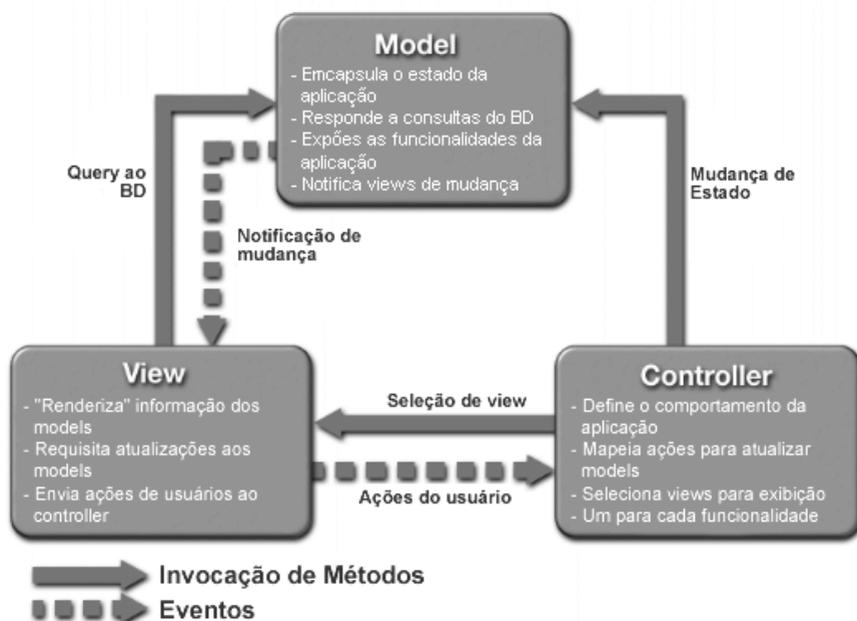


Figura 2.3 - *Design Pattern* de desenvolvimento Model-View-Controller.

Fonte: Traduzido de QURESHI; SABIR, 2013.

Cada uma das camadas foram definidas a partir do conceito de dividir tarefas, onde cada camada tenha bem definido suas funções e a comunicação entre camadas seja eficiente e de forma controlada. As camadas são descritas a seguir (DURELLI; VIANA; PENTEADO, 2008):

Model: Esta camada é responsável principalmente

pelo controle do armazenamento de dados, seja ele qual for. É o *Model* que tem contato com o banco de dados, ou seja, é onde as operações de busca, atualização, inserção e exclusão no banco de dados devem acontecer.

View: É a apresentação do sistema para seu usuário, onde se tem um design pré-definido para a apresentação das informações do sistema. Nenhuma outra parte como lógica de programação ou operações com o banco de dados acontece nesta camada, mas sim apresenta os resultados das duas coisas.

Controller: Nesta camada fica a lógica de programação, é onde se define quando, como e onde as operações acontecerão. Define o que deve ser consultado e a tela que exibirá o resultado, sendo assim um elo entre o *Model* e *View*.

3.3.7 PHP

Hypertext Preprocessor (PHP) é amplamente utilizada como linguagem de programação para o desenvolvimento web.

A linguagem PHP foi desenvolvida para que as páginas HTML, que são estáticas, pudessem ser geradas dinamicamente e rapidamente (NIEDERAUER, 2004). Seu código pode ser colocado diretamente dentro do

código HTML, desde que limitado pelas tags '`<?php`' e '`>`'.

O código PHP é executado diretamente no servidor, sendo enviado somente o código HTML resultante para o cliente. Sua sintaxe lembra C, Java e Perl. PHP infere os tipos das variáveis, ou seja, não é necessário declarar o tipo da variável, e também é fracamente tipada.

Exemplifica-se na Figura 2.4 o funcionamento da utilização da linguagem PHP para atender uma requisição de uma página HTML. Nela, o Cliente requisita para o servidor uma página HTML.

Com o PHP, é feito acesso ao banco de dados, que no exemplo abaixo é referente a um banco MySQL, e também as folhas de estilos utilizadas na página para dar cores e ajustar a responsividade, gerando assim, um layout amigável para diferentes resoluções. Uma página HTML é então construída e enviada pelo servidor ao cliente.

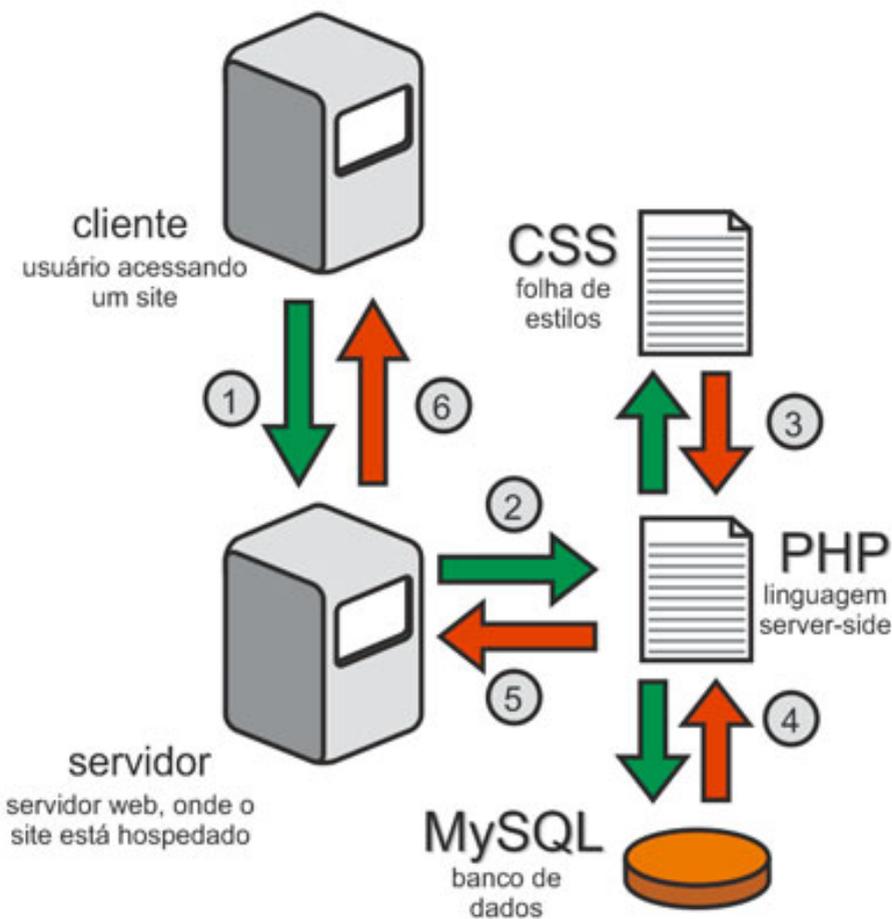


Figura 2.4 - Funcionamento da produção de uma página HTML por meio de PHP.

Fonte: Carambola Digital, 2015

3.3.8 HTML

Como descrito por Souza e Alvarenga (2004) HTML - *HyperText Markup Language* (Linguagem de Marcação de Hipertextos) é um conjunto definido de *tags* (marcações) criadas pela necessidade da construção de

documentos para serem exibidos nos computadores.

Um navegador ao ler um documento HTML interpreta as *tags* para definir como os dados contidos na página serão exibidos sem a necessidade de compilação.

3.3.9 CSS - *Cascading Style Sheetse*

De acordo com Neucamp e Lopes (2007), *Cascading Style Sheetse* (CSS) são folhas de estilo em cascata que possibilitam a criação de páginas web de uma maneira dinâmica aplicando certo estilo artístico na interface.

Permite que sejam definidos efeitos visuais e estéticos como formatação, os tipos de letras, cores, fundo, entre outros, com rapidez.

O uso do CSS é uma vantagem quando se pensa no desenvolvimento de um sistema com maior acessibilidade.

3.3.10 Documentação

A seguir são apresentados os requisitos mais importantes provenientes das reuniões periódicas realizadas no trabalho, assim como também a documentação da base de dados e da estrutura dos projetos mobile e web.

i. Requisitos

- Exibição do mapa digital (*Google Maps*)
- Criar marcadores nas localidades dos dados existentes
- Adicionar dados a marcadores existentes ou em um novo
- Edição de dados adicionados
- Exclusão de dados adicionados
- Personalização dos marcadores (cores e ícones)
- Perfis (Turma e Professor)
- Notas e seu relatório
- Importar/Exportar
- Visualizar, Adicionar e Remover mídias (áudio, vídeo e imagem)
- Regiões demarcadas
- Mapas temáticos
- Rotas
- Enviar dados para uma aplicação web
- Exibir dado e suas mídias na aplicação web e incluir ou excluir da base
- Exibir um mapa na aplicação web com marcadores dos dados existentes possibilitando mudar a localidade do dado recebido

- Criar e gerenciar usuários e suas respectivas permissões

3.3.11 Banco de dados

Os bancos de dados das aplicações web e mobile são semelhantes quanto sua estrutura, preservando as tabelas principais e diferenciando-se apenas em tabelas específicas.

As tabelas, seus respectivos atributos e relacionamentos são apresentados no Modelo Entidade e Relacionamento da Figura 2.5.

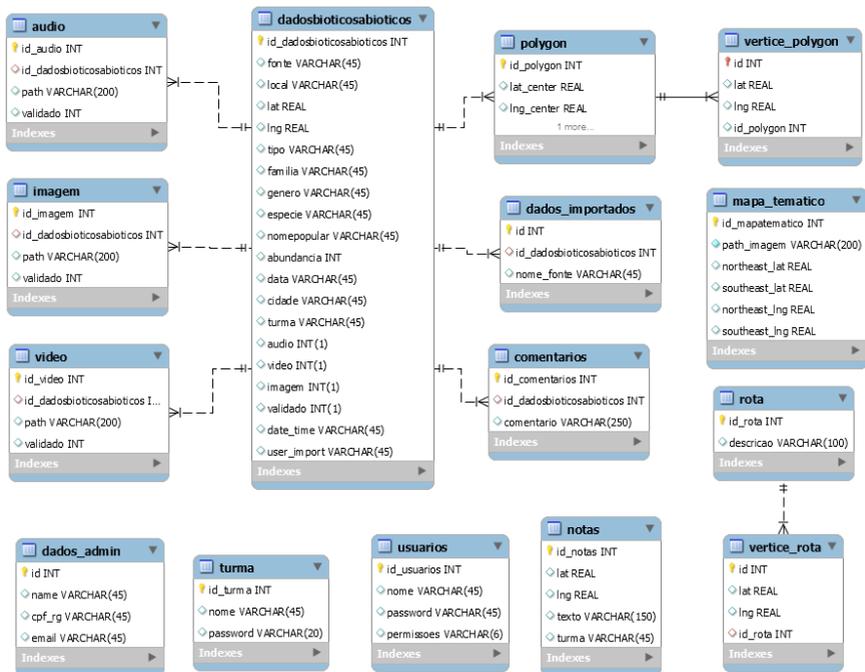


Figura 2.5 - Modelo Entidade e Relacionamento do banco de dados. Fonte: Os autores.

No DigiAtlas Mobile, as únicas tabelas que não estão presentes são "usuarios" e "comentários", enquanto no DigiAtlas Web as tabelas inexistentes são "dados_admin" e "turma". Tabelas como "rota" e "polygon" estão presentes na aplicação web, contudo, não estão sendo utilizadas pois ainda não foi desenvolvido a validação desse tipo de informação, o que pode ser um trabalho futuro.

A tabela "usuarios" define perfis de pesquisadores e suas respectivas permissões que serão responsáveis pela validação dos dados recebidos na aplicação web, e a tabela "comentarios" é utilizada para salvar comentários ou lembretes que podem ser úteis para a verificação de um dado específico.

A principal tabela da base de dados é a "dadosbioticosabioticos" que possui todos os atributos de um dado além de uma referência para cada tipo de mídia existente (atributos "audio", "video" e "imagem").

Esses atributos são utilizados para especificar os ícones de acordo com as mídias existentes nas listas de dados apresentadas na aplicação.

Além desses atributos, existe o atributo "date_time" que serve para comparar ao importar novos dados se o dado específico já foi adicionado ou não,

evitando a duplicidade de dados importados.

Por último, o atributo "user_import" é referenciado pela tabela "dados_importados" para que seja possível apagar dados importados de uma fonte específica sem afetar dados importados de outras fontes. A tabela "dados_importados" contém apenas o nome das fontes dos dados importados, porém se necessário, é possível adicionar novos atributos especificando ainda mais a origem dos dados.

Um dado ("dadosbioticosabioticos") pode possuir ou não uma ou mais mídias, sendo essas adicionadas nas tabelas "audio", "video" e "imagem". Nessas tabelas existe um atributo "path" que indica o caminho absoluto do arquivo referente à mídia assim como também uma chave estrangeira ("id_dadosbioticosabioticos") para referenciar o dado.

A tabela "dados_admin" contém os dados do administrador da aplicação mobile referente a perfil Professor. Já a tabela Turmas indica os outros perfis que possuem permissão de edição e remoção somente em dados adicionados pelo próprio perfil. O atributo "turma" na tabela "dadosbioticosabioticos" diferencia os dados de cada perfil turma.

Cada localidade pode conter anotações que são salvas

na tabela "notas". Essa tabela possui os atributos de coordenadas (latitude e longitude, "lat" e "lng") e o texto salvo como anotação. Além disso, existe o atributo "turma" que diferencia o perfil que a adicionou e tem permissão para excluir ou alterá-la.

As delimitações de regiões referenciam um dado e são salvas na tabela "polygon". Nessa tabela os atributos "lat_center" e "lng_center" referenciam as coordenadas do dado na tabela "dadosbioticosabioticos". Em uma mesma região em sua extremidade, existem diversos pontos onde são traçadas as linhas que a determinam, tais pontos são adicionados na tabela "vertice_polygon" e referenciam a mesma região por meio do atributo "id_polygon".

Na tabela "mapa_tematico" é onde são adicionadas as coordenadas nordeste e sudoeste da imagem que será sobreposta no mapa digital e o caminho absoluto da imagem. Cada coordenada é composta e a latitude e a longitude que são representadas pelos atributos "northeast_lat" e "northeast_lng" para as coordenadas nordeste e "sotheast_lat" e "sotheast_lng" para as sudeste. O atributo "path" representa o caminho absoluto da imagem do mapa temático.

Por fim, na tabela "rotas" são salvas as rotas

determinadas pelos usuários, sendo uma rota constituída por dois ou mais pontos. As coordenadas dos pontos de uma rota são salvas na tabela "vertice_rota", e o atributo "id_rota" é uma chave estrangeira para a tabela "rota" diferenciando assim, os pontos de cada rota.

3.3.12 Produção do apk

Apresenta-se na Figura 2.6 a organização do projeto mobile. Dentro da hierarquia do projeto existem primeiro, considerando uma ordem de cima para baixo, as bibliotecas necessárias para aplicações do sistema Android.

No diretório "src" após as bibliotecas, ficam os pacotes com as classes do projeto que serão descritos posteriormente.

O diretório seguinte é o "gen", que possui arquivos de referências gerados automaticamente pelo ambiente de programação e que não devem ser modificados.

No diretório "assets" ficam o arquivo de banco de dados e as mídias iniciais que serão carregados após a instalação da aplicação.

Os diretórios "bin" e "libs" possuem arquivos referentes a bibliotecas adicionais utilizadas na aplicação, que no caso do DigiAtlas fica a referência para a

biblioteca *Google Play Services* responsável pela exibição e interação com o mapa.

- ▼  DigiAltas_Mobile
 - >  Android 4.4.2
 - >  Android Private Libraries
 - >  Android Dependencies
 - >  src
 - >  gen [Generated Java Files]
 - >  assets
 - >  bin
 - >  libs
 - ▼  res
 - >  drawable
 - >  drawable-hdpi
 - >  drawable-ldpi
 - >  drawable-mdpi
 - >  drawable-nodpi
 - >  drawable-xhdpi
 - >  drawable-xxhdpi
 - >  layout
 - >  layout-land
 - >  layout-port
 - >  menu
 - >  values
 - >  values-sw600dp
 - >  values-sw720dp-land
 -  AndroidManifest.xml
 -  lint.xml
 -  local.properties
 -  proguard-project.txt
 -  project.properties

Figura 2.6 - Organização DigiAtlas Mobile.

Fonte: Os autores

No diretório “res” estão os diretórios que referem a algo da interface. Nele estão contidos diversos diretórios de imagens (*drawable*) e textos (*values*).

Esses diretórios especificam diferentes resoluções, assim, a aplicação sabe automaticamente em qual diretório procurar itens para melhor adaptar o conteúdo nas diferentes resoluções e tamanho de telas dos dispositivos.

O diretório “land” contém os arquivos de interface da aplicação que se adaptam automaticamente a orientação do aparelho, paisagem ou retrato.

Para obter-se uma melhor adaptação, pode-se definir uma interface para cada tipo de orientação, para isso, deve-se criar um arquivo na pasta “layout-land” que determinará o layout da orientação de paisagem e outro de mesmo nome na pasta “layout-port” que será o layout da orientação de retrato, assim a aplicação consegue automaticamente a partir do estado de orientação do dispositivo móvel exibir a interface correspondente.

Os menus localizados na barra superior da aplicação são definidos no diretório “menu”.

Por último, o arquivo “AndroidManifest.xml” define as configurações do projeto, por exemplo, a versão

mínima necessária para utilizar a aplicação e também a chave de licença para fazer conexão com a API Google Maps.

Essa chave e as chaves para outros recursos da Google são obtidas sem custo no seu site de desenvolvedores.

- ▼  src
 - >  com.digiatlas.anexodados
 - >  com.digiatlas.controllers
 - >  com.digiatlas.galerias
 - >  com.digiatlas.hardware
 - >  com.digiatlas.import_export
 - >  com.digiatlas.mapa.controller
 - >  com.digiatlas.mapa.model
 - >  com.digiatlas.mapa.view
 - >  com.digiatlas.mapas_tematicos
 - >  com.digiatlas.models
 - >  com.digiatlas.object
 - >  com.digiatlas.pesquisa
 - >  com.digiatlas.util
 - >  com.digiatlas.views

Figura 2.7 - Pacotes da aplicação mobile.

Fonte: Os autores

Os pacotes da aplicação foram estruturados conforme indicado na Figura 2.7. Para maior facilidade na descrição de cada pacote, os mesmos são referenciados somente pela parte final de seu nome

desconsiderando a parte inicial "com.digiatlas", ou seja, o pacote "com.digiatlas.mapa.controller" será referenciado somente como "mapa.controller".

Seguindo uma sequência de cima para baixo na lista de pacotes, o pacote "anexodados" contém as classes responsáveis por compactar diretórios ("ZipDiretorio.java"), enviar os dados para o servidor ("FileUpload.java"), cadastrar o usuário administrador ("CadastroAdmin.java") e anexar os dados que serão enviados para a aplicação web.

No pacote "controllers" estão os controladores de edição e exibição de dados, lista de dados, login, notas e das funcionalidades da tela secundária.

O pacote "galerias" contém as classes responsáveis pela exibição e gerência das listas de áudios, vídeos e imagens assim como a reprodução/exibição dessas mídias.

As classes responsáveis por utilizar os recursos do aparelho para adicionar novas mídias estão contidas no pacote "hardware".

As classes responsáveis pela importação e exportação de dados assim como também a definição da fonte dos dados importados estão no pacote "import_export".

Os controladores de Rotas, Mapas Temáticos, Regiões Abióticas, marcadores, dados e localização ficam no pacote "mapa.controller".

No pacote seguinte "mapa.model", ficam os *models* dos marcadores, Regiões Abióticas, Rotas e Mapas Temáticos.

É no pacote "mapa.view" que está localizada a view responsável pela exibição do mapa digital.

A *view* dos mapas temáticos se encontra no pacote "mapas_tematicos", assim como também a classe responsável pela exibição da lista de mapas temáticos sobreposta no mapa.

Os *models* da edição e remoção de dados, lista de dados, login, notas e funcionalidades da tela secundária se encontram no pacote "models".

No pacote "object" existem duas classes que representam o objeto de um dado ("Dado.java") e o de um marcador do mapa ("MarcadorDA.java").

A *view*, o *model* e o *controller* da funcionalidade pesquisa estão no pacote "pesquisa".

Inseridas no pacote "util" estão as classes de auxílio. Dentre elas está a classe "Cnt.java" que contém constantes, como exemplo, o nome de tabelas do banco de dados.

Também existe a classe “Mensagem.java” utilizada para a exibição de mensagens para o usuário, e a classe “OpenURL.java” que serve para redirecionar a sites no navegador do dispositivo.

No pacote “views”, estão todas as classes responsáveis pela exibição das interfaces que não sejam relacionadas ao mapa. São elas: cadastro de turmas, criar banco de dados com as mídias existentes na primeira utilização, edição de dado, edição de perfis, login, tela secundária, janela de informações, relatório de notas e informações do grupo.

3.3.13 Produção da versão web

A estrutura da aplicação web utilizando como base a estrutura do framework CodeIgniter (Figura 8). Os principais diretórios dessa estrutura são “config”, “controllers”, “helpers”, “models” e “views”. Esses diretórios são descritos a seguir.

Toda a parte de configuração é realizada nos arquivos PHP encontrados no diretório “config”. É nesse diretório que se encontra a configuração para o acesso ao banco de dados assim como também o pré-carregamento de bibliotecas auxiliares.

As configurações são facilmente encontradas na

documentação do *framework* assim como também em fóruns de dúvidas.

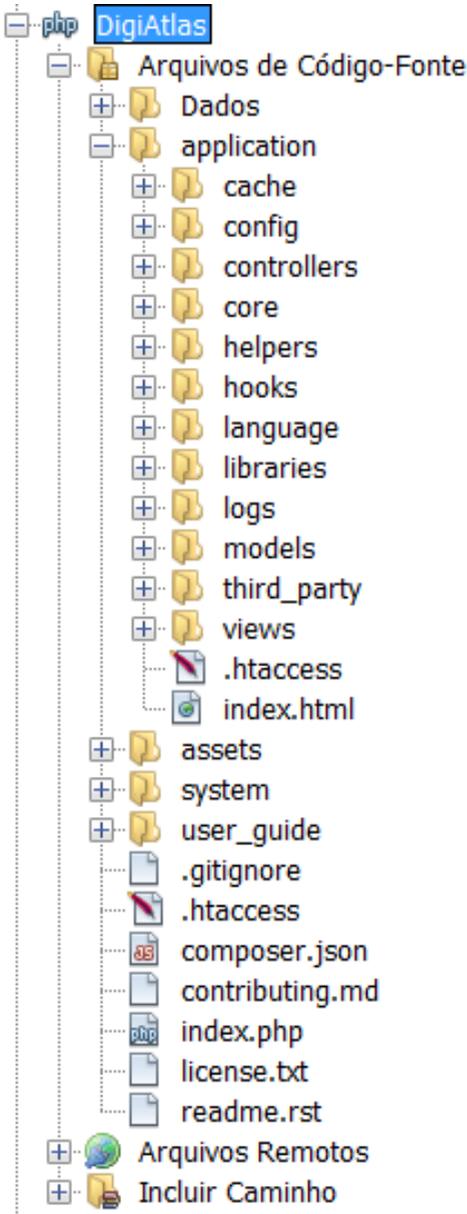


Figura 2.8 - Organização DigiAtlas Web.
Fonte: Os autores

Como o próprio nome induz, o diretório "controllers" possui os arquivos dos controladores. No DigiAtlas Web até o momento existem os controladores "Administration.php", "Dado.php", "Extrair.php", e "User".

O primeiro deles é responsável pelas operações de gerenciamento de usuários e permissões. Já o controlador "Dado.php" é onde ocorre a validação ou exclusão de um dado com suas mídias. O controlador "Extrair.php" é que recebe novos dados, adicione-os no banco da aplicação e organiza os arquivos em pastas específicas que se encontram no diretório "Dados" e serão descritas mais adiante. "User.php" é responsável pelas operações da tela inicial de autenticação da aplicação.

O diretório "helpers.php" não foi utilizado, contudo, é nele que ficam arquivos com funções que podem ser acessadas de qualquer controlador evitando a duplicação dessas funções em controladores diferentes.

No diretório "models" é onde estão localizados os arquivos de acesso ao banco de dados diferenciando entre esses arquivos o tipo de informação acessada/modificada.

Nele temos "Comentarios_model.php",

“Dados_model.php”, “Marker_model.php” e “User_model.php”.

Todas as interfaces incluindo o cabeçalho e rodapé das páginas web estão no diretório “views”. Os arquivos são: “alterar_senha.php”, “criar_usuario.php”, “editar_usuario.php”, “dado.php” (exibe o dado), “index.php” (autenticação) e “main.php” (seleção de lista de dados de um tipo).

4 Resultados e discussões

Como resultado, surgiram duas aplicações, o DigiAtlas *Mobile* e o DigiAtlas *Web* que serão descritas nos tópicos a seguir. Posteriormente, serão apresentados os resultados dos testes de usabilidade sendo o primeiro com professores do projeto e o segundo com estudantes de graduação dos cursos de Ciências Biológicas e Geografia. Neste capítulo em geral é apresentada a metodologia do trabalho.

4.1 DigiAtlas Mobile

O DigiAtlas Mobile foi projetado para gerenciar dados ambientais e ecológicos, podendo ser utilizado como ferramenta de ensino-aprendizagem em diferentes etapas e disciplinas do ensino fundamental e médio.

Como já descrito, inicialmente a aplicação possui dados disponibilizados por pesquisadores e pós-graduandos da Unifal-MG derivados de iniciações científicas, projetos de extensão ou pós-graduação. Novos dados (imagens, sons, vídeos e textos) podem ser inseridos na base local do aplicativo em atividades desenvolvidas pelos professores e estudantes.

Os dados estão dispersos em um mapa, dando uma melhor noção da dispersão dos dados e melhor visualização da localidade de cada dado.

É no mapa onde ocorre as operações de gerenciamento (adição, edição e remoção) e outras funcionalidades que serão descritas adiante.

O DigiAtlas Mobile foi desenvolvido para dois perfis: Professor e Turma. O perfil Professor é único, o administrador da aplicação e não pode ser excluído. Ele é quem pode criar ou apagar um ou mais perfis Turmas além de poder modificar dados desses perfis.

O perfil Turma tem acesso somente a suas anotações e pode realizar operações de exclusão e edição somente em seus dados, assim, é possível que em um mesmo aparelho, a aplicação possa ser utilizada com diferentes classes abordando diferentes assuntos.

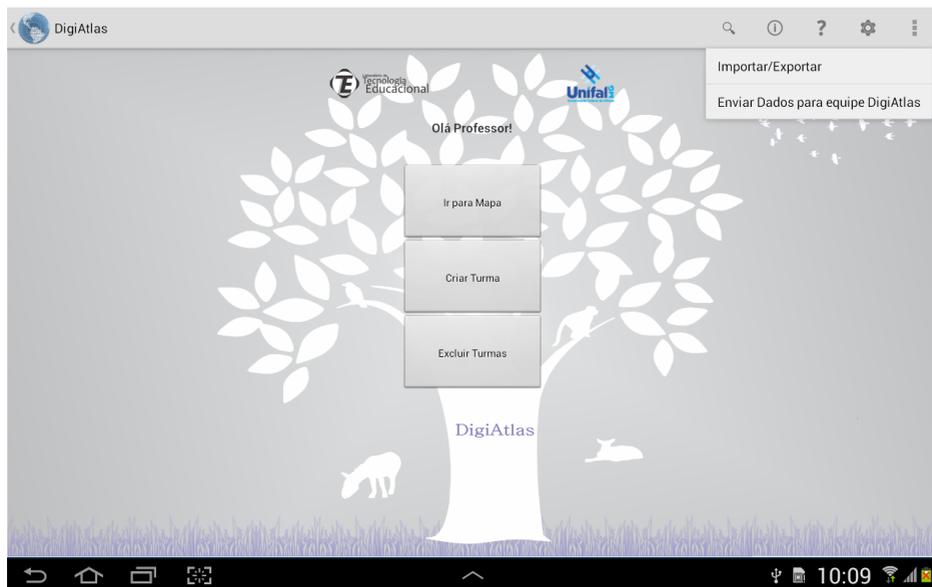


Figura 2.9 - Tela secundária do DigiAtlas Mobile.

Fonte: Os autores

Observa-se na Figura 2.9 a interface após autenticação como Professor, onde existem várias funcionalidades disponíveis para o usuário. “Na barra superior em seu canto direito, existem as opções: Pesquisa, Informações, Ajuda, Configurações, Importar/Exportar e Enviar Dados para equipe DigiAtlas, essas três últimas disponíveis somente para Professor. Além disso, existem as funcionalidades “Ir para Mapa”, “Criar Turma” e “Excluir Turma”(FERREIRA et al., 2015).”

O mapa disponibilizado pela API Google Maps é apresentado no aplicativo e nele estão marcadas as

localidades que possuem pelo menos um dado coletado. Como em uma localidade pode existir mais de um tipo de dado, às informações de cada dado são exibidas após a seleção prévia do marcador.

No DigiAtlas *Mobile*, existem três tipos de marcadores, os verdes, amarelos e vermelhos como mostrado na Figura 10. Essas cores indicam se os dados existentes na localidade são validados ou não-validados.

Dados validados são aqueles que foram analisados e confirmados por pesquisadores/professores da UNIFAL-MG. Dados não-validados são aqueles inseridos pelos próprios usuários. A base de dados atual e as disponibilizadas por atualizações já estarão validadas (FERREIRA et al., 2015).

Os marcadores verdes indicam que naquela localidade existem somente dados validados (inseridos na base de dados original do aplicativo), os amarelos indicam aqueles que possuem dados validados e não-validados (inseridos no banco de dados local do aparelho pelo usuário), e por fim os vermelhos indicam as localidades que possuem apenas dados não-validados (somente dos inseridos pelo próprio usuário na base de dados locais).

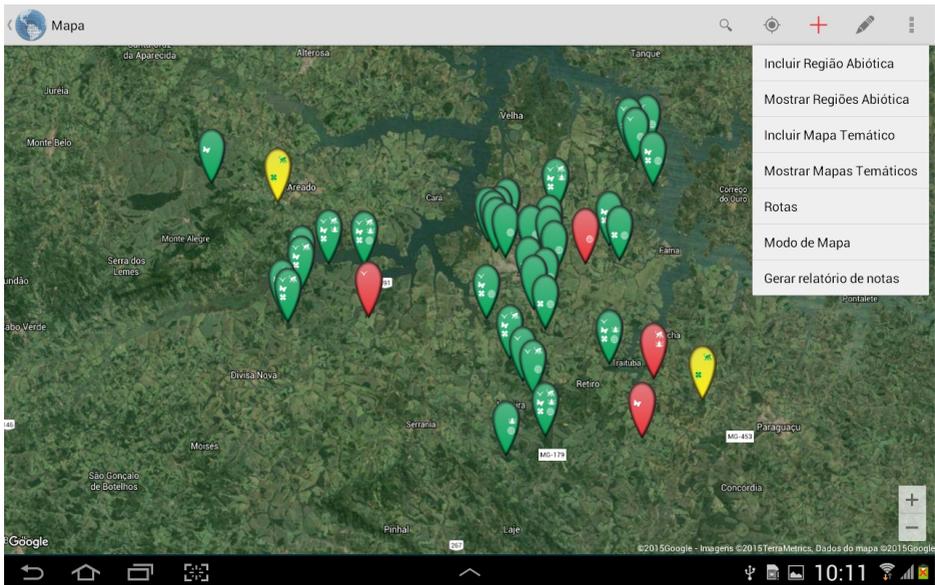


Figura 2.10 - Mapa com diversos marcadores e a opção de adição ativada.

Fonte: Os autores

O DigiAtlas possibilita que dados não-validados sejam exportados, permitindo assim o compartilhamento de dados entre os usuários da aplicação ou mesmo como forma de backup.

Quando realizada uma exportação, uma pasta contendo todos os arquivos de mídias (áudio, imagens e vídeos) dos dados é criada (Figura 2.11) e o usuário poderá optar pelo nome desejado.

Dessa forma, os dados obtidos pelos usuários (não validados), podem ser salvos para posterior utilização ou compartilhamento.

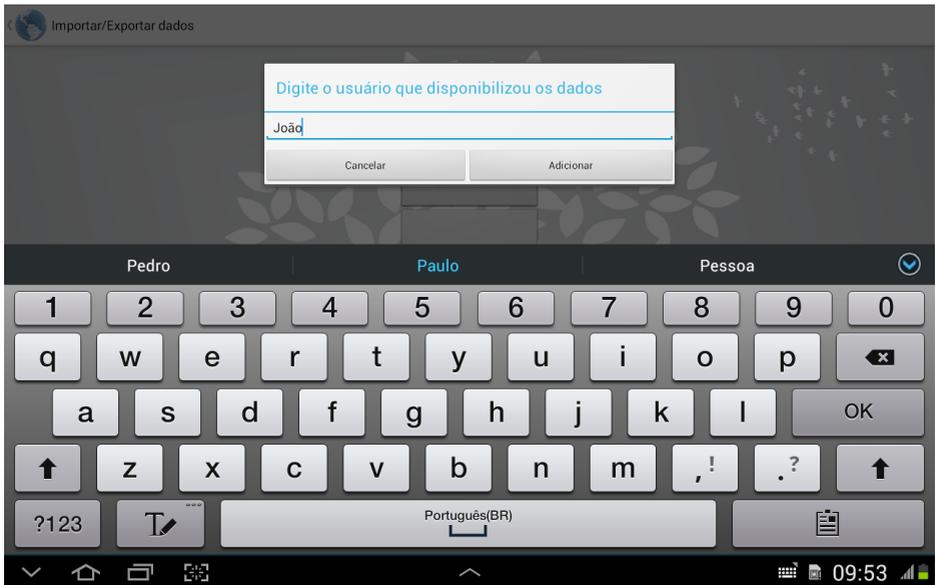


Figura 2.11 - Exemplo da tela para informar a fonte de dados na importação.

Fonte: Os autores

Com essas mídias, um arquivo de registro dos dados é criado, o "export.da". O compartilhamento acontece com a importação desses dados, necessitando apenas selecionar o arquivo "export.da" por meio de qualquer aplicativo gerenciador de arquivos e informar a fonte dos dados, pois assim, tem-se a possibilidade de exclusão dos dados de uma fonte específica, não interferindo na permanência de dados de outras fontes na aplicação (Figura 2.12).

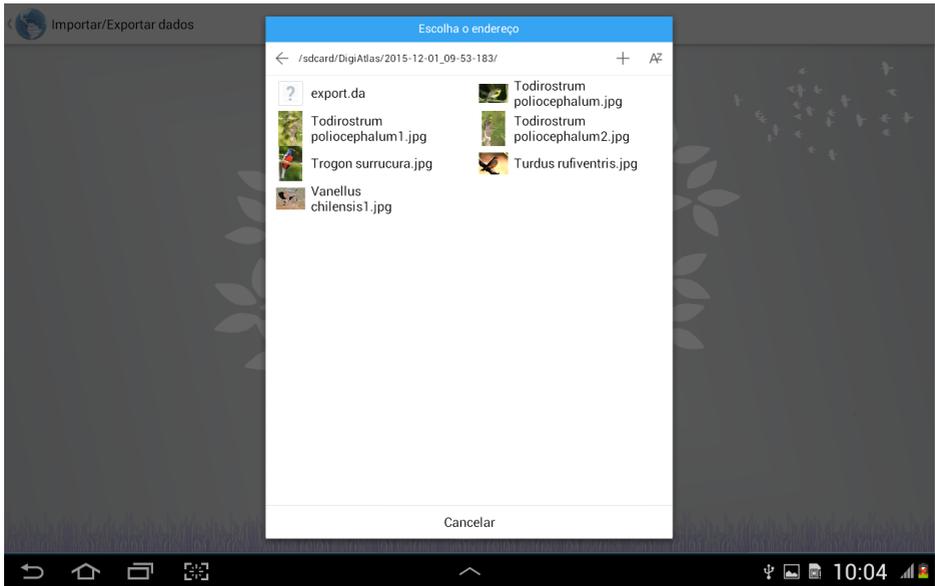


Figura 2. 12- Pasta com mídias e o arquivo “export.da”.

Fonte: Os autores

Existem seis tipos de dados na aplicação: Aves, Mamíferos, Invertebrados, Anfíbios, Plantas e Abióticos (que envolvem problemas ambientais e físicos).

Assim, dados adicionados/editados devem ser classificados em um dos seis tipos obrigatoriamente, resultando em uma melhor organização dos dados na aplicação. Os marcadores possuem ícones que indicam os tipos de dados existentes na localidade (Figura 2.13), com isso o usuário não precisa acessar os marcadores para saber os tipos de dados que ele possui, o que melhora a interação com a aplicação.



Figura 2.13 - Marcador com ícones dos seis tipos.

Fonte: Os autores

Na barra superior da interface do mapa (Figura 2.10), existem as seguintes funcionalidades: Pesquisa, Localização, Adição, Edição/Exclusão de dados, Adição de Região Abiótica, Exibir/Ocultar Região Abiótica, Adição de Mapa Temático, Exibir/Ocultar Mapa Temático, Rotas, Modo de Mapa (já disponível pelo *Google Maps*) e Gerar relatório de notas.

Quando ativa uma funcionalidade seu ícone torna-se vermelho, assim o usuário sabe exatamente qual funcionalidade está ativa diminuindo as chances de erros na utilização da aplicação.

A operação de Pesquisa além de ser acessível por meio do mapa, também está disponível na interface após a autenticação (Figura 2.9) por meio do ícone da lupa na barra superior no canto direito.

A priori, a busca é feita em toda a base de dados e em todos atributos, ou seja, os dados que estarão no resultado da busca, são aqueles em que há ocorrência do que foi especificado pelo usuário em algum atributo.

Após a exibição do resultado da busca, “existe a possibilidade de filtragem de acordo com um atributo desejado, o que permite diminuir a quantidade de resultados apresentados.”(FERREIRA et al., 2015).

No final da linha de cada dado no resultado da Pesquisa, existem ícones que indicam se aquele dado possui ou não alguma mídia.

É possível após o resultado de uma busca ir para o mapa e visualizar somente os marcadores dos dados resultantes por meio da operação “Ver Dados no Mapa” na barra superior (Figura 2.14).

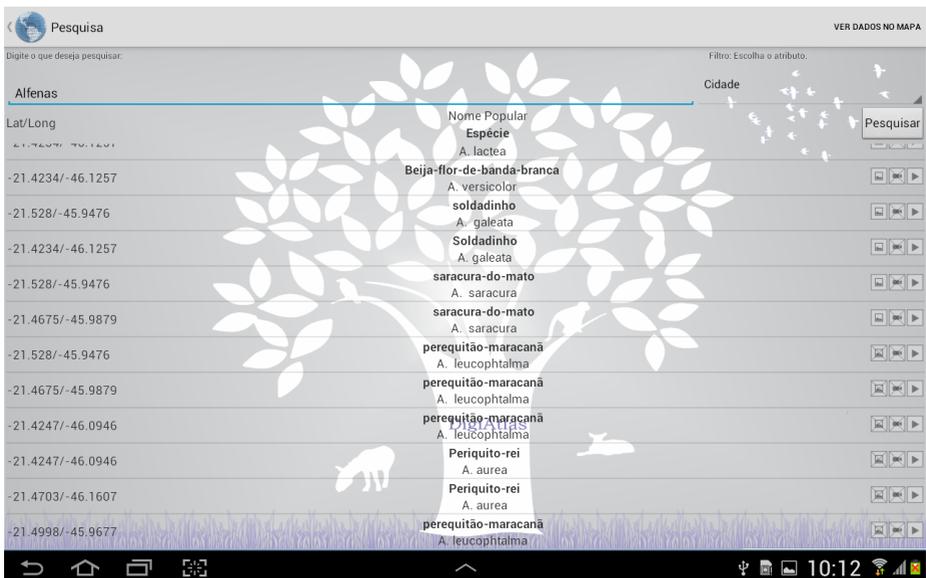


Figura 2.14 - Pesquisa.
Fonte: Os autores

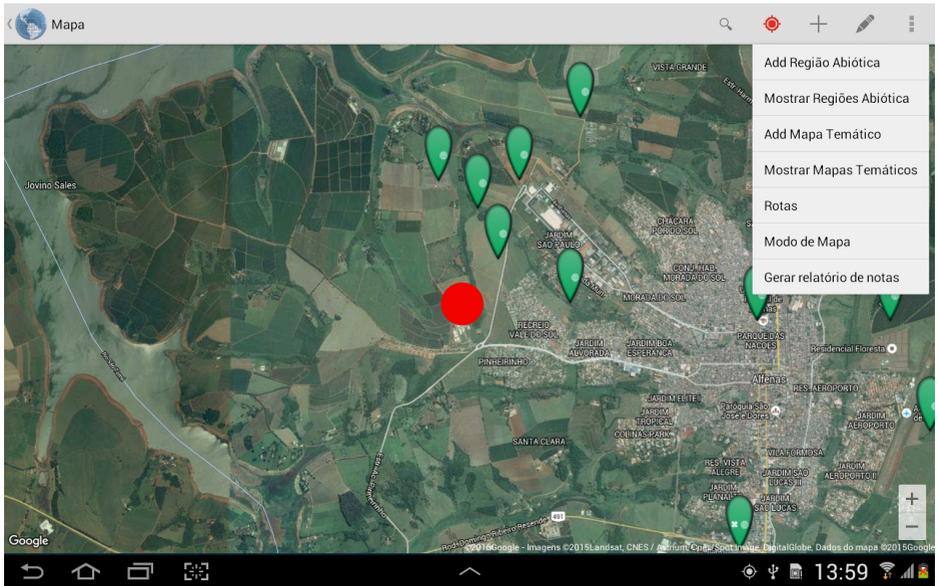


Figura 2.15 - Localização.

Fonte: Os autores

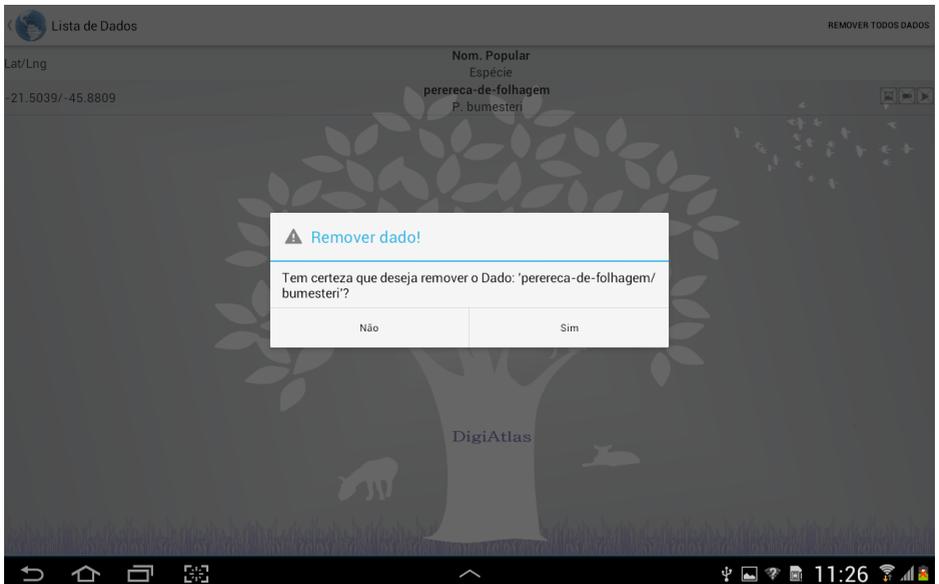


Figura 2.16 - Confirmar remoção e opção "Remover Todo Dados" em segundo plano.

Fonte: Os autores



Figura 2.17 - Edição

Fonte: Os autores

O campo de visão do mapa desloca-se e um círculo vermelho surge quando o usuário requisita uma localização (Figura 2.15).

A operação de Adição permite “adicionar novos dados em marcadores existentes ou ainda adicionar um novo marcador, e em seguida adicionar o novo dado neste marcador (FERREIRA et al., 2015).”

As operações de Edição/Remoção estão juntas e só podem ser realizadas em dados não-validados. Quando ativas essas operações, ao tocar em algum marcador, a lista de dados não-validados da referente localidade é exibida. Para excluir algum dado basta um toque longo e

confirmar a exclusão.

Existe ainda a possibilidade de excluir todos os dados da lista por meio da opção “Remover Todos Dados” na barra superior (Figura 2.16).

Para editar basta um toque rápido e todos os atributos do dado estarão disponíveis para a alteração (Figura 2.17).

Região Abiótica é qualquer região que seja relevante para o estudo, por exemplo, uma área de preservação ambiental.

No DigiAtlas Mobile é possível delimitar essas regiões no mapa, basta selecionar a opção “Inserir Região Abiótica” e marcar os pontos que limitam a área. Para visualizar as regiões marcadas deve-se acionar a opção “Mostrar Regiões Abióticas”.

Quando esta opção for selecionada, em seu lugar surge “Ocultar Áreas Marcadas”, que como o próprio nome já diz, oculta as áreas marcadas no mapa (Figura 2.18).

É possível adicionar imagens sobrepostas no mapa que contenham qualquer informação relevante para o estudo como o índice de chuva ou a temperatura em diferentes épocas de uma região. Essas imagens são denominadas Mapas Temáticos.

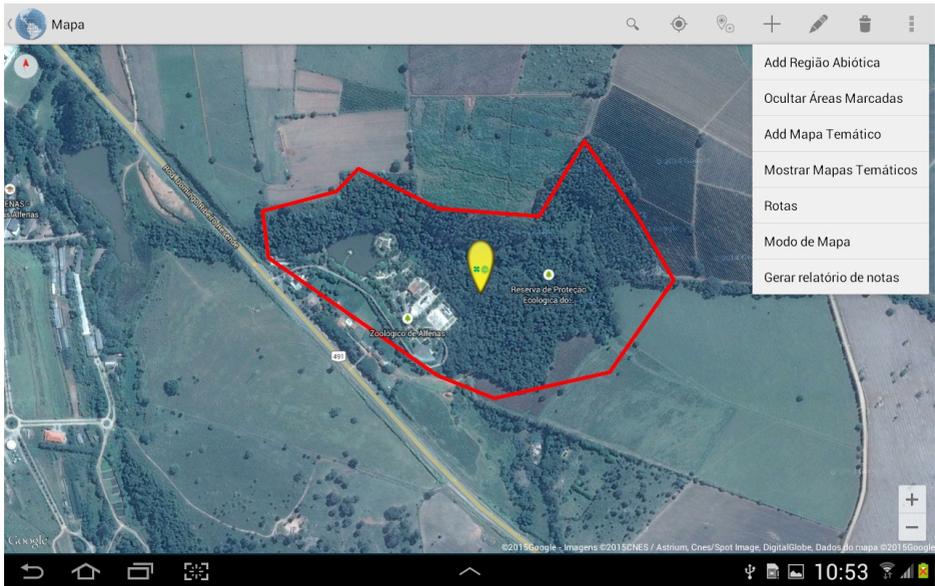


Figura 2.18 - Região abiótica.

Fonte: Os autores

Para adicionar um mapa temático, deve-se selecionar a opção “Inserir Mapa Temático” e em seguida informar as coordenadas nordeste e sudoeste.

A opção “Mostrar Mapas Temáticos” exibe todos os mapas já adicionados, e se transforma em “Ocultar Mapas Temáticos”, possibilitando ocultar os itens exibidos. Um mapa temático pode estar totalmente ou parcialmente sobre outro, por isso, ao exibir esses mapas na aplicação, é possível determinar quais mapas são exibidos ou não e também a ordem de sobreposição entre eles. Além disso, pode-se alterar a transparência da imagem sobreposta no mapa.

Os usuários podem determinar rotas em qualquer parte do mapa para, por exemplo, para auxiliar a ida dentro de uma mata fechada e chegar ao local de um dado. Ao selecionar a opção "Rotas", todas as rotas são exibidas e a opção de criar novas rotas aparece na barra superior. Para adicionar uma nova rota, usuário deve tocar no mapa sequencialmente ponto a ponto, onde entre esses pontos são criadas linhas que formarão a rota. No momento da criação de uma rota, existe a possibilidade de desfazer a última adição, possibilitando que ao selecionar não intencionalmente um ponto fora da rota, o usuário não necessite recomeçar a marcação dos pontos (Figura 2.19).

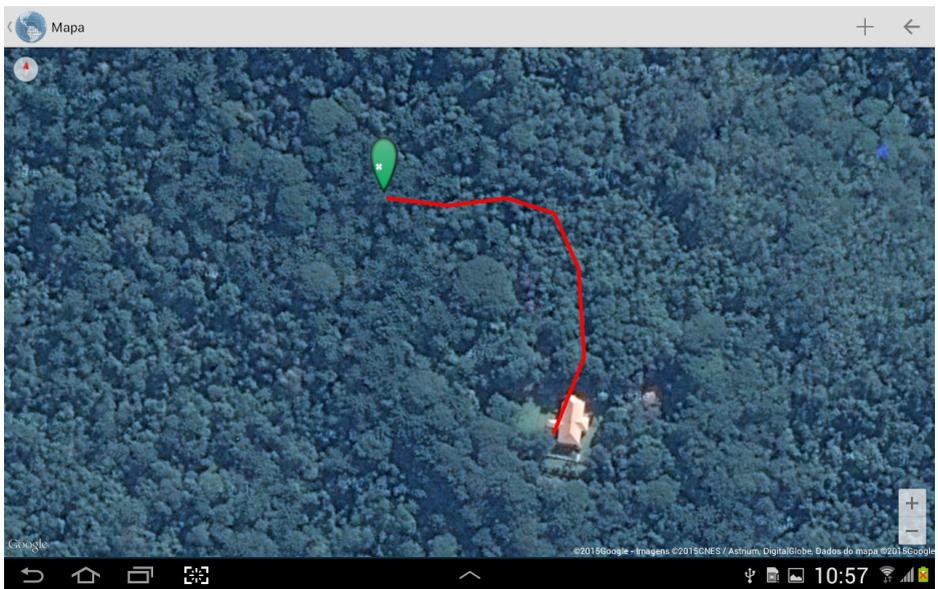


Figura 2.19 - Função de criação de rotas.
Fonte: Os autores

Por último, a opção gerar relatório de notas exibirá todas as anotações do usuário autenticado com suas respectivas coordenadas, exceto para o professor que pode visualizar as anotações de todos os usuários (Figura 2.20).



Figura 2.20. Relatório de notas.
Fonte: Os autores

Quando seleciona-se um marcador (Figura 2.21) uma janela de informações é exibida. Nesta janela existem imagens nítidas que representam os tipos de dados existentes nessa localidade, e imagens acinzentadas, que representam os tipos de dados inexistentes. Após as imagens, são exibidas as anotações específicas da localidade, sendo possível adicionar/editar tais anotações. Ao final da janela de informações estão

as coordenadas desse marcador (Latitude/Longitude).

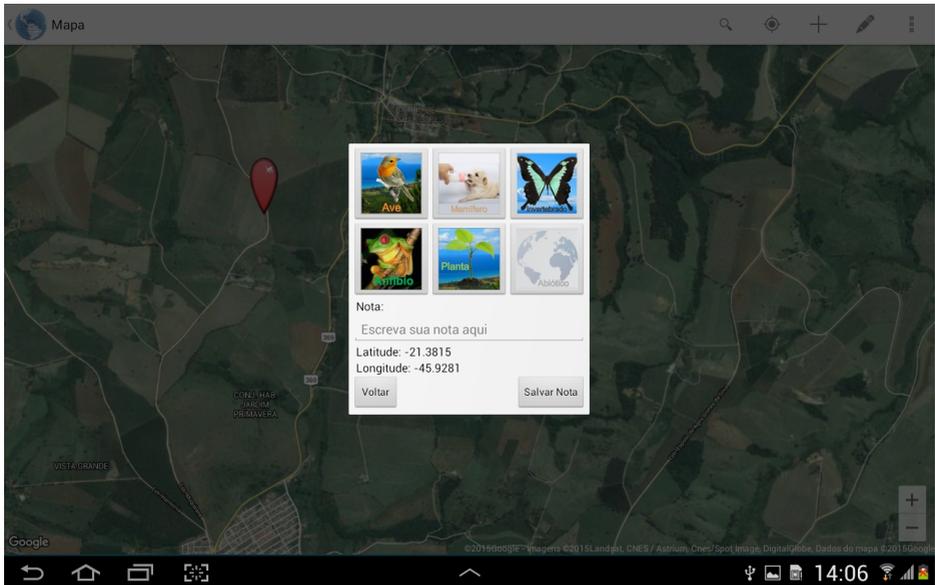


Figura 2.21 - Janela de informações.

Fonte: Os autores

As informações de um dado (Figura 2.22) são apresentadas depois de uma pesquisa ou a partir da interação com o mapa. Os atributos são os mesmos apresentados na seção “Dados ambientais”, e como dito um dado não necessariamente possui todos atributos.

Os atributos são: Fonte que coletou e disponibilizou o dado; Latitude/Longitude que especifica a localização; Local em que foi realizada a coleta; Tipo de dado (Aves, Mamíferos, Invertebrados, Anfíbios, Plantas e Abióticos); Família, Gênero e Espécie; Nome Popular (se existir); Ocorrência especificando o número indivíduos visualizados no local; Data da coleta; Cidade em que foi realizada a coleta; Comentário ou alguma informação extra sobre o dado (FERREIRA et al., 2015, p. 699).

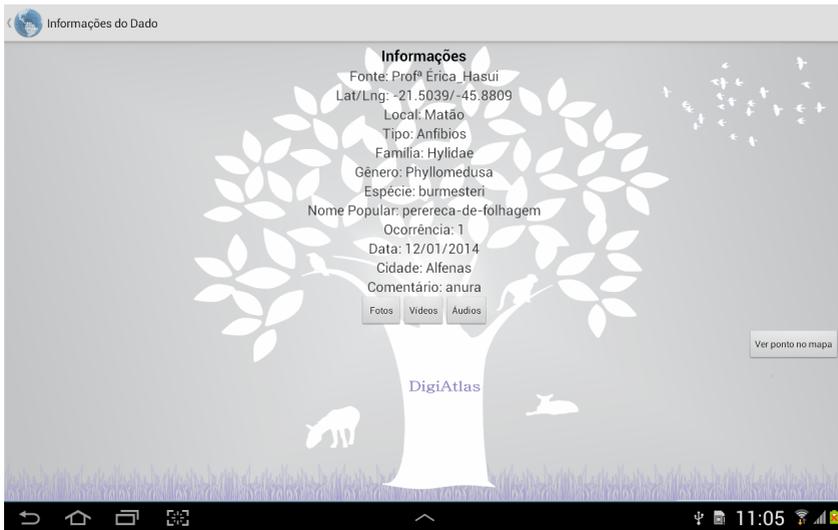


Figura 2.22 - Informações de um dado.

Fonte: Os autores.

A partir das informações do dado pode-se ir para o mapa e ver a localização do marcador desse dado, além de ter acesso às galerias de áudio, fotos e vídeos. Nessas galerias, as mídias existentes são apresentadas e ainda é possível adicionar novas mídias com os recursos disponíveis no próprio aparelho.

4.2 DigiAtlas Web

O DigiAtlas Web foi desenvolvido pela necessidade de verificar dados que os usuários desejam compartilhar com todos os outros da aplicação, e assim, manter a integridade do banco de dados.

Na aplicação, cada dado será verificado por um pesquisador especialista em seu tipo, analisando sua

ortografia e sua(s) mídia(s) (imagens, áudios e vídeos) quando existente(s).

No DigiAtlas Mobile, quando selecionado pelo usuário a funcionalidade “Enviar Dados para a equipe DigiAtlas”, os arquivos são exportados para uma pasta contendo as mídias e o registro dos dados não-validados no arquivo “export.da”.

Após isso, a pasta é comprimida criando um arquivo que tem como nome a data e o horário do momento, e é iniciado o upload desse arquivo para o servidor. Os arquivos recebidos ficam inicialmente em um diretório de nome “novos”.

Automaticamente ao término do *upload*, a pasta contendo as mídias e o registro de dados é exportada para o diretório “midias” e o arquivo comprimido é movido para o diretório “backup”. Após isso, o arquivo “export.da” é interpretado para que os novos dados sejam adicionados na base de dados do DigiAtlas Web e exibidos para a validação ou exclusão.

Os dados que forem validados como já dito, serão disponibilizados para a aplicação mobile posteriormente por meio de atualizações.

Para utilizar o DigiAtlas Web é necessário autenticar-se. Existe um usuário administrador que é o

responsável por gerenciar todos os outros perfis, ou seja, é o administrador que cria ou exclui e determina quais são as especialidades de cada perfil (Figura 2.23 e 2.24).

DigiAtlas: Olá, Admin

Pesquisar...

Dados a serem validados

Aves

Mamíferos

Plantas

Anfíbios

Invertebrados

Dados Abióticos

Dados já validados

Criar Usuário

Usuário:

Login do novo usuário

Senha

Senha

Digite a senha novamente

Permissões de Validação

Aves Mamíferos

Plantas Invertebrados

Anfíbios Abióticos

Criar Usuário

- Criar Usuário
- Gerenciar Usuários
- Alterar Senha
- Logout

Figura 2.23 - Criar usuário.

Fonte: Os autores

As especialidades são referentes aos tipos dos dados e definem se o usuário logado em pode ou não analisar o dado. Por exemplo, o perfil do usuário “Lucas” (Figura 2.24) pode validar qualquer dado dos seis tipos: “Aves”, “Mamíferos”, “Plantas”, “Invertebrados”, “Anfíbios” e “Abióticos”.

A qualquer momento o administrador pode alterar essas permissões ou mesmo excluir o perfil do usuário por meio da funcionalidade “Gerenciamento de Usuários”.

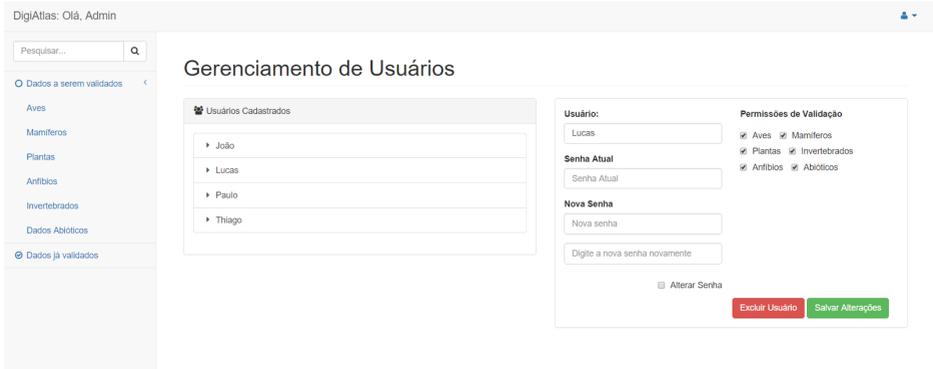


Figura 2.24 - Gerência de usuários.
Fonte: Os autores

Para criar um usuário basta utilizar a operação “Criar usuário” e definir o nome, uma senha e as permissões.

Tais funcionalidades estão presentes na barra superior no canto direito e estão disponíveis somente para o administrador. Além dessas, na barra existe as opções “Alterar senha” e “Logout” disponíveis para todos os usuários (Figura 2.25).

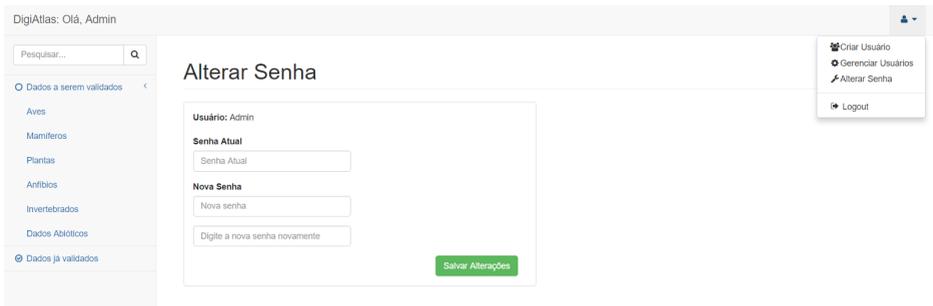


Figura 2.25 - Alterar senha.
Fonte: Os autores

No menu posicionado no lado esquerdo da

interface, existe um link para cada uma das listas de dados disponíveis para a validação como mostra a Figura 2.26.

Ao entrar em qualquer dado da lista, todos os atributos são mostrados e se o usuário possuir permissão para validar esse tipo de dado, é possível editar qualquer atributo, selecionar quais mídias estarão ou não disponíveis se o dado for validado ou ainda se for o caso, excluir o dado e as mídias.

Na interface onde os atributos e as mídias do dado são exibidos, existe um mapa onde a localização em que o dado foi coletado é representada por um marcador vermelho e as demais localidades por verde.

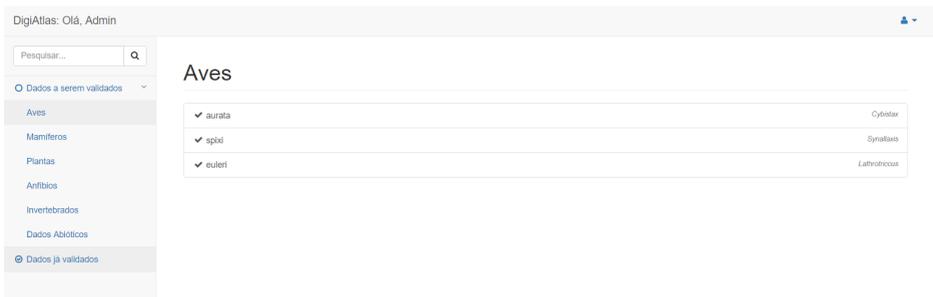


Figura 2.26 - Lista de dados do tipo aves.
Fonte: Os autores

Quando existir algum marcador próximo, se for conveniente o especialista pode definir as mesmas coordenadas para o dado evitando a criação diversos marcadores que referem a um mesmo local, por

exemplo, uma mata extensa.

Essa definição pode ser feita por meio da digitação das coordenadas nos campos “Lat” e “Lng” (Latitude e Longitude) ou pelo botão na janela de informações do marcador, que é exibida após a seleção do próprio (Figura 2.27).

A posição do marcador vermelho se altera dinamicamente conforme muda-se os valores dos campos de coordenadas (“Lat” e “Lng”), permitindo que o usuário confira visualmente a localidade em que o dado será adicionado.

The screenshot displays the 'Validar Dado' (Validate Data) interface in the DigiAtlas application. On the left, there is a search bar and a sidebar with navigation options: 'Pesquisar...', 'Dados a serem validados', 'Aves', 'Mamíferos', 'Plantas', 'Anfíbios', 'Invertebrados', 'Dados Abióticos', and 'Dados já validados'. The main form is titled 'Validar Dado' and contains the following fields:

- Lat/Lng (Recebida):** -21.71531-45.8645
- Família:** Furnariidae
- Lat a ser salva (4 casas decimais):** -21.7153
- Gênero:** Synallaxis
- Lng a ser salva (4 casas decimais):** -45.8645
- Espécie:** spixi
- Local:** Comprido
- Nome Popular:** Pardal
- Cidade:** Atenas
- Abundância:** 1
- Fonte:** Profª Erica_Hasui
- Data:** 27/8/2015
- Comentário:** (empty text area)

Below the form, there are radio buttons for selecting the taxonomic group: Aves (selected), Mamíferos, Plantas, Invertebrados, Anfíbios, and Abióticos. At the bottom of the form, it says 'Usuário que disponibilizou:'. To the right of the form is a map titled 'Verifique a localização do dado no mapa:'. The map shows a satellite view of a region with several green location markers. A red marker is highlighted, and a pop-up window displays its coordinates: 'Latitude: -21.3720' and 'Longitude: -46.1694'. Navigation buttons for the map are visible at the bottom.

Figura 2.27 - Verificação do dado parte 1.

Fonte: Os autores

Acima do mapa, existem botões que referenciam dados anteriores ou próximos da lista de dados do mesmo tipo como mostrada na Figura 2.27. Isso facilita

a navegação entre todos os dados sem que haja a necessidade de voltar para a lista sempre que deseja-se exibir outro dado.

Na Figura 2.28 temos a continuação da validação do dado onde são exibidas as mídias, que no caso são dois áudios e um vídeo.

O especialista deve verificar se cada uma das mídias realmente se referem ao dado apresentado podendo selecionar individualmente quais não serão excluídas e estarão nas galerias do dado validado.

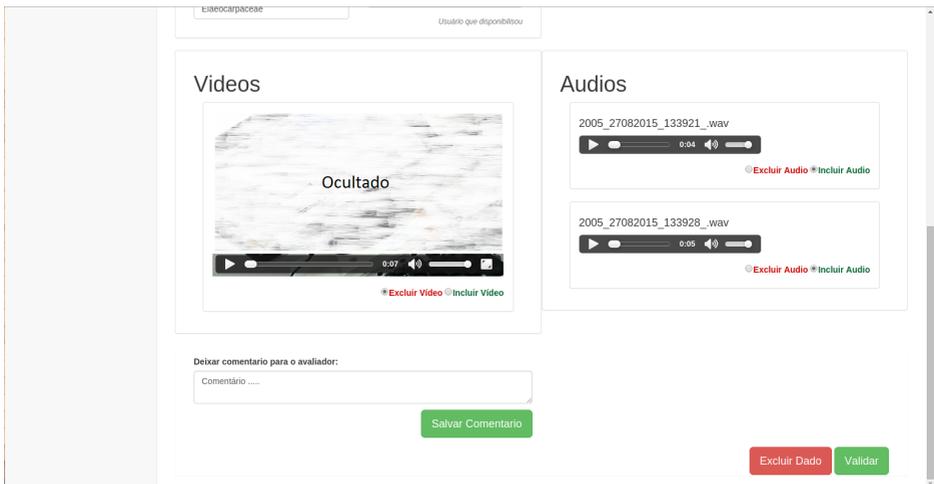


Figura 2.28 - Verificação do dado parte 2.

Fonte: Os autores

Para todos os usuários, mesmo os que não possuem permissão para o tipo do dado, podem deixar ao final da exibição dos atributos e mídias do dado um

comentário, que pode servir como lembrete ou conter informações relevantes na hora de avaliar se o dado está totalmente correto.

Ao final da exibição de todas as mídias e o comentário para o avaliador, os usuários que tiverem a permissão podem excluir o dado e todas as suas mídias ou, validar o dado e incluir as mídias que foram selecionadas para tal.

4.3 Avaliações de usabilidade

Duas avaliações de usabilidades foram realizadas para verificar possíveis melhorias e validar o DigiAtlas Mobile quanto a facilidade de uso e sua eficiência. As avaliações de usabilidade foram realizadas de forma qualitativa conforme Nielsen e Budiu (2014), utilizando-se do método do percurso cognitivo no qual o usuário explora a interface e tenta aprender por conta própria utilizar a aplicação (LEWS; WHARTON, 1997).

Além disso, utilizou-se da técnica de “pensar em voz alta” em que o usuário a todo momento diz em que está pensando e o que levou-o a pensar nisso (NIELSEN; LORANGER, 2007).

Definiram-se, previamente, ações para que os usuários executassem. Enquanto tentavam realizar tais

ações, foi observado o comportamento de cada um onde anotações foram feitas para análise futura.

Também, quando conveniente, perguntava-se em que o usuário tinha pensado e o por que, para assim encontrar defeitos e validar funcionalidades.

As avaliações foram realizadas com 5 meses de intervalo, sendo a primeira realizada no final do mês de abril de 2015 e a segunda no final do mês de setembro do mesmo ano. Assim, após a primeira avaliação, foram feitas modificações e correções que na segunda avaliação e puderam ser testadas.

A primeira delas foi realizada com docentes da Unifal-MG do curso de Ciências Biológicas. As ações a serem feitas na primeira avaliação e o nível de dificuldade classificado entre: muito fácil, fácil, intermediário, difícil e muito difícil, são apresentados a seguir no Quadro 2.1.

Os resultados do primeiro teste foram satisfatórios visto que a maioria das dificuldades apresentadas estavam classificadas como fácil e muito fácil. Algumas modificações foram realizadas ao notar que em algumas vezes os professores inicialmente tentaram realizar determinada tarefa de forma diferente, mas logo descobriram a forma correta.

Quadro 2.1 - Dificuldade por ação indicada pelos professores.

Ação	Prof. 1	Prof. 2	Prof. 3
Crie uma turma.	Muito fácil	Muito fácil	Muito fácil
Através da ferramenta pesquisa (lupa), encontre as localidades da espécie <i>Cariama cristata</i> (<i>C. cristata</i> ou <i>seriema</i>).	Fácil	Intermediário	Muito fácil
Abra a imagem e o áudio da espécie.	Fácil	Fácil	Muito fácil
Crie um marcador para essa nova localidade.	Intermediário	Muito fácil	Intermediário
Adicione o novo dado nesse marcador	Fácil	Muito fácil	Fácil
Edite a data e ocorrência do dado adicionado e faça logout.	Muito fácil	Intermediário	Fácil

Fonte: Os autores

Um exemplo disso são os ícones que demonstram quais mídias certo dado possui, como vimos na Figura 2.10. Os usuários tentaram acessar a galeria de imagem e áudio ao clicar nos ícones correspondentes, contudo foram direcionados para a tela de informação do dado, e em seguida conseguiram acesso a cada uma das galerias (FERREIRA et al., 2015). A aplicação agora permite o acesso às galerias a partir do toque nestes ícones.

Outra modificação realizada foi a retirada da

funcionalidade que adicionava um novo marcador nas coordenadas da localização do aparelho. Tal funcionalidade era especificada por um ícone também na barra superior, ao lado da operação "Adição".

Sua retirada deve-se ao fato de ter gerado confusão em dois dos três participantes do teste na hora de realizar a ação "Criar novo marcador".

Essa alteração não tira a possibilidade de localizar-se no mapa e em seguida adicionar um marcador obtendo o mesmo resultado, e também diminui as possibilidades de gerar confusão.

Ao concluírem as ações e indicarem o nível de dificuldade, os professores foram incentivados a utilizarem a aplicação por conta própria. Elogios foram feitos quanto a iconografia dos tipos existentes nos marcadores e também a possibilidade de Importação e Exportação.

Um dos professores sugeriu a união das funcionalidades de edição e remoção, o que tornaria o processo dessas operações mais intuitivo e reduziria o número de ícones na barra superior (FERREIRA et al., 2015). Essa alteração foi feita e como apresentado na seção de resultados, as duas operações funcionam em conjunto.

A segunda avaliação foi realizada com estudantes da graduação do curso de Geografia e Ciências Biológicas.

As ações realizadas nesta segunda avaliação estão descritas no Quadro 2.2 e possui os mesmos possíveis níveis de dificuldade do teste anterior.

Quadro 2.2 - Ações executadas no teste de usabilidade por estudantes da graduação.

Ação	Estudante 1	Estudante 2	Estudante 3
1- Crie uma turma.	Muito fácil	Muito fácil	Muito fácil
2.1- Através da ferramenta pesquisa (lupa), encontre as localidades (3) que a espécie <i>Cariama cristata</i> (<i>C. cristata</i> ou seriema) ocorre.	Fácil	Fácil	Intermediário
2.2- Abra a imagem e o áudio da espécie.	Muito fácil	Fácil	Intermediário
3.1- Crie um marcador para essa nova localidade.	Difícil	Fácil	Intermediário
3.2- Adicione o novo dado nesse marcador	Muito fácil	Fácil	Muito difícil
3.3- Edite a data e ocorrência do dado adicionado e faça logout.	Intermediário	Fácil	Muito difícil

Fonte: Os autores

A segunda avaliação mostrou resultados que indicaram maior dificuldade por parte dos estudantes ao tentarem realizar as ações, contudo mesmo assim a maioria dos resultados foram “Muito fácil” e “Fácil”, o que também indica que a aplicação não é complexa de se utilizar.

Algumas sugestões foram feitas também quanto a melhoria da funcionalidade de pesquisa, para que seja possível realizar buscas em campos sem a necessidade de acentuação exata. Também houve a sugestão de melhorias na funcionalidade de adição, entre elas, realçar os campos de preenchimento obrigatórios. No geral, os comentários dos estudantes foram positivos e satisfatórios quanto ao uso do aplicativo e as possíveis melhorias a serem realizadas.

5 Considerações finais

O uso das tecnologias digitais na educação, em especial os dispositivos móveis, pode potencializar os processos de ensino-aprendizagem, principalmente quando utilizados em propostas integradoras e motivadoras, buscando a ação e colaboração dos estudantes.

Uma boa aplicação dos dispositivos móveis se

relaciona com os trabalhos de campo (presenciais ou virtuais), uma vez que, por meio destes, disponibilizam-se instrumentos que facilitam o registro e visualização destes dos dados ambientais que poderão ser, posteriormente, trabalhos em sala pelos professores das diversas áreas.

Nesse contexto, o DigiAtlas Mobile pode potencializar o interesse dos estudantes no ensino multidisciplinar por meio de sua proposta lúdica e inovadora. Além disso, traz uma oportunidade para que o professor explore e seja criativo em novas práticas de ensino.

Até o momento, várias funcionalidades foram implementadas e foram verificadas quanto sua facilidade e eficiência por meio dos testes de usabilidade aplicados. Com os testes pode-se observar que a aplicação possui pouca complexidade quanto a sua utilização, o que é positivo pois tal complexidade poderia atrapalhar no ensino-aprendizado.

O DigiAtlas Web é o ponto chave para o crescimento da base de dados da aplicação mobile. Com ele será possível receber dados não-validados de qualquer usuário do DigiAtlas Mobile e verificá-los para que posteriormente quando forem disponibilizados aos

usuários, a integridade da base de dados seja mantida. Essa disponibilização gera o compartilhamento de dados entre os usuários e deixa a base de dados mais completa com dados de diversas regiões.

Por fim, o funcionamento em conjunto das duas aplicações mostra-se potencialmente forte para facilitar o ensino-aprendizado e pesquisas.

Agradecimentos

À Capes e Fapemig pelo apoio financeiro e bolsas, à Proex/ Unifal-MG, pelas bolsas de Extensão.

Referências

ANDROID DEVELOPER SDK. Disponível em: <<http://developer.android.com/intl/pt-br/develop/index.html>> Acesso em: 30 de novembro de 2015.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *ISO 9241-11* - Requisitos ergonômicos para o trabalho com dispositivos de interação visual. Parte 11: Orientações sobre usabilidade. Rio de Janeiro, 2002.

BLANCO. J. A.; UPTON, D. *CodeIgniter* 1.7. Birmingham: Packt Publishing Ltd., 2009.

CODEIGNITER BRASIL: MODEL, VIEW, CONTROLLER. Disponível em: <<http://codeigniterbrasil.com/passos-iniciais/mvc-model-view-controller>>. Acesso em: 30 nov. 2015.

CRUZ, F. *Scrum e PMBOK: unidos no gerenciamento de projetos*. Rio de Janeiro: Brasport, 2013.

DESENVOLVENDO um programa de cadastro usando SQLite e Adapters. Devmedia, 2013. Disponível em: <<http://www.devmedia.com.br/desenvolvendo-um-programa-de-cadastro-usando-sqlite-e-adapters/29329>> Acessado em 30 nov. 2015.

DISTIMO. The Mobile App Ecosystem: Global trends in established and growth markets. 2013. Disponível em: <http://www.distimo.com/download/publication/Distimo_Publication_MEF_Mobile_App_Ecosystem_-_September_2013/EN/archive> Acesso em: 14 de Maio de 2015.

DIX, A. et al. *Human-Computer Interaction*. 4th ed. Local: Prentice-Hall International, 2004.

DURELLI, V. H.; VIANA, M. C.; PENTEADO, R. A. Uma proposta de reuso de interface gráfica com o usuário baseada no padrão arquitetural MVC. In. SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO, 4, 2008, Rio de Janeiro-RJ. *Anais SBSI Rio de Janeiro, BDBCcomp*: 2008. p. 48-59.

DUSO, L.; BORGES, R. M. R. Mudança de Atitude de Estudantes do Ensino Médio a Partir de Um Projeto Interdisciplinar Sobre Temática Ambiental. *Alexandria Revista de Educação em Ciência E Tecnologia*, Florianópolis, p. 51-76, 2010.

ATLÂNTICA, SOS Mata; INPE. Atlas dos remanescentes florestais da Mata Atlântica: período 2012-2013. SOS Mata Atlântica, São Paulo, 2014.

FERREIRA, L. F., et al. DigiAtlas: Dispositivos Móveis Auxiliando o Ensino Multidisciplinar de Questões Ambientais. In. CONGRESSO BRASILEIRO DE INFORMÁTICA, 4., 2015; CONFERÊNCIA LATINO-AMERICANA DE OBJETOS E TECNOLOGIAS DE APRENDIZAGEM, 10., 2015, Maceio. Anais dos Workshops do Congresso Brasileiro de Informática na Educação Maceio, SBC: 2015. Disponível em: <<http://dx.doi.org/10.5753/cbie.wcbie.2015.693>>. Acesso em 10 nov. 2015.

GOOGLE MAPS. Disponível em: <<https://developers.google.com/maps/documentation/android/start?hl=pt-br>>. Acesso em: 30 nov. 2015.

LEWIS, C.; WHARTON, C. *Handbook of human-computer interaction*. Chapter 30-Cognitive Walkthroughs, 2nd edition. Oxford: Elsevier Science B. V., 1997. p. 717-730

MAÇAL, E.; et al. *Geomóvel: um aplicativo para auxílio a aulas de campo de geologia*. Anais do SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 24, 2003. Campinas, Unicamp, SBC: 2013. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2483>>. Acesso em 20 de jul. 2018.

MONTEIRO, J. B. *Google Android: crie aplicações para celulares e tablets*. Casa do Código. Casa do Código, 2014. Disponível em: <<https://www.casadocodigo.com.br/products/livro-android>>. Acesso em: 05 maio 2018.

MYSOIL APP. Disponível em: <<http://www.bgs.ac.uk/mysoil/>>. Acesso em: 25 maio 2015.

NET MARKET SHARE: market share statistics for internet technologies. Disponível em: <<https://netmarketshare.com/operating-system-market-share.aspx>>. Acesso em: 30 nov. 2015.

NET MARKET SHARE: market share statistics for internet technologies. 2018. Disponível em: <<https://netmarketshare.com/operating-system-market-share.aspx>>. Acesso em: 20 jul. 2018.

NEUCAMP, L. S., LOPES, G. N. *Aplicação de PHP, Javascript, CSS e MySQL na criação do Portal Agro@mbiente On-line*. Roraima: Agro@mbiente On-line, 2007. v. 1.

NIEDERAUER, J. *Desenvolvendo websites com PHP*. São Paulo: Novatec, 2004.

NIELSEN, J.; BUDIU, R. *Usabilidade Móvel*. Rio de Janeiro: Elsevier Editora Ltda, 2014.

NIELSEN, J.; LORANGER, H. *Usabilidade na web*. Rio de Janeiro: Elsevier Brasil. 2007.

PINTO, M. L. C. Trabalho de Campo e o processo de aprendizagem em busca de um método. *Espaços da Escola, O ensino da geografia na educação básica*. UNIJUÍ, Ijuí, ano 12, n. 47, p. 15-20. 2003.

QURESHI, M. R. J.; FATIMA S. *A comparison of model view controller and model view presenter*. CoRR abs/1408.5786, 2013. Disponível em: <<https://www.semanticscholar.org/paper/A-comparison-of-model-view-controller-and-model-View-Controller-and-Model-View-Presenter-Qureshi-Sabir/7c1630c0786a5c4968ca327db41b007401c59fd4>>. Acesso em 20 jul. 2018.

SCRUM. Disponível em: <<https://www.scrum.org/resources/what-is-scrum/>>. Acesso em: 23 fev. 2015.

SILVA, M. C.; FERREIRA, L.F.; BRESSAN, P.A.; RANIERO, M.; RAMOS, F. R.; CUNHA, R. G. T.; HASSUI, E.; SANTOS, C.A.; HORNINK, G. G. *DigiAtlas mobile, Versão 2*. 2018. Disponível em: <<https://play.google.com/store/apps/details?id=com.digiatlas>>. Acesso em: 06 set. 2018.

SILVA, M. C.; FERREIRA, L.F. ; BRESSAN, P.A. ; HORNINK, G. G. . *DigiAtlas Web. Versão 1*. 2018. Disponível em: <<http://www.desenvolvimentovirtual.com/digiatlas>>. Acesso em: 06 set. 2018.

SOARES, N. S. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. *Revista Eletrônica de Sistemas de Informação*. Curitiba, v. 3, n. 1, 2004.

SOUZA, R. R.; ALVARENGA, L. A Web Semântica e suas contribuições para a ciência da informação. *Ciência da Informação*, Brasília, v. 33, n. 1, p. 132-141, 2004.

PHP: Hypertext Preprocessor (PHP). Disponível em: <<https://secure.php.net/>>. Acesso em: 30 nov. 2015.

SUN MICROSYSTEMS Inc. Java BluePrints Model-View-Controller. 2002 Disponível em: <<https://www2.cs.duke.edu/courses/fall07/cps108/papers/sunmvc.pdf>>. Acesso em: 05 de maio de 2018.

Capítulo 3 O uso do Scratch como proposta multidisciplinar educacional

*Daniele de Almeida Dias
Gabriel Gerber Hornink*

A sociedade atual exige cada vez mais profissionais em constante formação. Para acompanhar essa exigência, são necessárias mudanças nas escolas, principalmente nos métodos de ensino. Deve-se permitir que o estudante construa seu próprio conhecimento de forma ativa e colaborativa. A informática, por meio de softwares educativos, é uma grande aliada nesse novo processo de aprendizagem. O Scratch se encaixa nessa proposta por ser um software construcionista. Trata-se de uma linguagem de programação visual, que permite a criação de projetos multimídia, tornando possível aprender conceitos específicos de programação e fazendo do estudante o construtor de seu próprio conhecimento. Considerando a defasagem do método de ensino atual e a importância que os computadores têm apresentado na educação, este artigo tem como objetivo expor os indícios obtidos, por meio de uma avaliação, de que o software Scratch pode ser utilizado como uma nova proposta multidisciplinar educacional.

1 Introdução

O avanço da tecnologia e as mudanças culturais e comportamentais, tornam cada vez mais difícil conseguir a atenção dos estudantes na sala de aula.

Livros e apostilas educativas, há séculos os instrumentos de ensino mais utilizadas no mundo, têm de competir com celulares, *tablets* e outros dispositivos.

No entanto, a dedicação às novas tecnologias pode se tornar uma aliada no aprimoramento da prática pedagógica e na criação de novas dinâmicas em sala de aula por meio da utilização de recursos tecnológicos facilmente encontrados em computadores, tais como vídeos, áudios, objetos de aprendizagem e conteúdos que incentivam a participação, a curiosidade e a interatividade entre estudantes e professores.

Com o intuito de facilitar a aprendizagem, criou-se uma nova linguagem de programação no *Media Lab* do Instituto de Tecnologia de Massachusetts, no ano de 2007. O Scratch, como é chamado, é muito simples e intuitivo e, por isso, é recomendado para crianças acima de 8 anos de idade e principiantes que estejam interessados no mundo da programação de computadores (MALONEY et al., 2010).

A facilidade dessa linguagem está na disposição dos comandos, organizados em blocos pré-definidos, o que evita erros sintáticos, que poderiam ser cometidos facilmente por programadores iniciantes. Seu funcionamento se dá por meio de ações *drag and drop* (arraste e solte) sobre esses blocos, que podem ser interligados da maneira que o usuário preferir, desde que haja lógica entre eles, dando sentido ao projeto. Os projetos Scratch podem se tornar jogos, histórias animadas, apresentações de conteúdos, entre outras funcionalidades (KELLEHER; PAUSCH, 2005; TANRIKULU; SCHAEFER, 2011).

A tecnologia e a educação caminham lado a lado e pesquisas comprovam que os computadores ajudam a aumentar o desempenho escolar e a motivação dos estudantes frente ao conteúdo dado. Moraes (1996) e Gladcheff, Zuffi e Silva (2001) garantem que o uso da informática traz benefícios para a concentração, a autonomia e o desenvolvimento cognitivo dos estudantes. Além disso, segundo Valente (1993), é possível representar, por meio do computador, inúmeros conceitos, possibilitando ao estudante adquirir conhecimentos interdisciplinares relevantes ao seu desenvolvimento intelectual.

O Mapa da Exclusão Digital (NERI, 2003) exemplifica esse contexto. Segundo o estudo, estudantes de 13 a 18 anos que tiveram acesso a computadores apresentaram um aumento de 17% no desempenho de provas de matemática em relação àqueles sem acesso a essa tecnologia e que, portanto, não tinham o auxílio de ferramentas que proporcionassem mais fontes de pesquisa e estudo, que facilitassem a compreensão dos conteúdos por meio de instrumentos educacionais diferenciados.

Com base nisso, este trabalho apresenta como proposta a avaliação do Scratch a fim de ter indícios da viabilidade de uso em sala de aula, inovando a prática pedagógica ao criar dinâmicas escolares, que promovam a interatividade e incentivem a curiosidade do estudante.

1.1 O contexto educacional

No contexto atual, desde o fim da Segunda Guerra Mundial, o mundo vem sofrendo drásticas mudanças nos âmbitos político, ambiental, econômico, científico e tecnológico. A educação, inclusive, mudou rapidamente nesse período, se comparado a períodos anteriores. Porém, segundo Coombs (1976, apud PATTO, 1997, p.21), os sistemas de ensino

[...]têm-se adaptado muito vagarosamente ao ritmo mais veloz dos acontecimentos que os rodeiam. O conseqüente desajustamento – que tem assumido as mais variadas formas – entre os sistemas de ensino e o meio a que pertencem constitui a essência da crise mundial da educação.

Duas possíveis razões, expostas por Coombs (1976, apud PATTO, 1997), para a crise na educação são: o atraso da sociedade devido aos costumes religiosos, às atitudes antiquadas da sociedade e às estruturas institucionais; e o atraso dos sistemas de ensino. Essas e outras razões, segundo o autor, não permitem que a população eleve seu nível de escolaridade ou atenda à demanda crescente de mão de obra especializada, indispensável para o desenvolvimento do país.

Segundo Moraes (1996), há mais de 300 anos o paradigma tradicional de ensino tem sido influenciado pelo positivismo, uma corrente filosófica decorrente de associações de pensamentos da Revolução Francesa e do Iluminismo. Esse paradigma é caracterizado por dividir o conhecimento em assuntos e especialidades em uma escola centrada no professor e na transmissão de seu conhecimento. Trata-se de um ensino mecânico, no qual exige-se memorização e repetição e que recompensa o

conformismo e a boa conduta, punindo os erros e as tentativas de liberdade de expressão.

Com o passar das décadas, cientistas como Einstein, Heisenberg, Niels Bohr e Prigogine, por meio de descobertas como a Teoria da Relatividade, a Teoria Quântica, o Princípio da Incerteza e a Lei da Complementaridade, propuseram que o mundo passasse a ser concebido em “termos de movimento, fluxo de energia e processo de mudança” (MORAES, 1996).

Essas descobertas deixaram de fundamentar a objetividade científica da velha ciência positivista – a qual, segundo Silvino (2007), considera os fatos empíricos como a única base do verdadeiro conhecimento – e, dessa forma, passaram a corroborar um novo paradigma, o paradigma emergente, caracterizado por enfatizar a percepção de consonância entre todos os acontecimentos físicos, biológicos, psicológicos, socioculturais e, também, educacionais, que extrapolam as fronteiras disciplinares e conceituais (MORAES, 1996).

Dessa forma Behrens (1999) propõe a reformulação da escola, a fim de sustentar essas novas fronteiras, tornando-a um espaço de trabalho colaborativo, onde a prática pedagógica possibilite a

construção de conhecimento. Ao mesmo tempo que o professor, nesse contexto, deixa de ser a única fonte de informação e passa a ser um especialista no processo de aprendizagem, diante de uma educação que prioriza o desenvolvimento de habilidades, competências, inteligências e valores (ROCHA, 2008).

Assim, a escola é capaz de atender às exigências da sociedade, que requisita, no futuro, profissionais em constante formação, autônomos, proativos e com boa relação interpessoal (BEHRENS, 1999).

No entanto, segundo Melo Neto (2007), a educação não tem autonomia para acompanhar o desenvolvimento social e tecnológico. Tal autonomia só será alcançada com uma nova postura pedagógica, que ainda está muito distante do cotidiano escolar.

Uma maneira de inovar as ações pedagógicas é inserir novos ambientes de aprendizagem informatizados, que abrirão espaço aos novos métodos de ensino e conhecimento.

O uso da informática pode aumentar a concentração, a autonomia, o desenvolvimento cognitivo dos estudantes e o aprendizado por meio dos seus próprios erros, uma vez que pode ser adaptado ao ritmo de cada um (GLADCHEFF; ZUFFI; SILVA, 2001;

MORAES, 1996).

Além disso, o computador permite uma ampla gama de representações, o que possibilita ao estudante adquirir conhecimento de diversas áreas relevantes ao seu desenvolvimento intelectual (VALENTE, 1993).

Porém, é preciso cautela ao dispor dessa ferramenta de ensino. Para utilizá-la, Valente (1993) enfatiza que é necessário um software educativo e um professor qualificado para adotar esse software de maneira adequada.

Ainda, para o bom uso, segundo Gladcheff, Zuffi e Silva (2001), o professor deve definir objetivos e a metodologia que serão utilizados na concepção do conhecimento para, então, escolher o software educativo que atenda a esses propósitos.

Por consequência, deve haver investimento direcionado à capacitação dos professores para o uso da tecnologia como ferramenta educacional e seu emprego como meio de aprendizagem.

Assim, os professores não ficarão atados somente à informática básica. Esse incentivo fará com que o professor se torne apto a utilizar qualquer tecnologia que tenha esse fim, instigando a independência e a busca de aprendizado (MELO NETO, 2007).

Valente (1993) classifica o uso do computador para fins educacionais em duas categorias: computador como máquina de ensinar e como ferramenta educacional.

O uso como máquina de ensinar consiste em um paradigma chamado instrucionista, a informatização dos métodos de ensino tradicionais. Já o computador como ferramenta educacional se caracteriza como paradigma construcionista, abordagem na qual o estudante constrói seu conhecimento utilizando-se do computador.

Sendo assim, o instrucionismo não possui uma prática pedagógica inovadora. Esse paradigma nada mais é que o positivismo informatizado. Diferentemente do construcionismo, que inova em sua metodologia, fazendo com que o estudante fundamente seu aprendizado na construção do conhecimento por meio de softwares que permitam o crescimento. Por isso, é tão importante que o professor saiba ponderar sobre os inúmeros softwares educativos disponíveis, pois o simples fato de utilizar a informática na educação não caracteriza uma prática inovadora (BEHRENS,1999).

Alguns exemplos de softwares instrucionistas são aqueles cujo o ensino é de abordagem tutorial, por meio de lições e exercícios, por meio de jogos ou, também, por meio de simulação. Já no paradigma construcionista

a aprendizagem ocorre por meio de “elaboração de textos [...], resolução de problemas de diversos domínios do conhecimento e representação desta resolução segundo uma linguagem de programação”, entre outras (VALENTE, 1993).

Representar a resolução de um problema em uma linguagem de programação quer dizer sistematizar essa resolução através de raciocínio lógico. O raciocínio lógico é muito desenvolvido durante o aprendizado de qualquer linguagem de programação.

Além disso, Resnick et al., (2009) afirmam que a programação expande a quantidade de assuntos que se pode aprender e o aprendiz se torna mais criativo, bem como aprende a refletir sobre seus próprios pensamentos.

Utilizando a programação, pode-se adquirir a fluência digital, que é a capacidade de desenhar, criar e inventar um novo projeto multimídia (RESNICK et al., 2009).

Uma maneira de representar a resolução de um problema em linguagem de programação é por meio do incentivo ao estudante, por parte do professor, para a construção de projetos.

De acordo com as Orientações Curriculares para o

Ensino Médio (OCEM), publicadas em 2007,

[...]o ensino por meio de projetos, além de consolidar a aprendizagem, contribui para a formação de hábitos e atitudes, e para a aquisição de princípios, conceitos ou estratégias que podem ser generalizados para situações alheias à vida escolar[...] (OCEM, 2007, p. 27).

O professor deve estabelecer metas educativas e de aprendizagem que devem ser alcançadas por meio dos projetos, além de escolher os conteúdos conceituais e preparar as atividades. Com isso, as aulas se tornam mais investigativas e interativas e o estudante é capaz de consolidar o que aprendeu, construindo seu próprio conhecimento e ampliando seu raciocínio (OCEM, 2007).

1.2 Conhecendo o Scratch

Projetado desde 2003 e lançado oficialmente em 2007 pelo Media Lab do Instituto de Tecnologia de Massachusetts (MIT), o Scratch é um software que permite que os usuários criem seus próprios projetos utilizando recursos interativos, sonoros e visuais, ao mesmo tempo em que aprendem programação de computadores.

Por meio do Scratch é possível criar conteúdos multimídia dos tipos: histórias animadas, jogos, projetos científicos, tutoriais, entre outros (MALONEY et al.,

2010).

Aprender a programar pode ser muito difícil para os estudantes iniciantes (KELLEHER; PAUSCH, 2005). É preciso identificar a estrutura e a lógica de programação de um problema para resolvê-lo por meio de uma linguagem. Concomitantemente, o estudante deve ficar atento aos detalhes técnicos que não estão relacionados à resolução deste problema (TANRIKULU; SCHAEFER, 2011).

O grande diferencial do Scratch é omitir a parte técnica, utilizando um paradigma de programação visual, pois, assim, o estudante não necessita manter o foco em aprender a sintaxe da linguagem que está sendo utilizada (MALONEY et al., 2010).

Este diferencial é importante, tendo em vista que os erros são comuns nessa fase do aprendizado e acabam desencorajando esses estudantes (KELLEHER; PAUSCH, 2005).

Este paradigma deixa a criação do projeto livre de erros sintáticos e permite que o usuário se concentre somente na lógica de programação, pois ele não tem que escrever código algum nem lidar com as peculiaridades de cada linguagem (TANRIKULU; SCHAEFER, 2011).

O Scratch foi especialmente desenvolvido para um

público-alvo de crianças e jovens entre 8 e 16 anos. Por isso, o software foi projetado de acordo com os interesses desse público e, principalmente, para aqueles que não tiveram contato prévio com programação. Além disso, é uma ferramenta muito acessível, pois está disponível em mais de 50 idiomas e é disponibilizada gratuitamente no website (MALONEY et al., 2010).

No Scratch existem as seguintes ferramentas: o palco, os atores, os roteiros, as fantasias e os sons.

O palco representa o cenário do projeto que pode mudar conforme a interação.

Os atores, também chamados de *sprites*, são os objetos interativos que atuam conforme os comandos dados pelo usuário e possuem as fantasias, que são diferentes aparências de um mesmo objeto.

Os roteiros são os blocos de instruções, separados por categorias, como, por exemplo, movimento, aparência e eventos. E, por fim, os sons, que são acionados por meio dos blocos de instruções.

Todas essas cinco ferramentas são personalizáveis, ou seja, é possível criar, importar ou editar as que já estão disponíveis para o usuário por meio das áreas de edição presentes no próprio software. O layout do software está representado na Figura 3.1.

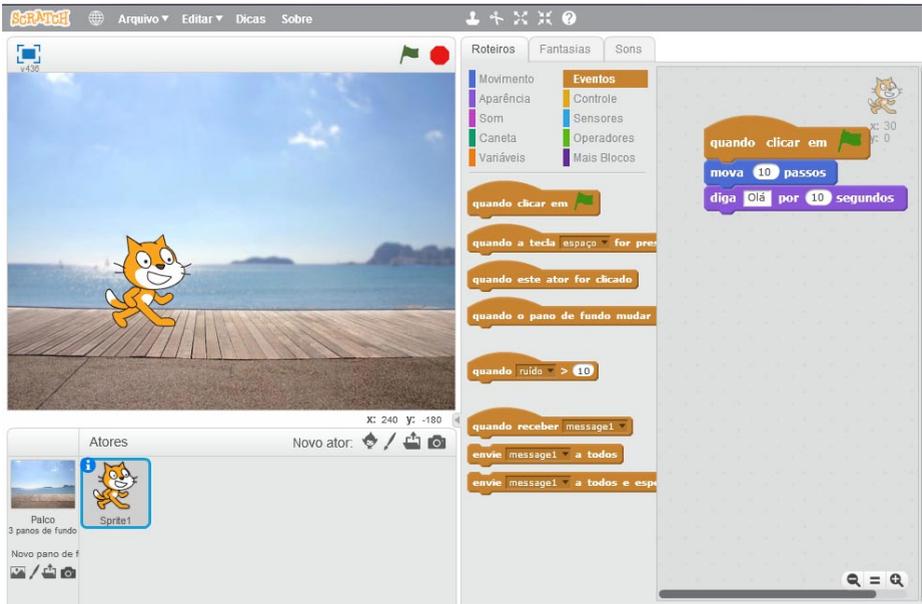


Figura 3.1 - Interface do software Scratch

Fonte: Tela do Scratch captura pelos autores.

Observa-se na lateral esquerda da Figura 3.1 o "palco", onde pode-se manipular e visualizar os componentes visuais; na parte central as cores agrupando semanticamente os comandos (blocos coloridos de encaixe e as abas para mudança dos menus (roteiros/ comandos, fantasias e sons); e, na lateral direita, a área onde os blocos são combinados para criar o código do Scratch.

Para criar um código no Scratch, é necessário arrastar os blocos de instruções para a área de comando do sprite correspondente. Os blocos se encaixam somente se apresentarem compatibilidade entre eles, ou

seja, se houver uma estrutura lógica, para, então, formar um algoritmo.

Para Voelcker, Fagundes e Sidel (2008), por meio do Scratch é possível aprender conceitos específicos de programação, tais como: “sequência, iteração, condição, variáveis, execução paralela, sincronia, interação em tempo real, lógica booleana, números randômicos, tratamento de evento e criação de interfaces”.

Segundo Silva e Tavares (2012), construir projetos no Scratch faz com que o estudante deixe de ser um consumidor passivo e passe a ser construtor de seu próprio conhecimento.

Sendo assim, por essas características, podemos concluir que o Scratch é uma ferramenta construcionista (MALONEY et al., 2010).

Além da possibilidade de construir seu próprio projeto, é possível compartilhá-lo no *website* (Figura 3.2) do Scratch (www.scratch.mit.edu), receber *feedback* e incentivo de outros usuários, acessar e aprender mais com os projetos compartilhados, assim como remixar projetos existentes (MALONEY et al., 2010).

O sistema do Scratch cria uma árvore de remixagem, indicando a linha de origem e reconstrução dos aplicativos, garantindo a transparência da autoria.

Crie histórias, jogos e animações
Compartilhe com pessoas do mundo todo



Uma comunidade de aprendizado criativo com **9.635.180** projetos compartilhados

[SOBRE O SCRATCH](#) | [PARA EDUCADORES](#) | [PARA OS PAIS](#)

Projetos em Destaque

How I recover from ...
por cinnabungrit3

Better Pen Thing
por PuluJosh

A Guide Through La...
por xXKawaii_GirLxX

Beach Break!
por FairyNinja

Vany Dress Up Gam...
por xVanyx

Estúdios em Destaque

APIs

Media Library!

Virtual Architecture

Food Puns!

Projetos Administrados por Malik44 [Aprenda mais](#)

Figura 3.2 - Interface do website Scratch.

Fonte: Tela do website do Scratch captura pelos autores.

1.3 Experiências no ensino

Em uma escola estadual de Belém – PA foi elaborado por Sousa (2013) um curso de introdução ao raciocínio lógico e do desenvolvimento de algoritmos tendo como público-alvo os estudantes do ensino médio. O curso foi ministrado por estudantes do Programa Institucional de Bolsa de Iniciação à Docência (PIBID) da

graduação em Licenciatura em Computação da Universidade Federal Rural da Amazônia (UFRA).

Os objetivos principais do curso foram propor a colaboração no processo de ensino e aprendizagem de estudantes do curso de licenciatura e fazer que os estudantes do ensino médio se aproximassem do ensino superior. O curso teve carga horária de total de 40 horas e 28 estudantes puderam concluí-lo. Os resultados foram satisfatórios, pois, além de comprovar ser possível ensinar algoritmos no ensino médio, foi possível concluir que o curso proporcionou, realmente, a proximidade com o ensino superior.

Realizou-se um estudo análogo ao anterior, no Vale do Mamanguape da Paraíba, uma oficina de programação em Python proposta por Marques et al., (2011) a fim de atrair e aumentar o interesse para a área de computação dos estudantes do ensino médio desta região. A oficina teve duração de 16 horas e a participação de 16 estudantes. Por meio dos resultados obtidos pode-se concluir que a oficina atingiu seu propósito de aumentar o interesse dos estudantes pela área de informática e a maioria deles declarou que faria algum curso na graduação relacionado a essa área.

Outra intervenção interdisciplinar, mas agora por

meio do Scratch, ocorreu em Divinópolis – MG, onde Silva e Tavares (2012) utilizaram o computador como desafiador de conflitos cognitivos. O objetivo principal era vincular a linguagem de programação Scratch e a Literatura de Cordel. Essa experiência foi realizada com estudantes do 5º ano do ensino fundamental da rede pública municipal da cidade. Após a intervenção, pode-se observar que houve uma melhora no desempenho lógico-matemático e a ampliação do letramento.

Este trabalho difere dos anteriores, pois visa avaliar o Scratch como uma proposta multidisciplinar educacional e, também, avaliar a familiaridade que os estudantes tiveram com a linguagem ao longo do curso, levando em conta o contato prévio que cada um teve com o computador.

2 Procedimentos

O método usado neste trabalho tem como fundamento uma abordagem qualitativa, apesar de possuir dados quantitativos, a respeito do curso de Introdução ao Software Scratch.

Segundo Dalfovo, Lana e Silveira (2008), as pesquisas que se denominam qualitativas ou quantitativas são caracterizadas por possuírem um

problema específico e adotarem a utilização de questionários e de entrevistas, diferindo apenas em como os dados serão interpretados. As pesquisas quantitativas têm “a intenção de garantir a precisão dos trabalhos realizados, conduzindo a um resultado com poucas chances de distorções”, e as pesquisas qualitativas “não têm um significado preciso em quaisquer das áreas onde sejam utilizados [...] e identificam-se com a observação participante” (DALFOVO; LANA; SILVEIRA, 2008, p. 7).

Este trabalho foi dividido entre as seguintes etapas: montagem e aplicação do curso de extensão *Introdução ao Software Scratch*, incluindo a elaboração dos questionários, das atividades e do projeto final dados em sala e a avaliação das atividades; e análise das respostas obtidas nos questionários.

2.1 Construção e aplicação do curso

O curso foi oferecido na Universidade Federal de Alfenas (UNIFAL-MG), por meio do projeto Jovens Talentos, com o professor Luiz Eduardo da Silva como professor responsável e os discentes Romário Borges, bolsista do projeto, e Daniele de Almeida Dias, discentes do curso de Ciência da Computação, como ministrantes.

As inscrições foram abertas à comunidade da cidade de Alfenas, podendo ser efetuadas gratuitamente através do Controle de Ações de Extensão (CAEX) da UNIFAL-MG, com limite de 25 vagas.

2.1.1 Participantes

Das 25 vagas ofertadas, todas foram requisitadas até o fim das inscrições. Porém, no primeiro dia de aula, compareceram apenas 18 estudantes. A faixa etária desses estudantes está representada na Figura 3.3. Dos 18 estudantes presentes no primeiro dia de aula, 14 concluíram o curso (cerca de 78%). Entre os concluintes, nove frequentam curso técnico ou superior e um é professor universitário.

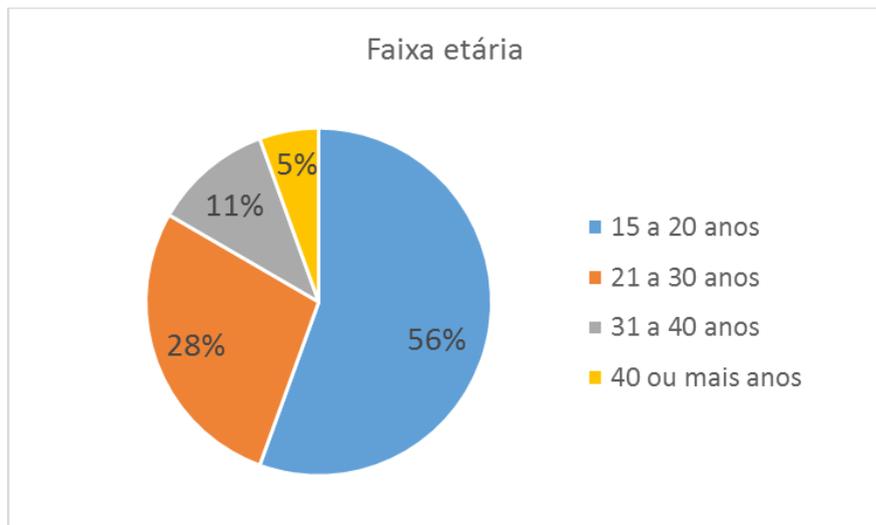


Figura 3.3 - Faixa etária dos estudantes

Fonte: Os autores

2.1.2 Aulas e atividades

A carga horária do curso foi de 15 horas, contando com dez horas presenciais, distribuídas em três dias (Aula 1, Aula 2 e Aula 3), e cinco horas de atividades extras. Durante o curso, propôs-se a atuação do professor como facilitador do aprendizado. Além disso, recursos didáticos foram utilizados para desenvolver as capacidades de abstração, de raciocínio lógico, de resolução de problemas e de autonomia cognitiva do estudante.

No primeiro dia de aula, procurou-se contextualizar o software, seu objetivo e suas propostas. Disponibilizou-se o material complementar e ensinado como outros recursos, como o próprio site da ferramenta e outros sites relacionados, podem facilmente ser encontrados pela Internet. Visando agregar valor ao conteúdo escolar, ao fim dessa introdução, apresentaram-se alguns projetos feitos em Scratch que poderiam auxiliar no aprendizado em sala de aula. Então, a etapa de atividades práticas foi iniciada.

A primeira atividade prática realizada na primeira aula (Aula 1) está representada nas Figura 3.4 e 3.5 e a segunda atividade na Figura 3.6. Para familiarizar os estudantes com o conceito de programação, esses

exercícios foram resolvidos primeiramente na lousa por meio de comandos em linguagem natural com a participação ativa dos estudantes e sem a intervenção direta do professor.

Após o pseudocódigo de cada exercício ser escrito na lousa, os estudantes tiveram que implementá-los em linguagem Scratch.

Exercício 1

- Imagine a cena:

Um gatinho está andando todo pensativo pela rua, quando alguém passa e o cumprimenta. Como ele é educado, responde o cumprimento com “Olá!”

Vamos pensar em como programar essa situação?

Figura 3.4 - Enunciado do exercício 1 da Aula 1.

Fonte: Os autores

Primeiros passos

- A interação “cumprimentar o gatinho” será representada pela tecla “a”;
- O gatinho está pensando “Hmmm...”, enquanto ninguém o cumprimenta;
- Quando alguém o cumprimenta, ele diz “Olá!” durante 2 segundos.

Figura 3.5 - Considerações do exercício 1 da Aula 1

Fonte: Os autores

Exercício 2

- Faça um programa que receba dois números do usuário e depois mostre o resultado de sua subtração

Figura 3.6 - Enunciado do exercício 2 da Aula 1.

Fonte: Os autores

Por fim, outra atividade (Figura 3.7) foi implementada em forma de tutorial, ou seja, o professor falou, passo a passo, os comandos necessários para construir um projeto mais elaborado.

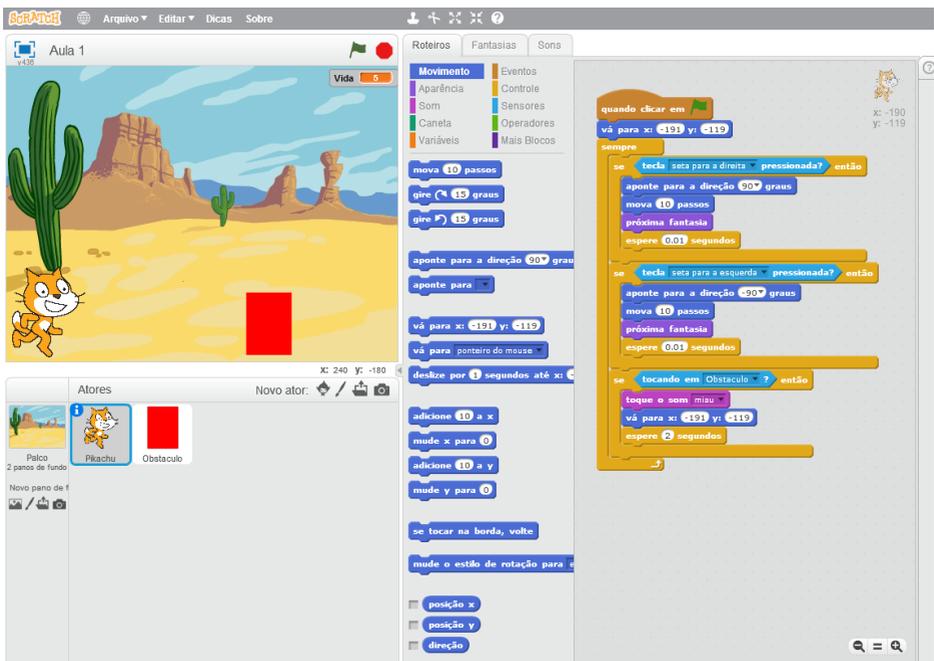


Figura 7 - Exercício 3 da Aula 1.

Fonte: Os autores

Para esta atividade (Figura 3.7), utilizaram-se vários conceitos de programação, como laços de repetição e condicionais. Além disso, gerou maior familiaridade com o software, por exemplo, importando sprites, fundos de tela e colisão/choque entre sprites.

A Aula 2 foi inteiramente prática, uma vez que os estudantes estavam familiarizados com a ferramenta, viabilizando aumentar a dificuldade da atividade.

De início, os estudantes tiveram que montar uma calculadora simples, utilizando os operadores matemáticos disponíveis na ferramenta e, por fim, programaram o jogo PacMan (Figura 3.8)

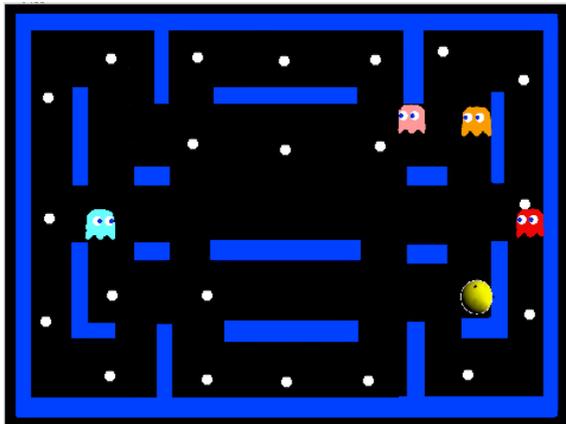


Figura 3.8 - Palco do PacMan utilizado na Aula 2.
Fonte: Os autores

Parte de uma das possíveis implementações do código está representada na Figura 3.8.

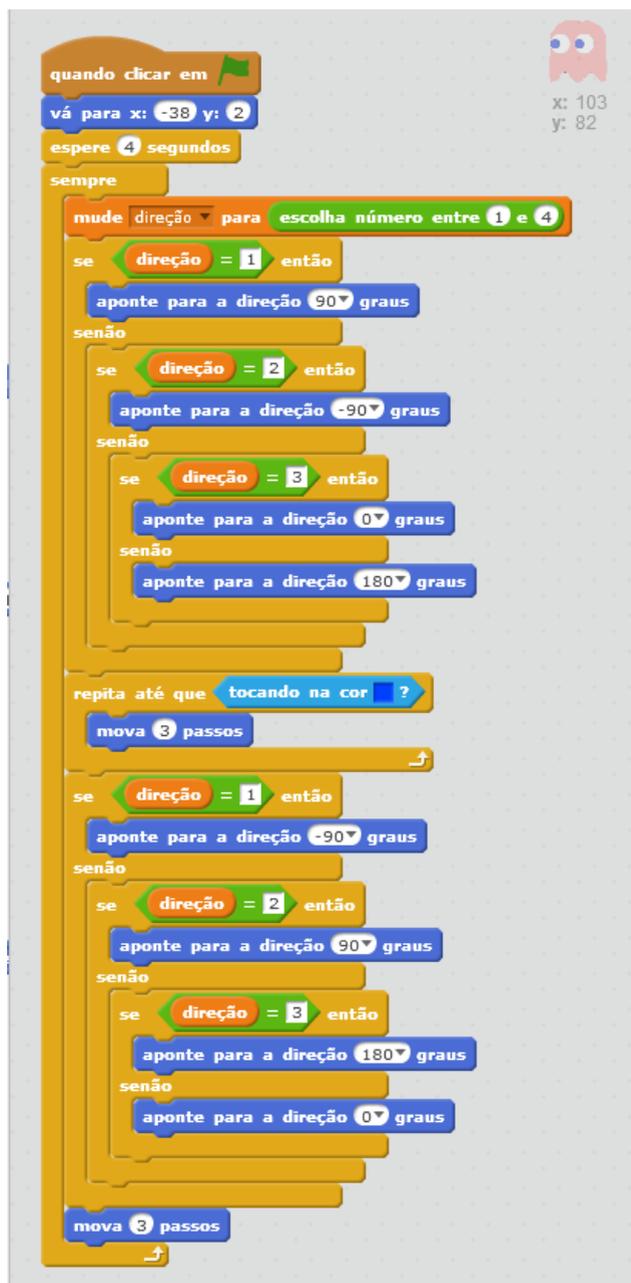


Figura 3.9 - Parte da implementação do PacMan utilizado na Aula 2.

Fonte: Os autores

Ao fim da Aula 2, os professores apresentaram a proposta do trabalho prático final, na qual os estudantes, em duplas ou individualmente, programaram projetos de tema livre ou relacionados ao seu dia-a-dia.

O projeto final teve alguns requisitos obrigatórios, que podem ser vistos no Apêndice 3.A. Vale destacar que a entrega de cada projeto foi, obrigatoriamente, compartilhada no site do Scratch, para que outras pessoas pudessem ter acesso.

Finalmente, na Aula 3, os estudantes tiveram que apresentar para todos os colegas do curso seus projetos finais. Para que houvesse maior estímulo entre os estudantes, propôs-se uma competição, na qual eles avaliaram todos os trabalhos em uma escala de zero a cinco, durante as apresentações.

Concomitantemente, os professores observaram se cada projeto obedeceu aos requisitos propostos e os avaliaram conforme o cumprimento desses requisitos, penalizando os trabalhos quando um requisito não fosse cumprido. As notas finais de cada projeto foram formadas pela média entre a nota dada pelo professor e das notas dadas pelos estudantes.

2.1.3 Questionário Inicial e Final

Com o objetivo de confirmar a hipótese a respeito do uso do Scratch como proposta multidisciplinar educacional, foram elaborados dois questionários, sendo um aplicado no início e o outro, ao término do curso.

O questionário inicial (QI) avaliou o contato que os estudantes haviam tido, até então, com o uso de computadores e com programação de computadores.

O questionário final (QF) abordou questões sobre usabilidade e sobre a viabilidade do emprego desse software como ferramenta educacional. O QI e o QF encontram-se nos Apêndices 3.B e 3.C respectivamente.

3 Resultados

Durante as aulas, por meio da observação participante dos professores e enquanto as atividades propostas eram executadas, pode-se perceber que os estudantes que conseguiam executar as atividades com mais facilidade procuravam ajudar os demais estudantes que obtiveram uma maior dificuldade. Todos trabalharam em equipe a favor de resolver os problemas que surgiam durante as atividades.

A cada atividade concluída, apesar do grau de dificuldade ser maior, os estudantes mostraram-se

bastante empolgados em resolver os problemas. A proposta de programar o jogo PacMan se tornou a mais motivadora entre todas as outras atividades, pois trata-se de um jogo bastante popular, mesmo para as gerações mais novas.

Para estimar melhor o conhecimento dos estudantes, foi aplicado, no primeiro dia de aula, um questionário para avaliar o contato prévio com o computador e com linguagens de programação.

Além disso, mesmo que alguns estudantes não estivessem familiarizados com programação de computadores, procurou-se avaliar, por meio do questionário, o pensamento lógico de cada um por meio de duas questões, nas quais os estudantes deveriam interpretar o pseudocódigo dado e, assim, encontrar a solução entre as alternativas (Figura 3.31 e Figura 3.32 do Apêndice 3.B).

A partir das perguntas sobre a motivação com relação ao curso (pergunta 5) e expectativas de aprendizagem (pergunta 6), os estudantes puderam expressar importantes informações que gerou um panorama dos envolvidos (Quadro 3.3).

Quadro 3.3 - Respostas das perguntas 5 e 6 (QI).

Part.	Pergunta 5	Pergunta 6
1	Ser independente quanto ao programa.	Preparar material didático.
2	Espero aprender a criar (eu mesma) atividades lúdicas que poderão facilitar de alguma forma o aprendizado dos meus futuros estudantes, principalmente aqueles que apresentarem uma maior facilidade de aprendizagem por meio da tecnologia.	Minha motivação é poder adquirir conhecimentos que poderei utilizar futuramente em minha profissão como professora e pedagoga e assim propiciar uma maior aprendizagem para meus estudantes.
3	Resposta nula.	Conhecer novas possibilidades para auxiliar em minhas pesquisas/estudos.
4	Os mecanismos básicos, ou seja, uma base para futuramente ter como aprofundar.	Será o "ponta-pé" inicial para o meu projeto de TCC.
5	Elementos básicos que satisfaçam a demanda do curso.	Requisito do Mestrado em História Ibérica.
6	A utilizar a ferramenta Scratch.	Quero aprender a utilizar mais recursos como meios educacionais.
7	Como manusear e como utilizar a ferramenta.	Utilizar com os estudantes.
8	Uma nova linguagem para o melhor aprendizado da organização em programação orientada a objetos.	Aprendizado.
9	Conhecer a ferramenta e aprender a utilizá-la.	Aprender novos meios de programação
10	Programação rápida e fácil usando scratch.	Aprender usar uma nova ferramenta de programação.
11	Melhores técnicas de programação.	Melhorar meu desenvolvimento em programação e aprender mais.

12	Melhores técnicas de programação.	Melhorar e aprender sobre novos meios de programação.
13	Espero que eu adquira conhecimentos sobre o "Scratch" de forma fácil e simples.	Minha motivação é muito grande, espero que com esse curso, eu aprenda mais sobre programação e sobre o "Scratch".
14	Aprender mais sobre o Scratch e programação.	Pretendo cursar algo voltado à computação e programação. Mas sei pouca coisa, e não tenho certeza se é realmente isso que quero. Por isso, me interessei em participar para aprender mais e, quem sabe, continuar programando.
15	Espero sair daqui entendendo realmente o que é o "Scratch" e melhorar a nossa lógica de programação.	A minha motivação é ter a oportunidade que vocês nos concederam para que possamos ampliar o nosso conhecimento na área de Informática e aproveitá-la da melhor forma possível.
16	Adquirir conhecimentos na área de programação e aprender mais sobre as lógicas computacionais. Essa é uma área que eu gosto muito por isso espero que esse curso me ajude ainda mais a ter certeza disso.	Minha maior motivação é a área de informática. Me interessa muito pela programação e por todo esse conteúdo. Pretendo aprender mais das lógicas e adquirir conhecimentos mais profundos.
17	Muitas coisas diferentes que não conheço e que pretendo aprender.	Aprender a programar e fazer coisas diferentes que não sabia.
18	Como desenvolver, programar, e fazer animações.	Crescer profissionalmente e ter uma boa carreira.

Part.: Participante

Fonte: Os autores

O mesmo ocorreu no questionário final, porém, perguntou-se sugestões de melhoria e quais aspectos positivos os estudantes manteriam no curso (Quadro 3.4 no Apêndice E).

No Quadro 3, pode-se observar que algumas pessoas disseram que a motivação em participar do curso incluía a possibilidade de inserir o Scratch como metodologia de ensino em seus papéis como professores. Enquanto outros, gostariam de aprender a programar ou aperfeiçoar suas técnicas como programadores.

No Quadro 3.4, a maioria dos estudantes afirmaram estarem satisfeitos com o curso e, como aspectos a serem melhorados, alguns estudantes disseram que poderia aumentar a carga horária, pois o aproveitamento seria maior.

Quadro 3.4 - Respostas das perguntas 3 e 4 (QF).

Part.	Pergunta 3	Pergunta 4
1	Nada. Foi muito bom tudo que foi apresentado.	Todos.
2	Quanto ao minicurso não percebo nada a ser acrescentado. Foi muito bem ministrado.	Todos.
3	Disponibilizar vídeo-aulas.	Todos adotados pelos ministrantes.
4	Mais testes de desempenho.	As aulas interativas e ótimas explicações.
5	Acho que deveria ser mais longo, pois em somente três aulas pode-se	Tudo em si. Gostei dos projetos

	explorar pouco do que o programa tem a oferecer. Deveria ser um curso mais longo.	elaborados em sala, dos que tinham que ser feitos em casa, a didática dos professores. Gostei bastante do curso, eu adoraria mais desses.
6	Sinceramente, gostei muito do curso. Não sei bem o que poderia melhorar, pois foi bem explicado. Poderia ter mais testes e trabalhos, mas está ótimo.	Explicação, a assistência quando se perde no conteúdo ou não entende o que aconteceu. Sem enrolações, bem prático e eficiente.
7	O curso poderia aprofundar mais os conhecimentos na área de jogos no Scratch. Quanto ao resto foi maravilhoso.	O curso tem um ótimo horário, e sendo uma vez por semana. Todos os ministrantes são muito atenciosos e inteligentes.
8	Uma nova tarefa de um programa que tivesse um bug e todos tentassem resolver. Assim, todos trabalhariam em prol de resolver esse bug. Quanto ao resto foi ótimo.	A organização, a vontade de trabalho, a criatividade, etc.
9	Na realidade, não acho que tenha que melhorar nada, apesar de que o curso seja por um pequeno período de tempo, é um pouco complicado entender tudo o que se pede e conseguir acompanhar o raciocínio dos professores.	A dinâmica da aula é bastante interessante e sempre que precisamos sabemos que podemos contar com a ajuda dos professores durante a aula e até mesmo esclarecer dúvidas através do e-mail.
10	Eu acho que nada, pois tudo o que precisa já está excelente porque é um bom horário, professores	Horário. Educação dos professores e a paciência.

	educados, atenciosos e boa didática.	Explicação clara.
11	Uma avaliação prática testando o entendimento e criatividade de cada um em sala e um aumento na carga horária do curso.	Toda a matéria foi explicada e passada de maneira prática e eficaz.
12	Nada vem em mente neste momento.	Carga Horária e didática.
13	Creio que nada.	A didática do minicurso.
14	O curso poderia ter sido aplicado em mais dias, achei um curso muito bom para ser aplicado em tão poucos dias.	A dedicação e a atenção dos professores.

Part.: Participante

Fonte: Os autores

Analisando as outras questões, observou-se que mais da metade dos respondentes usam o computador desde a adolescência ou infância (Figura 3.10).

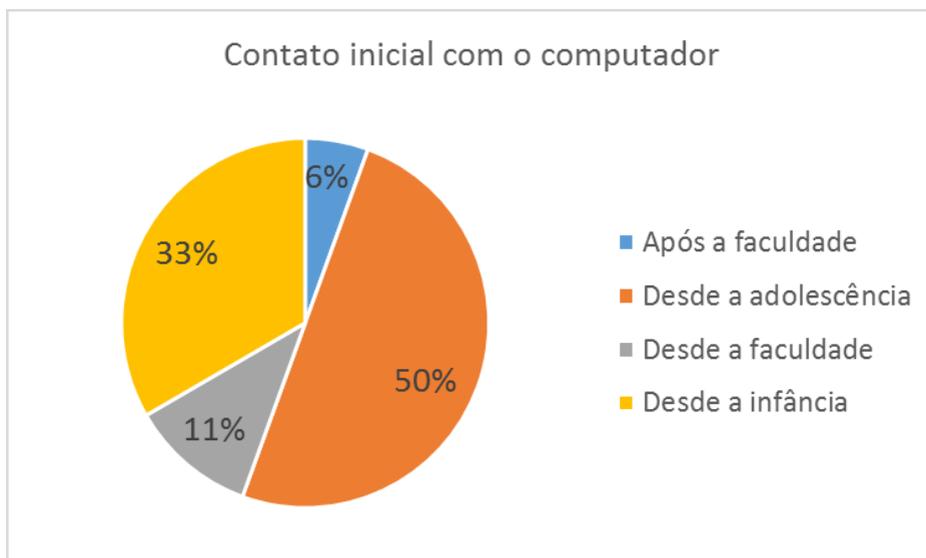


Figura 3.10 - Resultado da pergunta 1 (Q1).

Fonte: Os autores.

Observa-se na Figura 3.11 que a grande maioria alegou não conhecer o Scratch até a divulgação do curso. Isso é um indício que o Scratch não é muito utilizado, tanto nas escolas quanto nas universidades. Isso reforça nossa justificativa em buscar trabalhar o Scratch com estudantes no Brasil.

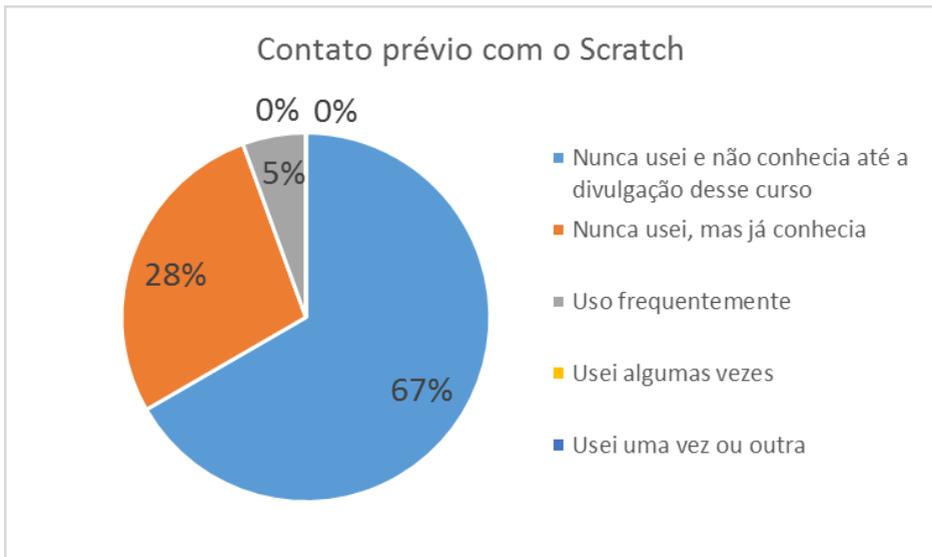


Figura 3.11 - Resultado da pergunta 3 (QI).

Fonte: Os autores.

Com relação ao conhecimento prévio em alguma linguagem de programação, observa-se na Figura 3.12 os resultados dos respondentes, ressaltando-se que 39% nunca programaram e 28% programaram algumas vezes, o que nos gera um panorama que a maior parte dos respondentes não tinham experiências prévias em

qualquer tipo de programação, demandando maior atenção durante as atividades por seu primeiro contato. Dessa forma, a importância de ter uma boa sequência didática aumenta.

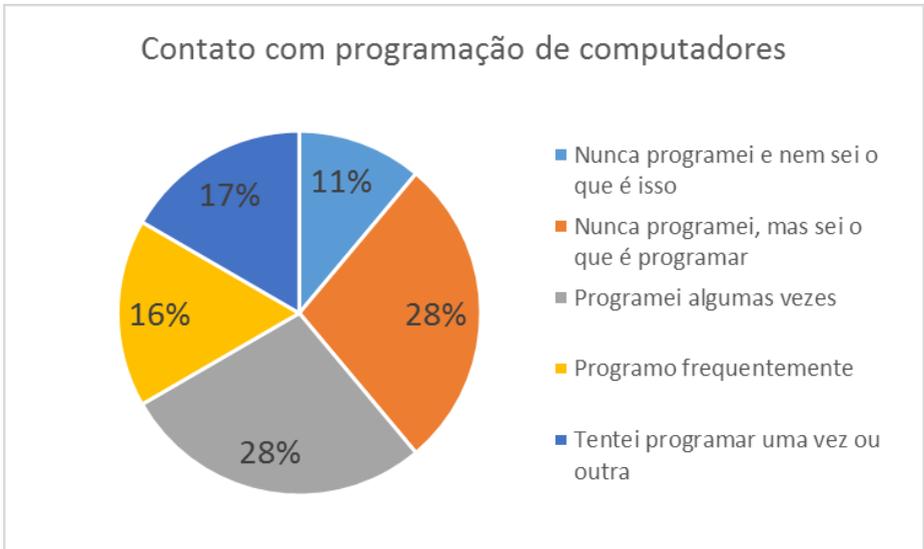


Figura 3.12 - Resultado da pergunta 2 (Q1).

Fonte: Os autores.

Com relação às questões de lógica, apresenta-se na Figura 3.13 os resultados do exercício 7 (Apêndice 3.B, Figura 3.28), no qual se apresenta uma grade com um pseudocódigo¹⁷ e pede-se que o estudante identifique a posição da bola ao fim do código, buscando-se verificar se o estudante compreende a ideia do código.

Observa-se que 78% das pessoas acertaram o

```
17circulo.movaParaCima(40);  
circulo.movaParaCima(3);
```

exercício 7, sendo as demais respostas erradas, indicando a dificuldade do estudante que pode advir de compreender a lógica que fundamentou o pseudocódigo ou de abstração espacial.



Figura 3.13 - Resultado do teste de lógica da pergunta 7 no questionário inicial (QI).

Fonte: Os autores.

Sequencialmente, no exercício 8 (Apêndice 3.B, Figura 3.28), apresentou-se novamente um pseudocódigo¹⁸, incluindo uma condicional, ou seja, um novo elemento que complexifica o código e exige maior raciocínio.

```
18repita(7)
    seta.mova (1);
    se(está na borda)
        gire.horario(90°);
    fimRepita
```

Observa-se (Figura 3.14) que 33% acertaram o exercício 8, as demais respostas estão erradas. Percebe-se que, quando comparado à questão 7, houve muito menos acertos, indicando claramente ser um tipo de raciocínio que deveria ser desenvolvido durante as atividades do curso com o Scratch.

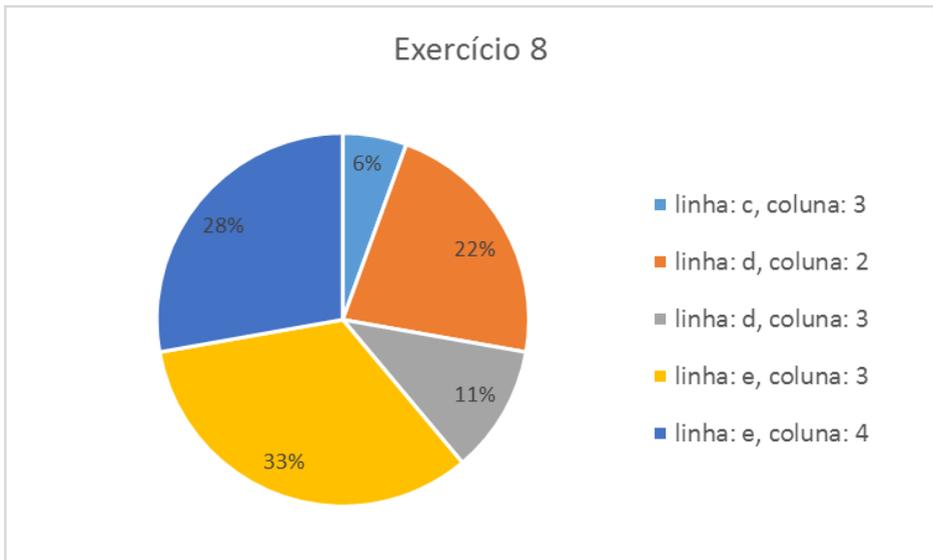


Figura 3.14 - Resultado do segundo teste de lógica da pergunta 8 no questionário inicial (QI).

Fonte: Os autores.

Para compreender a relação entre o perfil do respondente, com relação ao conhecimento prévio de programação, com as respostas aos exercícios 7 e 8, construíram-se os gráficos das Figuras 3.15 e 3.16.

Para o exercício 7, há uma distribuição com relação aos acertos entre as pessoas que programavam ou, pelo

menos, sabiam o que era, havendo a maior taxa de erros com quem nunca programou ou sequer sabia o que era.

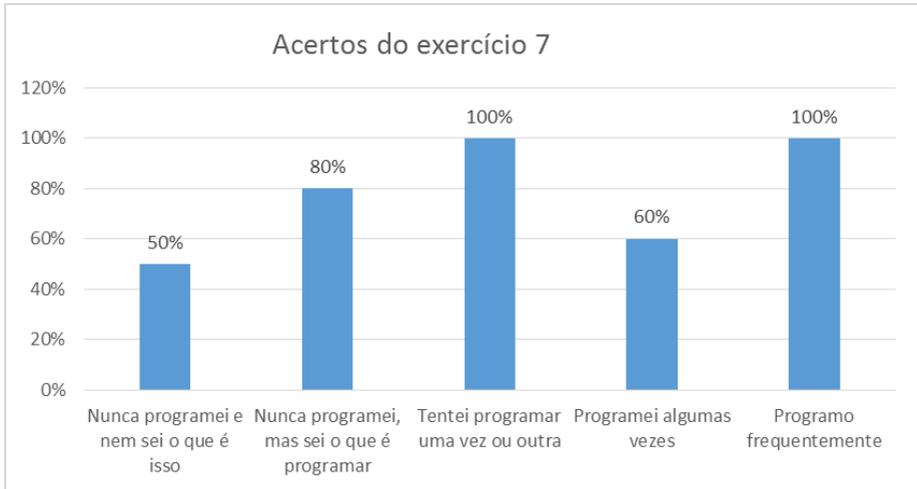


Figura 3.15 - Relação entre os acertos do exercício 7 com a frequência de programação de cada estudante.

Fonte: Os autores.

Quando observa-se os resultados de acertos ao exercício 8, o qual tinha maior complexidade e dificuldade, trazendo elementos de condicional, comumente utilizadas em programação. Neste caso, há uma diferença evidente entre quem programava frequentemente e quem não programava.

Apesar do raciocínio lógico das condicionais ser algo do dia a dia das pessoas, raramente elas sistematizam ou procedimentalizam tais ações, assim, quando se depararam com um problema deste tipo, tiveram maior dificuldade.

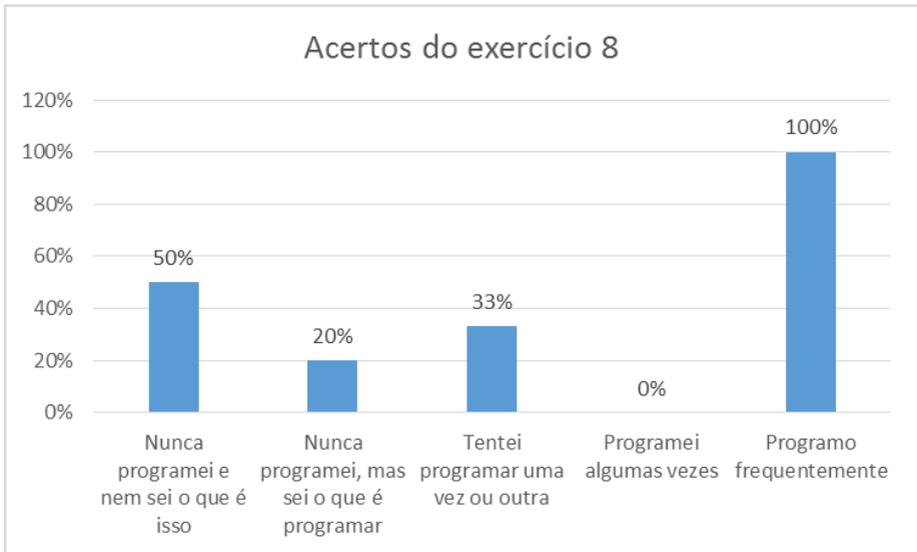


Figura 3.16 - Relação entre os acertos do exercício 8 com a frequência de programação de cada estudante

Fonte: Os autores.

Durante o curso, abordaram-se alguns conceitos de programação, tais como: constantes, condicionais, funções, procedimentos e tipos primitivos. Para identificar os conhecimentos prévios sobre estes, avaliou-se a autopercepção sobre os conhecimentos destes conceitos (Figura 3.17).

Identificou-se que, de modo geral, a maior parte não tinha conhecimentos sobre estes conceitos, sendo que 27% não tinha familiarização com nenhum deles, o que reforçou a necessidade de se trabalhar esses durante o curso.

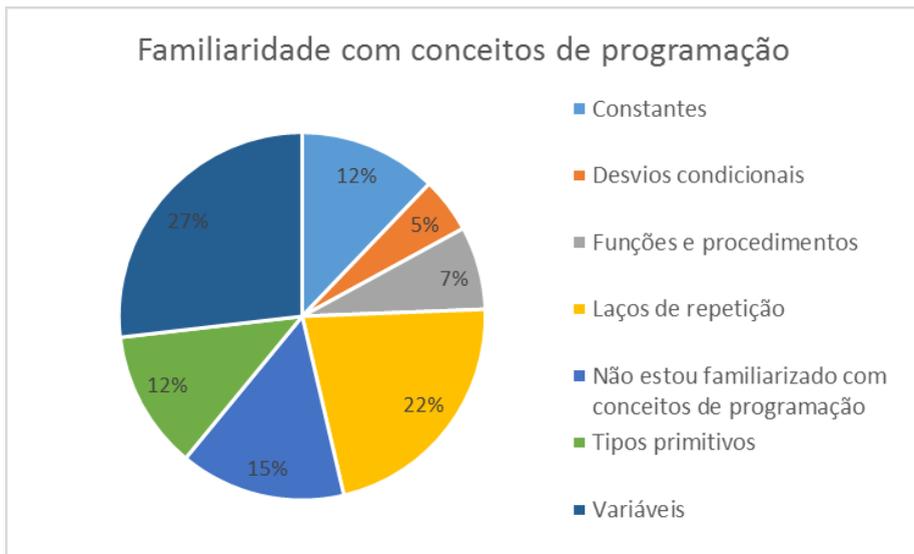


Figura 3.17 - Resultado da pergunta 4 (QI).

Fonte: Os autores.

Abordou-se no exercício 8 dois desses conceitos: laço de repetição e desvio condicional. Na Figura 3.18, apresenta-se a relação entre os estudantes que acertaram esse exercício e os que disseram estar familiarizados com esses conceitos, ou seja, dos estudantes que acertaram quais sabiam o que eram desvios condicionais e quais sabiam o que eram laços de repetição.

Destaca-se que 67% dos estudantes que disseram conhecer os conceitos de desvios condicionais acertaram o exercício 8, 33% que conheciam sobre laços de repetição e outros 33% não estavam familiarizados com nenhum desses dois conceitos e acertaram a questão.

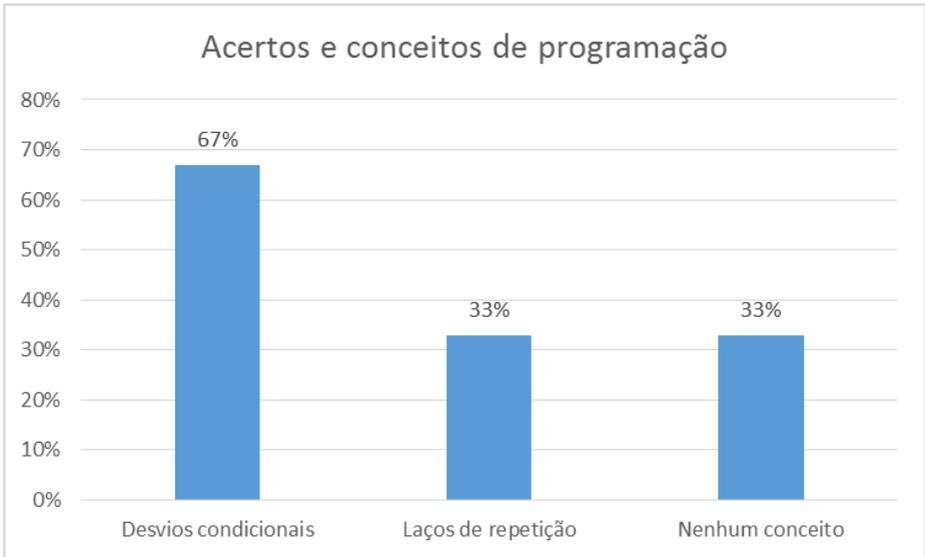


Figura 3.18 - Relação entre os acertos do exercício 8 com a familiaridade dos conceitos de programação

Fonte: Os autores.

A relação entre os estudantes que erraram esse exercício, mas, mesmo assim, disseram estar familiarizados com esse conceito, está representada na Figura 3.19.

Ressalta-se que 42% dos estudantes que erraram essa questão disseram estar familiarizados com laços de repetição e nenhum estudante desse grupo disse estar familiarizado com desvios condicionais.

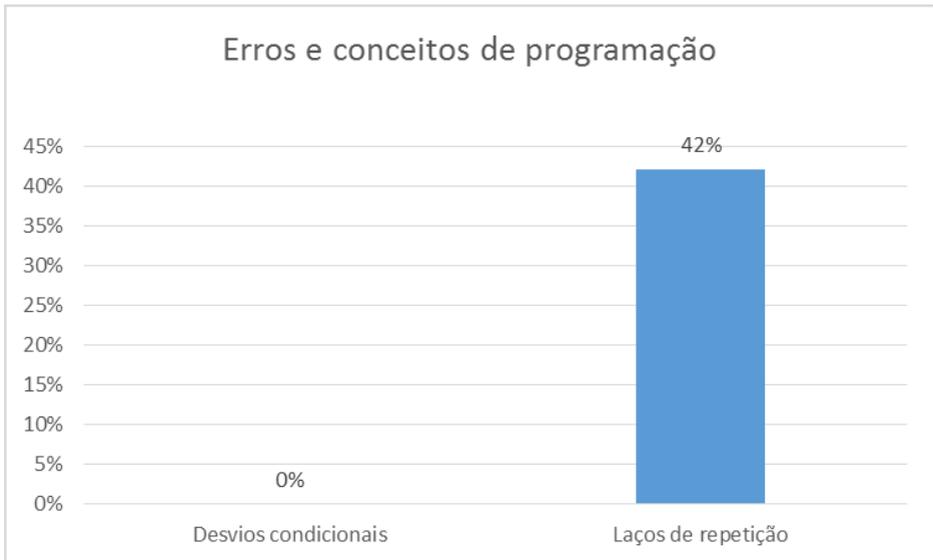


Figura 3.19 - Relação entre os erros do exercício 8 com a familiaridade dos conceitos de programação

Fonte: Os autores.

A fim de ponderar sobre o curso e avaliar se os estudantes tiveram afinidade ou não com o Scratch, aplicou-se o questionário final (Apêndice 2C).

Identificou-se que cerca de 57% dos estudantes disseram que o curso alcançou as expectativas (Figura 3.20).

Nenhum estudante avaliou as atividades com aspectos negativos, sendo que a maioria classificou, entre outras alternativas, as atividades como sendo interessantes (Figura 3.21).

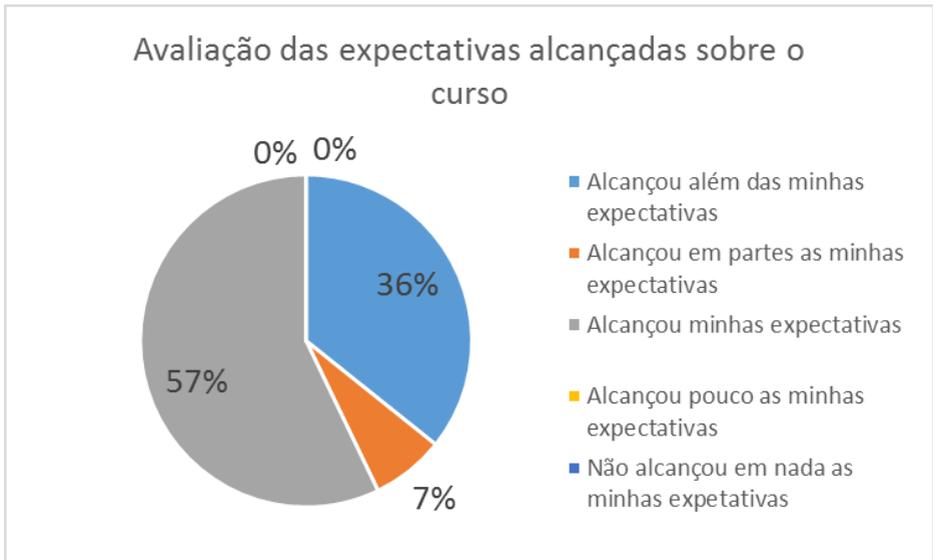


Figura 3.20 - Resultado da pergunta 1 (QF).

Fonte: Os autores.

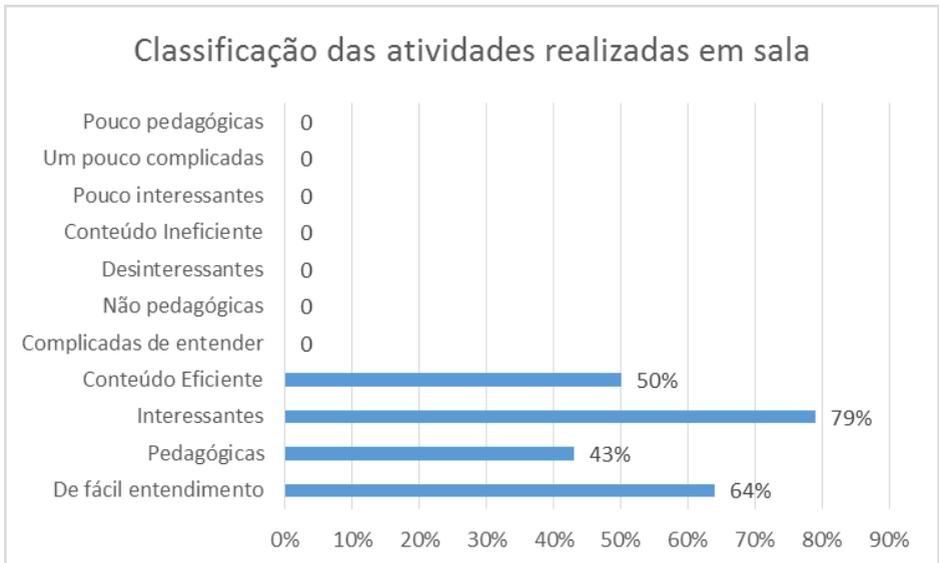


Figura 3.21 - Resultado da pergunta 2 (QF).

Fonte: Os autores.

Quase todos os estudantes disseram estarem aptos a utilizar todos os conceitos do Scratch (Figura 3.22). Com destaque aos blocos, que são a parte fundamental da ferramenta, todos os estudantes disseram compreendê-los e saber utilizá-los.

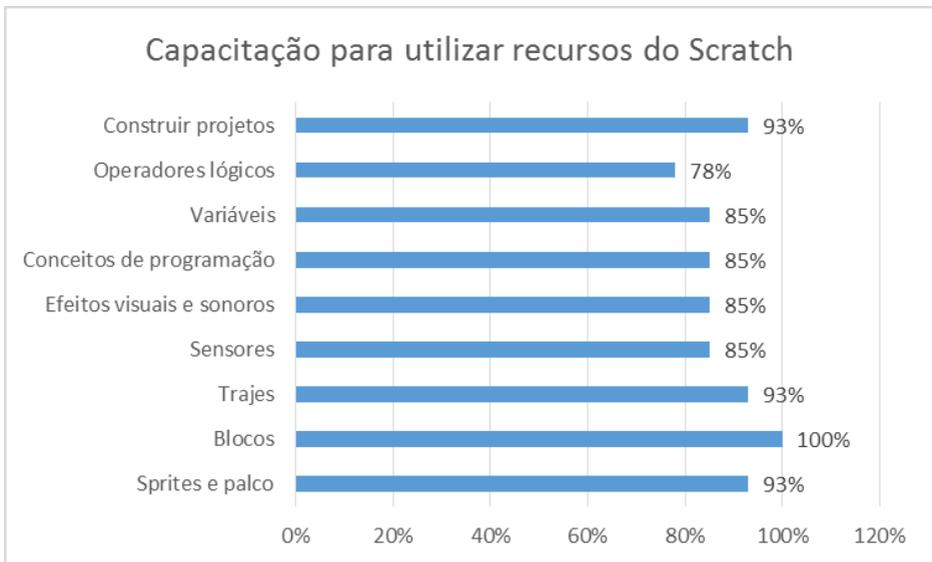


Figura 3.22 - Resultado da pergunta 5 (QF).

Fonte: Os autores.

Destaca-se que cerca de 80% disseram que o curso foi fundamental para a aprendizagem e melhora desses conhecimentos (Figura 3.23) e, a mesma quantidade, acredita que o Scratch pode melhorar o entendimento sobre programação (Figura 3.25), reforçando a importância deste curso ou outros que trabalhem os princípios da programação de forma lúdica.

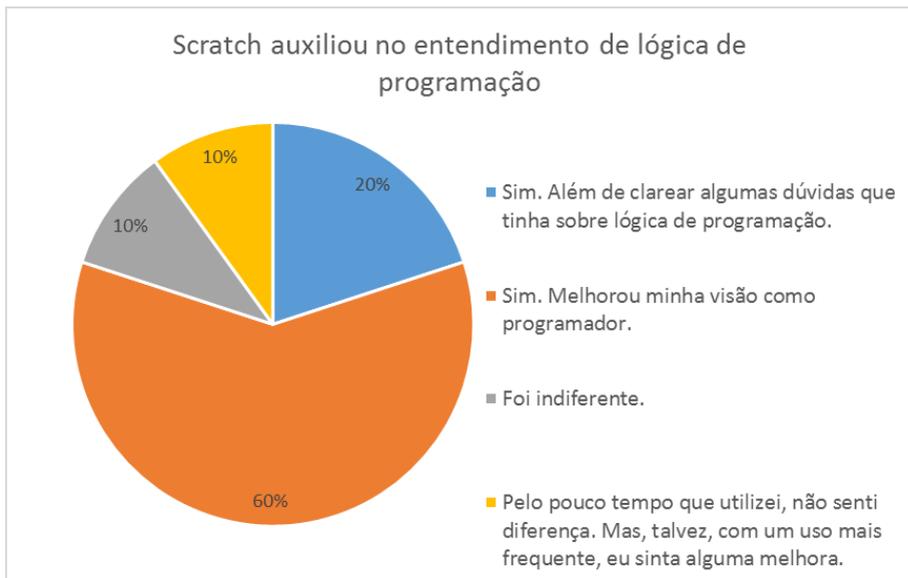


Figura 3.23 - Resultado da pergunta 8 (QF).

Fonte: Os autores.

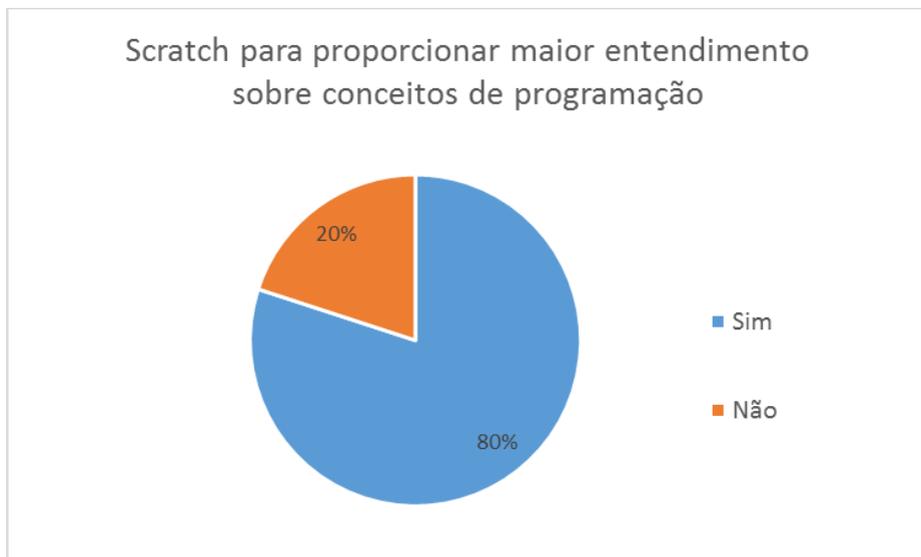


Figura 3.24 - Resultado da pergunta 7 (QF).

Fonte: Os autores.

Apresenta-se na Figura 3.25 as características de uso do Scratch apresentadas pelos participantes do curso, as quais indicaram a ferramenta como de fácil acesso, divertida e simples de usar, com fácil aprendizagem da mesma.

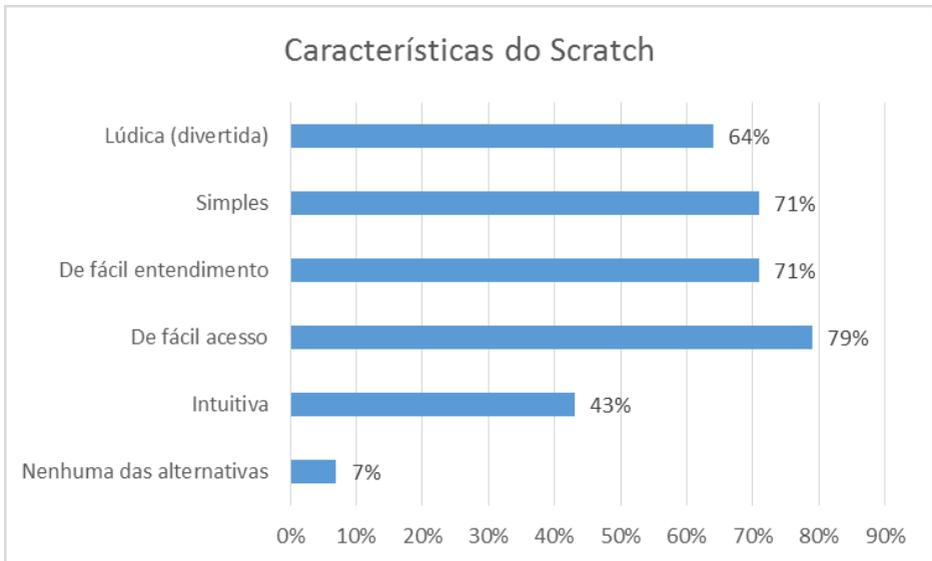


Figura 3.25 - Resultado da pergunta 9 (QF).

Fonte: Os autores.

Na Figura 3.26, observa-se que 36% dos participantes pretendem manter o uso do Scratch em seu cotidiano e 50% as vezes.

Somente 14% disseram não pretender utilizar o Scratch futuramente. Isso indica uma boa aderência a ferramenta, com indicação de aplicabilidade desta em outros contextos.

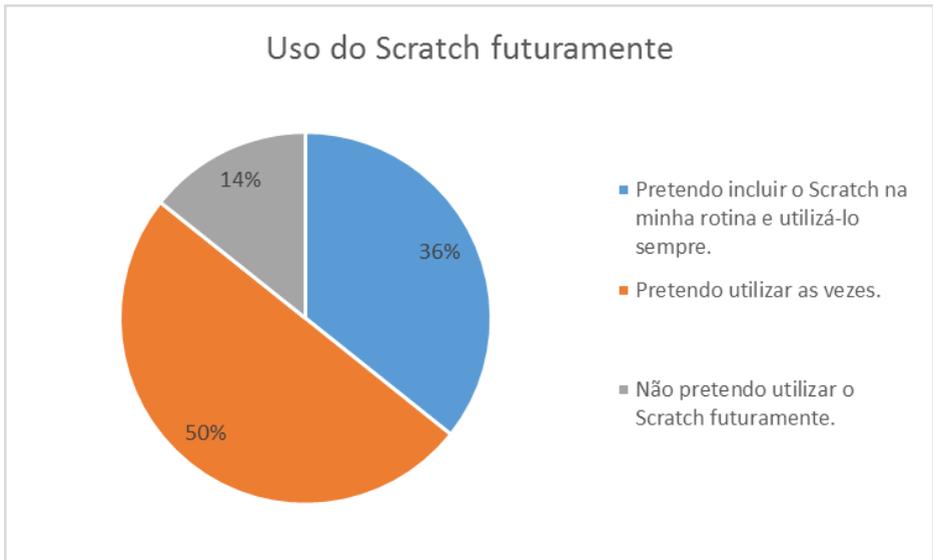


Figura 3.26 - Resultado da pergunta 10 (QF).

Fonte: Os autores.

Entre os participantes do ensino médio, 100% afirmou que o curso de Introdução ao Software Scratch poderá auxiliá-los a escolher um curso superior na área de tecnologia (Resultado à questão “O curso como auxiliar na escolha de um curso universitário”), uma vez que deu uma visão prévia do que seria programar um aplicativo.

Entre os estudantes ligados à educação superior, 80% acreditam que o Scratch tem o potencial para ser utilizado em sala de aula (Questão 12, Apêndice 3.C) e 60% acreditam que a ferramenta tem todos os recursos necessário para construir projetos educativos (Questão

13, Apêndice 3.C).

Esses resultados nos dão indícios que essa ferramenta poderia ser usada como ferramenta de autoria em salas de aula, seja pela facilidade de uso, pela motivação de construção de aplicativos próprios, facilidade de aprendizagem etc.

4 Considerações finais

Por meio desse trabalho, pode-se observar que o uso do Scratch pode ser um grande aliado na educação, pois, além de ser uma importante ferramenta para a construção lúdica de aplicativos, pode proporcionar uma ótima estratégia de ensino-aprendizagem. Da construção de projetos até a lógica de programação, o Scratch permite inúmeros benefícios, tanto para o desenvolvimento intelectual quanto para o desenvolvimento pessoal.

Há indícios de que o curso cumpriu seus objetivos, pois pode aproximar os estudantes da área de tecnologia, ensiná-los sobre lógica e conceitos de programação e, também, capacitá-los a serem independentes frente ao Scratch. Os estudantes classificaram a ferramenta como intuitiva e de fácil entendimento.

O curso e as atividades propostas foram bem-aceitas, porém, entre as sugestões dadas no questionário final, alguns estudantes disseram que a carga horária deveria ser maior para que o ensino do Scratch fosse melhor aproveitado.

Além disso, os estudantes aprovaram a metodologia em que o conteúdo e as atividades foram abordados, que foram por meio do professor como um facilitador no processo de aprendizagem e não como um transmissor de conhecimentos.

Por fim, por meio da revisão bibliográfica, foi possível fundamentar que o Scratch e o ensino de programação são capazes de aumentar a concentração, a autonomia e o desenvolvimento cognitivo dos estudantes, o que foi corroborado com os resultados obtidos nessa pesquisa.

Tem-se indícios que o Scratch tem potencial para se tornar um diferencial na educação nas escolas, formando indivíduos autônomos, proativos e com boa relação interpessoal.

Referências

BEHRENS, M. A. A prática pedagógica e o desafio do paradigma emergente. *Revista Brasileira de Estudos Pedagógicos*, Brasília, v. 80, p. 383-403, set./dez. 1999.

BRASIL, M. *Orientações Curriculares para o Ensino Médio*. Secretaria da Educação Básica. Brasília: Ministério da Educação, Secretaria de Educação Básica, 2007.

DALFOVO, M. S., LANA, R. A., SILVEIRA, A. *Métodos quantitativos e qualitativos: um resgate teórico*. *Revista Interdisciplinar Científica Aplicada*, Blumenau, v. 2, n. 4, p. 01-13, 2008.

GLADCHEFF, A. P., ZUFFI, E. M., SILVA, D. M. Um instrumento para avaliação da qualidade de softwares educacionais de matemática para o ensino fundamental. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 21., 2001. Fortaleza. *Anais...* Fortaleza: SBC, 2001. p. 1-12.

INTERNATIONAL STANDARD. *INFORMATION TECHNOLOGY*. ISO/IEC 9126-1. Software quality characteristics and metrics. 1997. Disponível em: <<http://www.sqa.net/iso9126.html>>. Acesso em: 8 abr. 2015.

KELLEHER, C., PAUSCH, R. Lowering the Barriers to Programming: a survey of programming environments and languages for novice programmers. *ACM Comput. Surv.*, v. 37, p. 83-137, 2005.

MALONEY, J., et al. *The Scratch Programming Language and Environment*. *ACM Transactions on Computing Education*, v. 10, nov. 2010.

MARQUES, D. L., et al. Atraindo Alunos do Ensino Médio para a Computação: uma experiência prática de introdução a programação utilizando jogos e Python. In: SBIE, 22; WIE, 27, 2011. Aracaju. *Anais...* Aracajú: SBC, 2011. p. 1138-1147. Disponível em:

<<http://dx.doi.org/10.5753/cbie.wie.2011.1138-1147>>.

Acesso em 06 set. 2018.

MELO NETO, J. A. *Tecnologia Educacional: formação de professores no labirinto do ciberespaço*. Rio de Janeiro: MEMVAVMEM, 2007.

MORAES, M. C. *O Paradigma Educacional Emergente: implicações na formação do professor e nas práticas pedagógicas*. Brasília: Aberto, 1996.

NERI, M. C. *Mapa da exclusão digital*. Rio de Janeiro: FGV/IBRE, p. 88-91, 2003.

NIELSEN, J. *Usability Engineering*. New York: Elsevier, 1994.

PATTO, M. H. S. *Introdução à Psicologia Escolar*. 3. ed. São Paulo: Casa do Psicólogo, 1997.

RESNICK, M. et al. *Scratch: Programming for All*. Communications of the ACM, v. 52, n. 11, nov. 2009.

ROCHA, S. S. D. O uso do computador na educação: a informática educativa. *Revista Espaço Acadêmico*, Maringá, n. 85, jun. 2008.

SILVA, A. R. O. S., TAVARES, M. A. O. Leitura de Cordel e a Programação de Computadores. *Revista Práticas Pedagógicas: Registros e Reflexões*, Santos, SP, v. 1, 2012.

SILVINO, A. M. D. *Epistemologia Positivista: Qual a sua Influência hoje? Psicologia, Ciência e Profissão*. Brasília, v. 27, p. 276-289, 2007.

SOUSA, D. F. Desenvolvendo a Lógica e Algoritmos no Ensino Médio. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2; WORKSHOP DE INFORMÁTICA NA ESCOLA, 19, 2013. Campinas. *Anais...* Campinas: SBC, 2013. p. 101-109. Disponível em:

<<http://dx.doi.org/10.5753/cbie.wie.2013.%25p>>. Acesso em: 06 set. 2018.

TANRIKULU, E., SCHAEFER, B. C. The users who touched the ceiling of scratch. *Procedia - Social and Behavioral Sciences*, [S.I], v. 28, p. 764-769, 2011.

TSUKUMO, A. N. *Qualidade de Software: Visões de Produto e Processo de Software*. CITS, 1997.

VALENTE, J. A. *Computadores e conhecimento: repensando a educação*. 2ª edição. Campinas, SP: UNICAMP/NIED, 1993.

VOELCKER, M. D.; DA CRUZ FAGUNDES, L.; SEIDEL, S. *Fluência digital e ambientes de autoria multimídia*. *RENOTE*, v. 6, n. 2, 2008.

Capítulo 4. InMapMoodle: ferramenta para moderação de fóruns

*Raphael Biavati
Gabriel Gerber Hornink*

O crescimento pela demanda de ferramentas para auxílio em ambientes virtuais de aprendizagem (AVAs) tem sido constante, disponibilizando assim abordagens interessantes para a ampliação de técnicas de desenvolvimento e estudo. Com o avanço das novas formas de educação à distância, os ambientes virtuais passaram a ser cada vez mais utilizados, de modo a permitir uma maior interação do estudante com outros estudantes e formadores. A possibilidade de aliar uma ferramenta que traga mais interação em um ambiente como esse, garante que a linha de pesquisa possa ser cada vez mais ampla e necessária. O foco do presente trabalho é apresentar princípios de desenvolvimento e características funcionais da ferramenta inMapMoodle©, desenvolvida no CEAD em parceria com o LP&D, que disponibiliza grafos direcionais das interações ocorridas nos fóruns, de modo que os resultados exibidos pela ferramenta possam servir como parâmetros para demonstrar a participação dos usuários em fóruns, auxiliando na moderação pelos professores e tutores e sirva como incentivo em futuros trabalhos na área de pesquisa.

1 Introdução

A área de educação a distância (EAD) tem ganhado cada vez mais reconhecimento e importância. A excelência em garantir maior interatividade ao estudante se torna algo indispensável, de modo que a necessidade pelo aprimoramento e desenvolvimento de ferramentas para os ambientes virtuais de aprendizagem tem crescido de maneira significativa.

Segundo Behar et al., (2004), o ambiente virtual de aprendizagem é um espaço, formado por usuários e objetos de estudos, que permite interações e formas de comunicação por meio de uma plataforma.

O ensino por meio da utilização de ambientes virtuais possibilita propor atividades interativas e significativas ao estudante. Para Santos (2003), não basta apenas criar um site e disponibilizá-lo na internet, o mesmo precisa ser totalmente interativo e a interatividade com o conteúdo publicado e com os próprios é o que faz com quem um site se constitua um AVA com maior potencial educacional.

Segundo Moore (2007), existem três tipos de interação em EAD: estudante-conteúdo, estudante-professor e estudante-estudante. O primeiro tipo se

desenvolve em formas de textos, imagens, vídeos, entre outros. Dessa maneira, é possível avaliar a interação do estudante com o conteúdo publicado e com a tecnologia usada.

O segundo tipo tem como objetivo destacar o papel do professor em motivar e direcionar o estudante ao aprendizado da maneira mais fácil, possibilitando abertura para estimular o interesse nos conteúdos e na disciplina como um todo. O terceiro e último envolve a colaboração entre os estudantes, envolvendo assim a possibilidade de trabalho em equipe e colaboração.

Ressalta-se que nos cursos *online* há outros papéis que ampliam as possibilidades de interação, como tutor presencial, tutor a distância, monitores, entre outros.

Destaca-se também que se entende que a interação com os conteúdos e demais mídias é uma interação mediada por artefatos culturais com outros sujeitos, ou seja, "agentes - agindo - com - ferramentas - culturais" (GIORDAN, 2008).

A partir do foco nas interações entre estudante-professor, estudante-estudante e outros sujeitos, a principal proposta desse capítulo é apresentar princípios de desenvolvimento e características funcionais da ferramenta inMapMoodle©, desenvolvida em sua

primeira versão por Ferraz et al., (2012) e segunda versão por Ferraz et al., (2016), que possibilita mapear essas interações pela troca de mensagens em fóruns da plataforma Moodle e exibi-las por meio da cartografia digital, sendo mais conhecida no ambiente computacional como grafos direcionais.

Os grafos direcionais são amplamente utilizados para demonstrar problemas reais em diversas áreas, o grafo não só ajuda na demonstração de interações como também apresenta uma melhor visualização do problema através de um conjunto de pontos ligados por retas, sejam elas direcionais ou não.

Apresenta-se neste texto as características funcionais da ferramenta inMapMoodle®, sua avaliação, além dos princípios de desenvolvimento de blocos Moodle, possibilitando a compreensão da construção desta ferramenta, seja para seu uso ou mesmo para novos desenvolvimentos de blocos para o Moodle.

2 Ambientes virtuais de aprendizagem

O avanço no desenvolvimento das tecnologias digitais está impulsionando a maneira de ensinar e aprender. Dentre essas tecnologias, os Ambientes Virtuais de Aprendizagem (AVAs) estão sendo cada vez

mais utilizados como uma ferramenta para atender a demanda da área educacional, pois além de permitir interatividade com o conteúdo, possui ferramentas que viabilizam a colaboração *online*, convergindo diversas ferramentas em um único ambiente.

O Ministério da Educação conceitua AVAs como:

[...] programas que permitem o armazenamento, a administração e a disponibilização de conteúdos no formato web. Dentre esses, destacam-se: aulas virtuais, objetos de aprendizagem, simuladores, fóruns, salas de bate-papo, conexões a materiais externos, atividades interativas, tarefas virtuais (webquest), modeladores, animações, textos colaborativos (wiki) (BRASIL, 2007, p. 11).

Os benefícios de um AVA com relação ao uso no processo de ensino-aprendizagem depende de vários fatores, tais como: o envolvimento do estudante, materiais vinculados, design, proposta didática, infraestrutura necessária para que essa plataforma permaneça *online*, de professores e tutores bem capacitados e, não muito menos de ferramentas e recursos tecnológicos utilizados para melhor a interação entre estudante (alta navegabilidade e usabilidade), professor e conteúdo.

Destaca-se também que, além dos fatores expostos, que os AVAs devem possibilitar a visão global das atividades desenvolvidas pelos estudantes,

fornecendo instrumentos para visualização de métricas para acompanhamento e direcionamento da moderação, isso sem desqualificar os aspectos qualitativos envolvidos nos processos de ensino aprendizagem que são de extrema importância para o acompanhamento das atividades *online*. Apresenta-se a seguir os dois AVAs abertos mais usados no Brasil.

2.1 Moodle

O Moodle é um *software* livre para gerenciamento de cursos destinado a auxiliar educadores na implantação de cursos em um ambiente virtual (TEODORO; ROCHA, 2007).

A plataforma Moodle originalmente refere-se às iniciais de *Modular Object-Oriented Dynamic Learning Environment*. Foi criada por Martin Dougiamas na *Curtin na University of Technology* (Austrália), em 2001, com base no construtivismo social, ou seja, foi desenhada e desenvolvida com base em contextos políticos e sociais (MOODLE, 2018).

Atualmente é desenvolvido de maneira colaborativa, por uma comunidade virtual, que geram versões constantes e atualizadas do sistema, trazendo melhorias e ajuste de erros os quais são postados no

sítio <http://www.moodle.org>.

O Moodle possui uma estrutura modular, uma comunidade ampla de desenvolvedores, grande quantidade de documentação, disponibilidade, escalabilidade, facilidade de uso, interoperabilidade, estabilidade e segurança.

O *software* foi desenvolvido na linguagem PHP e pode ser executado em diversos ambientes operacionais desde que o mesmo possua a capacidade de interpretar a linguagem na qual o *software* foi desenvolvido.

Apresenta-se na Figura 4.1 um exemplo de tela inicial de um curso no Moodle (versão 3.5), a qual foi organizada em semanas, sendo estas dispostas em abas clicáveis, por meio do bloco *Tabs*.

Na lateral esquerda se observa o menu principal, o qual pode ser oculto clicando no ícone superior esquerdo, de tal forma que a tela principal de trabalho pode ser ampliada para melhorar a usabilidade/visibilidade do sistema.

Por meio do menu é possível navegar pelas semanas do curso, assim como acessar as notas, competências, cronograma, além das funções administrativas.

The image shows a Moodle course page. On the left is a sidebar menu with items like 'Participantes', 'Competências', 'Notas', 'Painel', 'Página inicial do site', 'Calendário', 'Arquivos privados', 'Learner Dashboard', 'Instructor Dashboard', 'Competency Dashboard', 'Meus cursos', and 'Administração do site'. The main content area has a header 'Pensando em Códigos: Scratch 2018' and a sub-header 'Boas vindas'. Below this is a progress bar with tabs for 'Boas vindas', '1ª Semana', '2ª Semana', '3ª Semana', '4ª Semana', '5ª Semana', and 'Projeto Final'. The '1ª Semana' tab is active, showing the title '1ª Semana' and the date '02 a 8 de Abril - Scratch'. There is a Scratch logo and a text block explaining that Scratch programs are called scripts and are created by dragging and dropping blocks. Below this is an 'Introdução' section with a list of activities: 'Leia e siga as orientações (1S)', 'AULA 01 - Apresentação do Scratch', '1.1 - Apostila de Scratch', and '1.3 Avaliação - Questionário sobre o vídeo 01'. A 'Feedback' box indicates that the 'AULA 01' activity is marked as completed. At the bottom, a message says 'Você deverão assistir o vídeo acima para responder esta questão'.

Figura 4.1 - Página inicial de um curso no Moodle.

Fonte: os autores

2.2 TelEduc

O TelEduc é um ambiente semelhante ao Moodle e com a mesma finalidade de criação e administração de cursos de maneira online. É um *software* livre, desenvolvido na Universidade Estadual de Campinas (Unicamp), por pesquisadores do Núcleo de Informática Aplicada à Educação (Nied), inicialmente em 1997, sendo que foi aberto para comunidade em 2001, desta forma, sua criação e uso antecede ao Moodle.

A plataforma foi desenvolvida com o intuito de formar professores para a área educativa, baseado na

metodologia de formação contextualizada desenvolvida na Unicamp (GABARDO et al., 2010).

Segundo Simões e Lindernann (2007), as linguagens utilizadas para a sua criação são PHP para praticamente todas as páginas, Perl para ferramentas de bate-papo e Java na ferramenta Intermap. Destaca-se o Intermap pela similaridade com as funções do inMapMoodle©.

Projeto ensino FAPESP/PETROBRAS
Agenda - 2010 Busca Ajuda

Agendas Anteriores Editar Agendas

Ribeirão Anhumas na Escola

Apoio: **FAPESP** **CNPq** Patrocínio: **PETROBRAS AMBIENTAL** **BR** **PETROBRAS** **SEM FINEANÇAS**

Hoje! novembro de 2009 Imprimir Semana Mês

seq	ter	qua	qui	sex	sáb	dom
26	27	28	29	30	31	
		07:50 4ª Aula Form.	19:00 Reunião grup			
2	3	4	5	6	7	
		07:00 4º Campo do	19:00 Reunião grup			

Figura 4.2 - Página exemplo de um curso no TelEduc.

Fonte: HORNINK, 2010.

3 Comparativo entre as plataformas

Além dos ambientes Moodle e TelEduc há outros ambientes usados para o ensino a distância, de código aberto e fechado, mas que serão apresentados de maneira superficial no comparativo, tendo em vista que o foco do presente trabalho é apresentar a ferramenta inMapMoodle© desenvolvida unicamente para o

ambiente Moodle, além disso, demonstrar a ferramenta InterMap do ambiente TelEduc.

Segundo Gabardo et al., (2010), cada tipo de ambiente virtual de aprendizagem de ensino segue padrões para o seu desenvolvimento distinto, apresentando características os que tornam mais adequado para cada situação de ensino-aprendizagem, seja ela em cursos online ou nos modelos híbridos, como por exemplo o Amadeus, que é um sistema de gestão de aprendizagem de segunda geração, baseado no conceito de *blended learning*. Ou seja, a palavra *blended* refere-se a algo misto e nesse caso a um sistema onde a maior parte do conteúdo é transmitido em curso a distância, entretanto inclui a parte presencial.

Apresenta-se na Tabela 4.1 um comparativo entre as principais ambientes virtuais de aprendizagem (AulaNet, TelEduc, AmCorA, LEDS, MC2, WebCT, Moodle), levando em conta os serviços disponíveis em cada uma delas. Os serviços são separados em três categorias: comunicação, coordenação e cooperação.

O Moodle se destaca com o maior número de serviços disponíveis, ressaltando as ferramentas para cooperação entre/ com os estudantes, como o sistema de revisão por pares e glossário.

		AulaNet	TelEduc	AmCorA	LEDS	MC2	WebCT	MOODLE
Serviços de Comunicação	Correio	■	■	■	■	■	■	■
	Lista de Discussão	■	■	■	■	■	■	■
	Fórum	■	■	■	■	■	■	■
	Mural	■	■	■	■	■	■	■
	Chat	■	■	■	■	■	■	☒
	Mensageiro	■	■	■	■	■	■	■
Serviços de Coordenação	Agenda	■	■	■	■	■	■	■
	Relat de Atividades	☒	■	■	■	■	☒	■
	Acomp. da Particip.	☒	■	■	■	■	■	■
	Questionário	■	■	■	■	■	■	■
	Tarefas	■	■	■	■	■	■	■
	SubGrupos	■	■	■	■	■	■	■
	Gerenc. de recursos	■	■	■	■	■	■	■
	Orientação	■	■	■	■	■	■	■
	Votação	■	■	■	■	■	■	■
Serviços de Cooperação	Reposit. de Conteúdos	■	☒	■	■	■	■	■
	Quadro Branco	■	■	■	■	■	■	■
	Busca	■	■	■	■	■	■	■
	Glossário	■	■	■	■	■	■	■
	Links	■	■	■	■	■	■	■
	Jornal Cooperativo	■	■	■	■	■	■	■
	Gerenc. de contatos	■	■	☒	■	■	■	■
	Revisão em pares	■	■	■	■	■	■	■
	FAQ	■	■	■	■	■	■	■
	Anotações	■	■	■	■	■	■	■

Tabela 4.1 - Comparativo entre as plataformas.
Fonte: ALMEIDA; SIEBRA, 2007

Apresenta-se na Tabela 4.2 os requisitos mínimos para a instalação e configuração das plataformas Moodle e TelEduc.

Tabela 4.2 - Requisitos necessários para instalação de cada AVA.

AVA	Tecnologias utilizadas
TelEduc	Software livre e de código aberto, Sistema Operacional Linux, Linguagem PHP, Servidor Web Apache, Banco de Dados MySQL
Moodle	Software livre e de código aberto, Multiplataforma, Linguagem php, Servidor Web Apache (ideal), mas trabalha com qualquer outro que suporte php, Banco de Dados MySQL, Postgre

Fonte: PEREIRA et al., 2008

4 Fóruns

Fórum de discussão é uma ferramenta destinada a promover troca de mensagens abordando um tema ou uma questão específica. Portanto, o fórum é simplesmente um lugar de interação entre estudantes e professores que permite a participação de todos envolvidos em um processo de aprendizagem sobre determinado assunto, promovendo assim discussões e argumentações como base para a colaboração entre todos participantes de um determinado curso.

Segundo Silva (2006), esse tipo de ferramenta potencializa a aprendizagem colaborativa. Basicamente os fóruns são divididos por assuntos ou tópicos e as mensagens ordenadas conforme a ordem de debate.

Partindo do princípio do fórum na plataforma Moodle, ela pode ser considerada uma ferramenta poderosa de comunicação entre os usuários. Ela é basicamente um quadro de mensagens que estudantes e professores postam mensagens entre os grupos e que permite de maneira muito simples acompanhar o andamento das discussões.

O fórum por si só permite um tipo de comunicação assíncrona, ou seja, ele não estabelece que o participante necessite ficar online durante todo o período

de discussão, mas sim que cada participante manifeste sua opinião, acompanhe a discussão e elabore debates ao longo da mesma. Isso passa a permitir que as pessoas possam se manifestar com mais precisão e clareza.

Segundo Hornink (2010), o fórum é um importante meio mediacional nos processos de comunicação online e a complexidade de organização dos mesmos, exige uma análise mais detalhada com relação as distintas interações, troca de mensagens de forma cronológica, integração entre tópicos e participantes, entre outros. A criação de uma ferramenta que possibilite minimizar a complexidade e que torne as informações mais facilmente de serem visualizadas serviu como auxílio no desenvolvimento deste trabalho.

As discussões e as trocas de ideias podem ser pontos importantes em determinados cursos e o modo como mensurar isso quando a troca de mensagens é em grande quantidade pode ser complicado e muita das vezes exaustivo.

Surge então o conceito de cartografia e conseqüentemente a geração de mapa das discussões na forma de grafos de interação (HORNINK, 2010). Deste propósito surgiu o desenvolvimento da ferramenta

inMapMoodle© (versão 1: FERRAZ et al, 2012; versão 2: FERRAZ et al, 2016) apresentada neste trabalho, que de modo mais fácil e interessante permite que o professor ou moderador do curso possa visualizar todas as interações entre os usuários em determinado fórum, entre determinadas datas, etc.

Conseguir a participação de estudantes em determinados fóruns pode ser um desafio, de modo que uma ferramenta que possa mensurar a participação e atribuir isso como um retorno ao professor e/ou tutor, acaba sendo muito eficiente.

Porém, o mais importante em determinados aspectos seria além de mensurar a quantidade de mensagens dos estudantes, também atribuir a qualidade delas. Essa e outras características serão aprimoradas em futuros trabalhos e melhorias no desenvolvimento da ferramenta inMapMoodle©.

5 Grafos e suas aplicações computacionais

Grafos é um objeto de estudo da Teoria dos Grafos representado por um conjunto de pontos ligados por retas. As retas podem ser direcionadas ou não, dependendo do tipo da aplicação desenvolvida (THOMAS, 2000).

Os grafos são objetos interessantes para representar problemas reais de modo a garantir que os algoritmos específicos consigam realizar problemas não solucionáveis pela mente humana. Além disso, acaba se tornando uma ferramenta poderosa de visualização (THOMAS, 2000).

Segundo Thomas (2000), o grafo é uma representação gráfica das relações existentes entre elementos de dados.

5.1 Aplicações computacionais

A Teoria dos grafos é amplamente utilizada na área computacional, sendo que os algoritmos conhecidos em grafos são de extrema importância para as principais aplicações computacionais conhecidas atualmente (CORMEN et al., 2001).

Entre as aplicações estão:

- 1. Análise de redes sociais:** amplamente utilizado devido a sua capacidade de representação. Esse tipo de análise permite, por exemplo, a sugestão de amizade (Figura 4.2), tal como muitas redes sociais têm adotado atualmente.

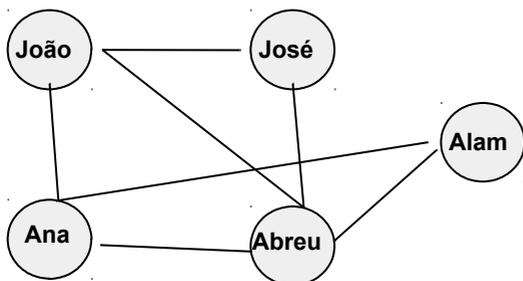


Figura 4.3 - Exemplo de grafo social.

Fonte: Os autores

2. Google Maps: permite a representação de localidades, assim como traçar caminhos entre um local de origem e um de destino (Figura 4.2). Isso permite traçar rota com tempo e distância, além de permitir representar isso de uma maneira simples de visualizar.

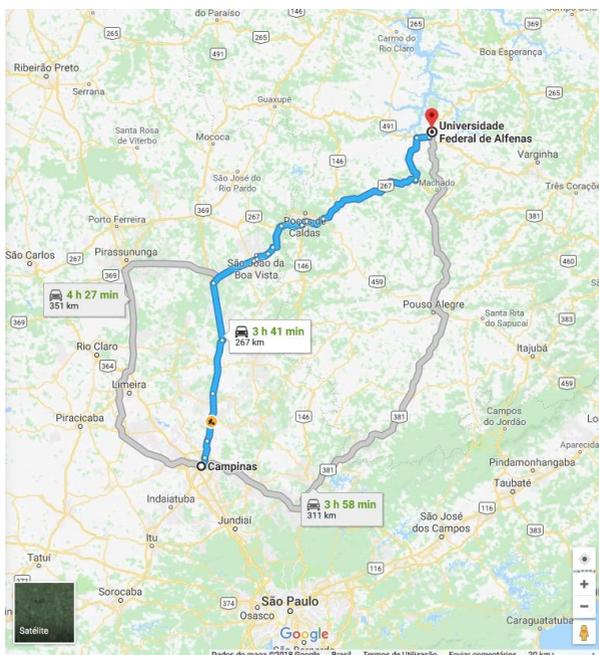


Figura 4.4 - Exemplo de rota no Google Maps.

Fonte: Tela capturada do Google Maps.

3. *InterMap*: Segundo Rocha et al., (2002), esta ferramenta foi criada para que o participante pudesse aprender sobre o histórico de discussões e a relação dos participantes em um determinado curso. O InterMap exibe interações nas ferramentas de Correio, fóruns e bate-papos e é gerado a partir de um Applet Java.

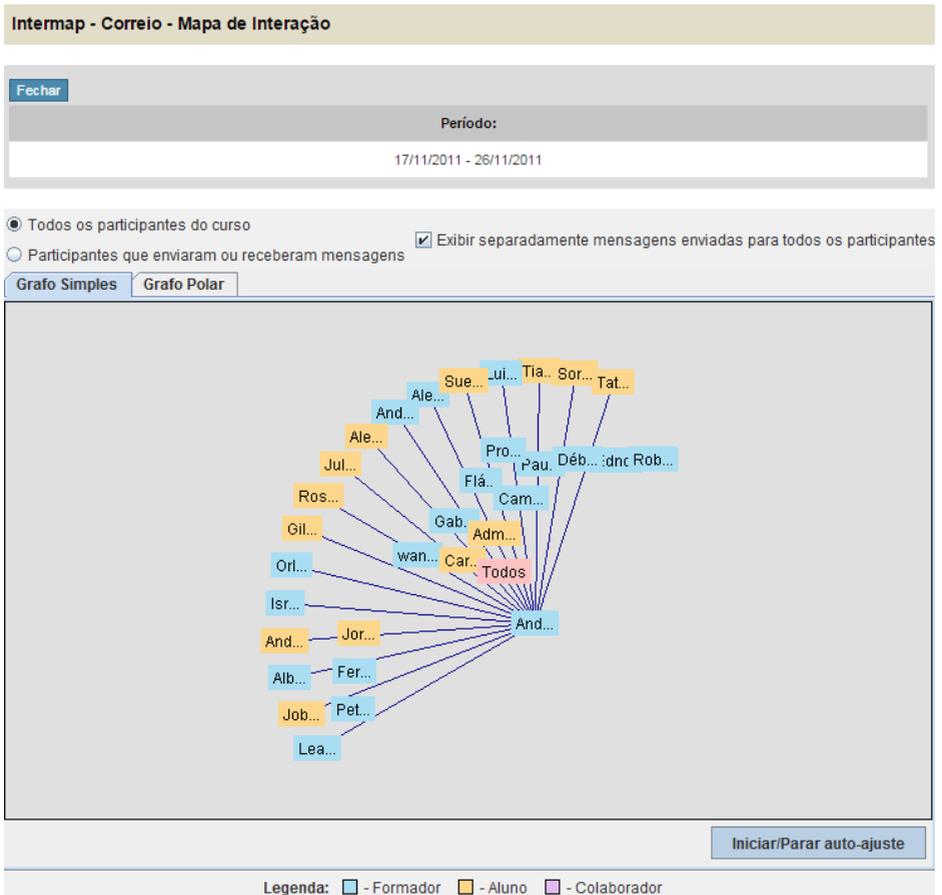


Figura 4.5 - Página exemplo do InterMap.

Fonte: TELEDUC, 2018.

5 A Ferramenta inMapMoodle

A ferramenta inMapMoodle© (FERRAZ et al, 2012), versão 1, foi desenvolvida no CEAD, em parceria com o Laboratório de Pesquisa e Desenvolvimento (LP&D), com o intuito de ser uma ferramenta gratuita para a comunidade do *software* livre Moodle e com a finalidade de demonstrar as interações entre os participantes dos fóruns de qualquer disciplina de um determinado curso.

A segunda versão (FERRAZ et al., 2016) foi resultado de melhorias advindas da avaliação de usabilidade da primeira versão, incluindo filtro de grupos e correções nos percentuais das legendas.

A ferramenta inMapMoodle se baseia na ferramenta InterMap, no que diz respeito a ser uma ferramenta para criar grafos dos fóruns, entretanto, difere pela plataforma de aplicação (Moodle) e pelas funcionalidades de direcionalidade (grafos direcionais) e pelo uso de legendas de cores para as quantidades de mensagens por participante.

Apresenta-se nas Figuras 4.6 e 4.7 dados estatísticos que ressaltam a utilização da plataforma Moodle mundialmente, justificando o desenvolvimento do inMapMoodle para esta plataforma.

Registered sites	98,562
Countries	232
Courses	15,223,607
Users	131,943,212
Enrolments	548,363,291
Forum posts	265,217,479
Resources	136,924,729
Quiz questions	788,886,849

Figura 4.6 - Estatísticas do Moodle - 2018

Fonte : Moodle Stats 2018



Country	Registrations
Estados Unidos da América	10,053
Espanha	8,353
Mexico	5,947
Brasil	5,149
Reino Unido da Grã-Bretanha e da Irlanda do Norte	3,504
Alemanha, República Federal da	3,098
Itália	2,996
Colômbia	2,888
Rússia (Federação Russa)	2,542
Índia	2,164

Top 10 from registered sites in 232 countries

Figura 4.7 - Países líderes no uso da plataforma Moodle - 2018

Fonte : Moodle Stats 2018

5.1 Desenvolvimento

Para o desenvolvimento dos requisitos levantados, fez-se necessário a utilização das tecnologias a seguir.

i. PHP

É uma linguagem interpretada capaz de gerar conteúdo dinâmico na web. A escolha da linguagem está relacionada a sua similaridade com relação ao Moodle e também pela portabilidade, velocidade e simplicidade.

ii. JavaScript

a. *Dracula Graph Library*

Para gerar a cartografia das interações, foi utilizada uma biblioteca javascript. Basicamente é um conjunto de ferramentas para demonstrar grafos de uma maneira totalmente interativa.

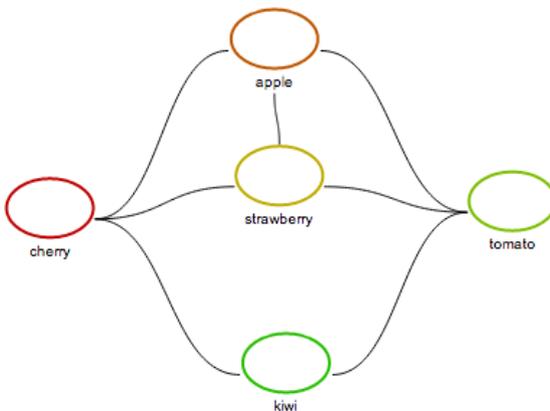


Figura 4.8 - Exemplo da biblioteca Dracula.

Fonte: Dracula Graph Library, 2013.

b. jQuery

É uma biblioteca javascript de código aberto com as principais funcionalidades: solução da incompatibilidade entre os navegadores; redução do código; utilização de *plugins* de terceiros; etc.

Além das tecnologias citadas, estabeleceu-se conexão direta com o banco de dados da aplicação Moodle para que se possa executar comandos sql para a execução de códigos para as principais funcionalidades da ferramenta.

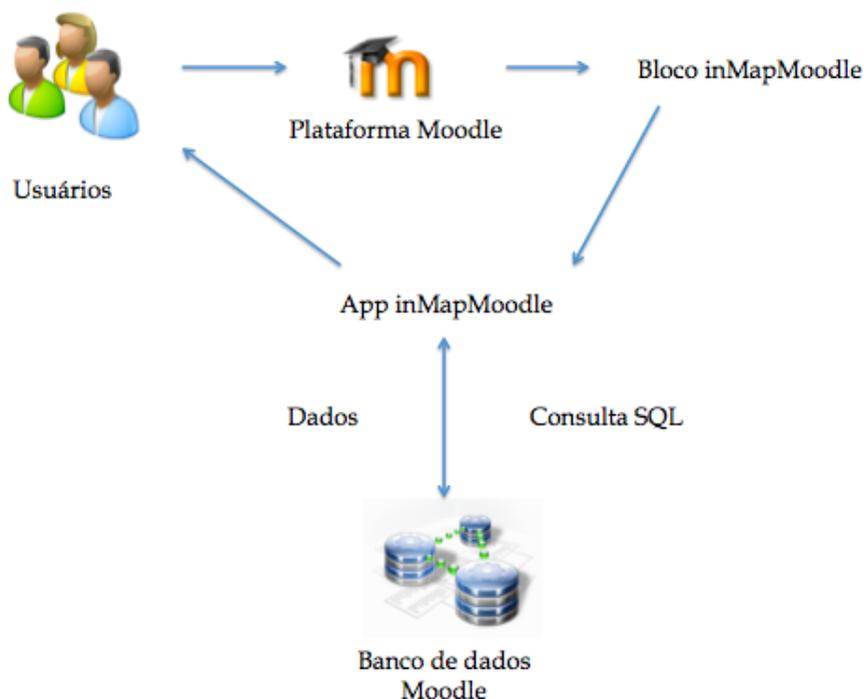


Figura 4.9 - Arquitetura da Ferramenta inMapMoodle©.

Fonte: FERRAZ et al., 2012

A ferramenta é disponibilizada por meio de um bloco para ser adicionado a página de cada disciplina.

A ideia principal da ferramenta é capturar o id do curso no banco de dados da plataforma Moodle, mostrar todos os fóruns disponíveis e permitir que o professor selecione a data e como deseja visualizar o mapa de interação.

A partir da seleção de um fórum, obtém-se o total de mensagens enviadas para se estabelecer o total (100%) e, a partir deste, criar a escala de cores baseadas no percentual de mensagens de cada participante, estabelecendo-se as seguintes faixas de percentual para a legenda:

- Muito Alta Interatividade (cor vermelha): de 41% a 100%;
- Alta Interatividade (cor azul): de 11% a 40%;
- Média Interatividade (cor verde): de 4% a 10%;
- Baixa Interatividade (cor amarela): de 1% a 3%;
- Nenhuma Interatividade (cor branca): 0%.

A partir dos cálculos do total de mensagem e total por participante, além dos registros de mensagem (quem enviou para quem), o sistema gera o mapa com balões e setas (Figura 4.10).

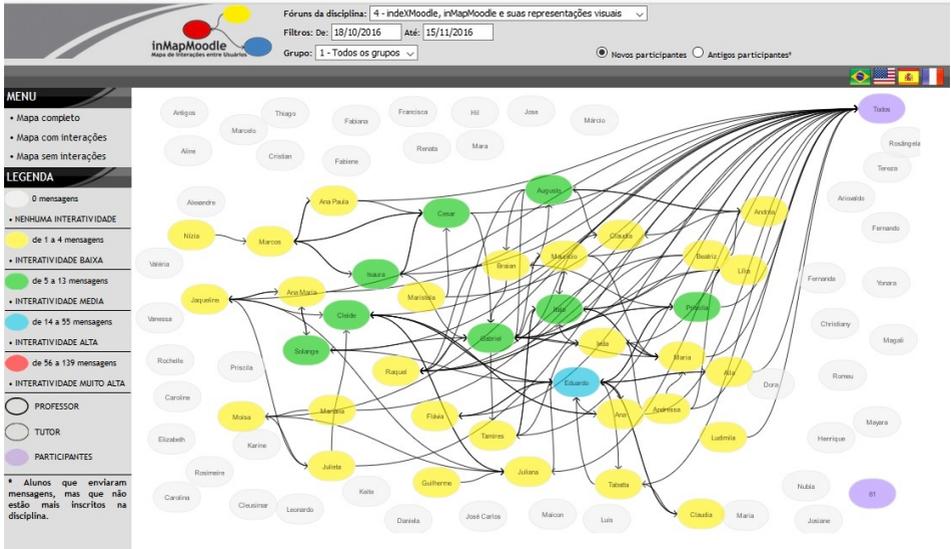


Figura 4.10 - Página inicial do inMapMoodle©.

Fonte: Tela capturada pelos autores com o inMapMoodle (FERRAZ et al, 2016)

Destaca-se que as linhas dos grafos apresentam setas para indicar o sentido da mensagem, o que gera indícios importantes sobre o processo de comunicação estabelecido no diálogo coletivo no fórum avaliado.

Na lateral inferior do fórum há o indicativo, no balão roxo, do total de mensagens postadas.

5.2 Características funcionais

A ferramenta inMapMoodle© possui várias características aprimoradas ao longo do desenvolvimento (FERRAZ; OLIVEIRA; HORNINK, 2015) e com foco na melhoria da avaliação da interatividade entre os participantes (Figura 4.11).

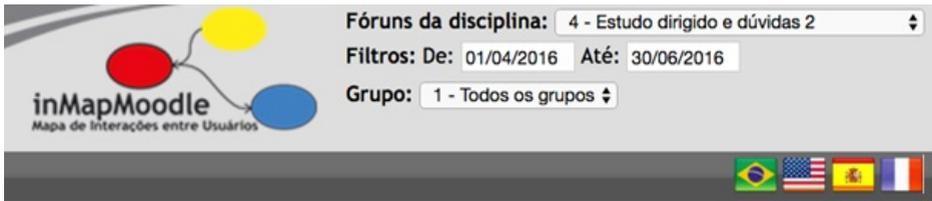


Figura 4.11 - Filtro inMapMoodle©.

Fonte: Tela capturada pelos autores do inMapMoodle (FERRAZ et al., 2016)

Entre as características principais estão:

I. Filtros

A ferramenta dispõe de quatro filtros para melhor auxiliar o professor na avaliação.

a. Por fórum: assim que a ferramenta é acessada automaticamente todos os fóruns da disciplina são listados para que o professor responsável possa escolher qual deseja avaliar. Os fóruns são colocados como ativos quando existe uma ou mais mensagens postadas e não ativo quando não existem mensagens.

Esse tipo de filtro permite que a ferramenta faça uma consulta das mensagens trocadas nesse determinado fórum.

b. Por data: filtro que permite limitar a busca por data no fórum escolhido. Dessa forma, o moderador pode compreender a dinâmica do fórum em todo período de realização ou por um período

de tempo específico (por dia, semana etc).

c. Por grupo: Esta funcionalidade foi adicionada somente na versão 2, a partir da qual pode-se selecionar um grupo específico (grupos criados dentro do curso pelo professor para organizar a turma) para geração dos mapas.

d. Novos e antigos participantes: filtro que permite ao professor gerar um mapa somente com novos participantes e que continuam inscritos no curso ou então gerar um mapa com antigos participantes do fórum, mas que porém não possuem mais inscrição na disciplina. As mensagens deixadas por antigos participantes no fórum continua sendo contabilizada, porém as suas interações só serão exibidas se o professor achar isso necessário. Esse tipo de filtro será necessário tanto para o cálculo da escala, quanto para a geração correta do mapa de interações.

II. Menu

O menu (Figura 4.12) oferece as seguintes opções:

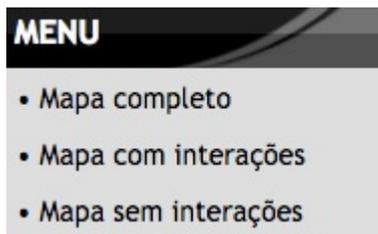


Figura 4.12. Menu do inMapMoodle©.

Fonte: Tela capturada pelos autores do inMapMoodle (FERRAZ et al, 2016)

a. Mapa completo: tipo de mapa que permite visualizar todos os participantes, inclusive os que não tiveram interações, dentro da data escolhida no filtro. Dessa maneira, os formadores têm uma visão completa de todos os participantes e os que exclusivamente trocaram mensagens no fórum selecionado e na data específica, possibilitando o cálculo de métricas avaliativas com relação a participação dos estudantes.

b. Mapa com interações: exibe somente o mapa com os participantes que possuem interações de acordo com a data escolhida no filtro. Por meio de um mapa único é possível visualizar a participação de todos, garantindo assim a visualização das interações entre os usuários participantes.

c. Mapa sem interações: exibe somente o mapa dos que não tiveram nenhuma participação

de acordo com a data escolhida no filtro.

III. Legenda

O inMapMoodle© apresenta legendas para o nível de interatividade (medido pela frequência de mensagens do indivíduo com relação ao todo) e para o tipo de participante (Professor ou tutor).

Apresenta-se na Figura 4.13 a legenda antes e após a seleção de um fórum, ilustrando a criação das faixas de interatividade calculadas automaticamente após seleção do fórum.



Figura 4.13 - Legenda inMapMoodle©

Fonte: Tela capturada pelos autores do inMapMoodle (FERRAZ et al., 2016)

IV. Mapa

O mapa de interações exibe a troca de mensagens entre estudante-estudante, estudante-professor, estudante-tutor e estudante-todos.

A ideia é ter o mapa representado por setas que mostram o direcionamento de quem mandou mensagem para quem (fluxo discursivo) e um “todos” que acaba sendo o ponto central de partida quando a mensagem é disparada para todos os usuários do curso.

Além disso, é exibido um número no balão participantes, representando a quantidade de participantes no determinado mapa mostrado. Por fim, quando o mouse passa sobre determinado balão, a função javascript *hover* é acionada permitindo visualizar o sobrenome cadastrado no banco de dados do determinado participante.

O mapa por si só é totalmente navegável, o que permite que o usuário arraste os balões para melhor visualizar a interação.

5.3 Aplicações da ferramenta

A ferramenta inMapMoodle© foi desenvolvida com intuito de ser algo extremamente poderoso na questão visual como métrica para inferir a participação entre

usuários de um fórum.

As principais aplicações da ferramenta são (FERRAZ; OLIVEIRA; HORNINK, 2015):

1. Permitir que, por meio do tipo de mapa gerado (completo, com interação e sem interação) e do cálculo de participação da legenda, a mesma possa servir como ferramenta de auxílio aos professores e/ou tutores na moderação de fóruns. Dessa maneira é possível acompanhar o andamento e a participação dos estudantes, de modo a conseguir avaliar a interação entre os usuários;
2. Permitir que os coordenadores de tutoria possam avaliar a participação dos tutores em discussões;
3. Mostrar um mapa da interação entre todos os participantes, tornando a ferramenta totalmente interativa.

6 Avaliação da ferramenta pelos tutores

Para avaliação da ferramenta, trabalhou-se com os oito tutores da disciplina de Bioquímica I do curso de Ciências Biológicas a distância (Unifal-MG), durante o primeiro semestre de 2013, os quais responderam a um questionário como método de validação inicial da ferramenta inMapMoodle©.

O questionário foi elaborado usando a escala de Likert para respostas e abordou as seguintes questões:

1. A ferramenta é de fácil acesso?
2. O mapa gerado é condizente com sua percepção da participação dos alunos no fórum escolhido?
3. A ferramenta mostrou resultados que poderiam servir como métricas avaliativas da discussão?
4. A escala criada (disposta na legenda) contribuiu para avaliar a participação dos alunos?
5. O mapa realmente possibilitou a visualização das interações entre os participantes?
6. Qual dificuldade de visualização do mapa gerado?
7. O mapa gerado lhe forneceu indícios das interações que poderiam ser utilizados em sua moderação com os alunos?

Baseou-se o questionário em questões que pudessem indicar a usabilidade da ferramenta e a viabilidade / aplicabilidade em usá-la como uma métrica de inferência das interações dos estudantes nos fóruns da plataforma.

Observa-se na Figura 4.14 que a maior parte dos tutores teve facilidade de utilização, o que seria um primeiro obstáculo na aplicabilidade, caso houvesse dados negativos.



Figura 4.14 - Indicação d facilidade de uso do sistema por tutores.
Fonte: Os autores.

Na questão (Figura 4.15), avaliou-se a percepção do tutor sobre correlação entre o que ele observou no fórum com o mapa, sendo os resultados indicaram ser condizente (88%), na maioria, ou muito condizente (13%), sem valores medianos ou negativos, o que reforça a aplicabilidade do sistema.

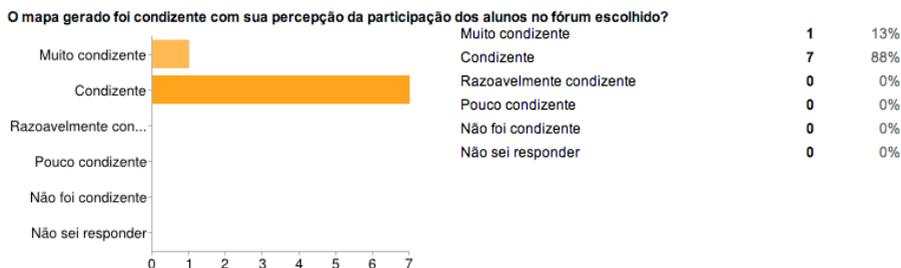


Figura 4.15 - Percepção da coerência entre o mapa e o fórum.
Fonte: Os autores.

Questionou-se sobre a escala de cores geradas automaticamente e o quanto ela contribuiu para avaliar os participantes (Figura 4.16), sendo que 75% indicou que contribuiu ou contribuiu muito, com apenas 25% indicando pouca contribuição.

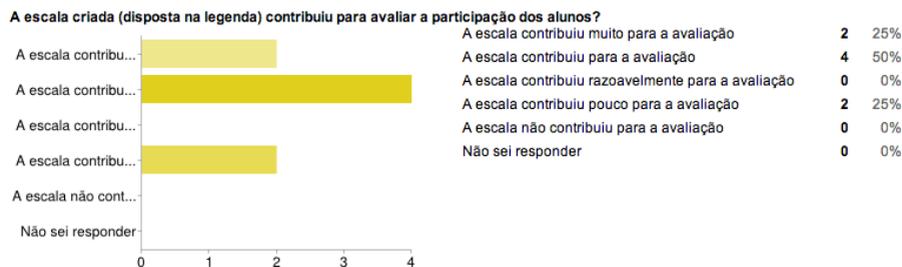


Figura 4.16 - Avaliação da contribuição da legenda inMapMoodle.
 Fonte: Os autores.

O questionamento sobre o uso dos mapas como métricas avaliativas (Figura 4.17) indicou que a maior parte dos tutores alegou que poderia servir (63%) ou poderia servir muito (25%), corroborando com as questões anteriores sobre a aplicabilidade do sistema.

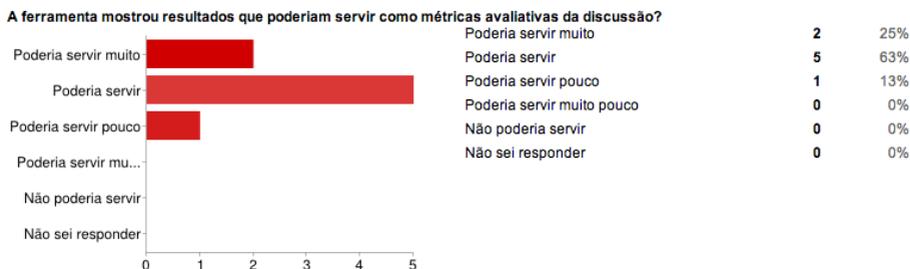


Figura 4.17 - Avaliação do inMapMoodle como métrica avaliativa.
 Fonte: Os autores.

A questão sobre o potencial de visualizar as interações dos participantes no mapa gerado foi importante (Figura 4.18), pois o tutor deve ter clareza de quem está conversando com quem e a dinâmica geral dos diálogos nos fóruns (fluxos dos enunciados).

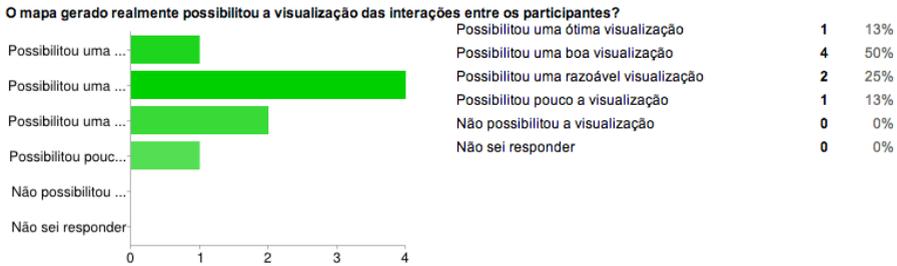


Figura 4.18 - Avaliação da indicação de participação nos mapas.
Fonte: Os autores.

Nesse sentido, 63% indicou boa ou ótima visualização e 38% razoável ou pouca visualização.

Em parte isso se deu pelo fato de, nesta avaliação, não existir o filtro grupo, assim, fóruns divididos por grupos eram visualizados em um único mapa, misturando os participantes dos grupos, dificultando a visualização. Isto foi corrigido na versão 2 do inMapMoodle.

Esse problema ficou ressaltado também nas respostas à questão abordando a dificuldade de visualização do mapa gerado (Figura 4.19), na qual os dados tenderam aos valores de normal (50%) e fácil (38%), mas sem valores negativos (dificuldade).



Figura 4.19 - Avaliação da dificuldade de visualização dos mapas.
Fonte: Os autores.

Além de saber se o mapa possibilitou a compreensão do fórum, era necessário saber se esses os mapas geravam informações que poderiam ser usadas na moderação dos fóruns pelos tutores (Figura 4.20).

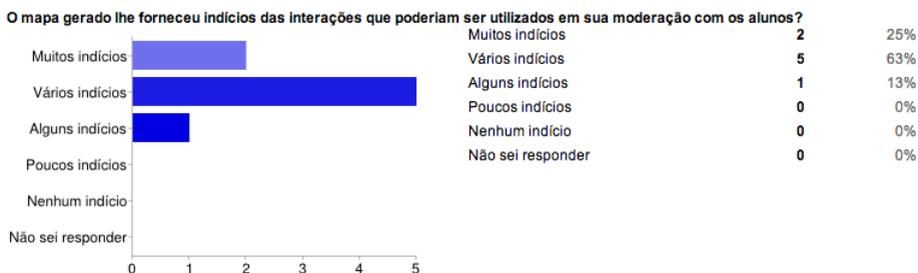


Figura 4.20 - Avaliação da aplicação dos dados para moderação.
Fonte: Os autores.

A maior parte dos tutores indicou que forneceu vários indícios (63%), seguido de muitos indícios (25%), o que, de certa forma, vai de encontro com os objetivos do aplicativo que seria de fornecer informações visuais sobre as relações nos fóruns de discussões que viabilizassem meios para melhorar a moderação pelos

tutores dos fóruns online.

Exemplos disso seria compreender as dinâmicas dos grupos de diálogo, verificando se há domínio de algum aluno pelo diálogo ou mesmo alunos que estão ficando de fora da discussão ou sem retorno de suas mensagens.

Com base nas respostas é possível concluir que a maior parte dos tutores não tiveram problemas com relação a visualização do mapa e a percepção das interações geradas e a usabilidade da ferramenta foi comprovada, tendo em vista que a porcentagem inicial apresentada com relação a facilidade de uso foi acima do esperado.

Os filtros de data possibilitaram gerar mapas não tão complexos, uma vez que continham um menor número de conexões, apresentando dados condizentes e válidos com a discussão vigente.

Além disso a legenda permitiu contribuir na maioria dos casos testados, mostrando assim total eficiência e dados necessários para avaliar o principal objetivo proposto.

7 Considerações finais

Elaborou-se presente trabalho com o objetivo de apresentar a ferramenta inMapMoodle© e os resultados obtidos para a validação da mesma, focando, principalmente, em apresentar os principais ambientes virtuais de aprendizagem, uma breve discussão sobre os fóruns online e a importância das aplicações em grafos no meio computacional.

A apresentação dos princípios de desenvolvimento e de como a ferramenta desenvolvida contribui para a melhoria na educação a distância foi abordada como tema principal junto aos resultados obtidos por um questionário realizado com os tutores de uma disciplina específica.

As plataformas por si só oferecem diversas ferramentas de comunicação que facilitam a troca de mensagens e interação entre os participantes. A ferramenta inMapMoodle© se apresenta como uma interface e demonstra dados que antes pareciam ser complexos se somente visualizados via respostas dos fóruns na plataforma Moodle.

A partir da análise dos dados do questionário, obteve-se indícios dos benefícios gerados pela ferramenta e que a mesma se propôs a realizar seu

objetivo de promover a visualização da interação entre participantes e permitir que o mapa e legenda gerados possam servir como métricas de avaliação.

Agradecimentos

À Capes/ UAB pelo apoio financeiro.

Referências

ALMEIDA, R. R. P., SIEBRA, S. A. Um Estudo Comparativo dos Ambientes de Aprendizagem Colaborativa usados no Brasil Segundo o Modelo 3C. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 23., 2007, São Paulo. *Anais...* São Paulo: SBC, 2007. p. 103-106. Disponível em: <<http://dx.doi.org/10.5753/cbie.sbie.2007.103-106>>. Acesso em 06 set. 2018.

BARROSO, M. M. A., Operações Elementares em Grafos e Aplicações. In: ENCONTRO REGIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 7., 2007, Uberlândia. *Anais...* Uberlândia: SBMAC - UFU, 2017. p.28.

BEHAR, P. A et al. Em busca das interações interindividuais no ROODA. *Revista Educação*, Porto Alegre, n. 1, p 169-199, jan./abr. 2004. Disponível em: <<http://revistaseletronicas.pucrs.br/faced/ojs/index.php/faced/article/view/378>>. Acesso em: 06 set. 2018/

BRASIL. *Referenciais para elaboração de material didático para EaD no Ensino Profissional e Tecnológico*. Brasília: Ministério da Educação e Cultura, 2007. Disponível em: <http://ltc-ead.nutes.ufrj.br/vivencias/recursos/45ref_materialdidatico.pdf>. Acesso em: 20 jun. 2018.

CORMEN. T. et al. *Introduction to Algorithms*. Cambridge: MIT Press, 2001.

DRACULA GRAPH LIBRARY. Disponível em: < <http://www.graphdracula.net/showcase/>> Acesso em 10 jan. 2013.

FERRAZ, P. F. O. et al. *InMapMoodle*, Versão 2. Alfenas: Universidade Federal de Alfenas, 2016.

FERRAZ, P. F. O. et al. *InMapMoodle*, Versão 1. Alfenas: Universidade Federal de Alfenas, 2012.

FERRAZ, P. F. O.; OLIVEIRA, P. T.; HORNINK, G. G. Desenvolvimento e implementação de indicadores de colaboração e participação no Moodle. *Informática na Educação (Online)*. v. 18, p. 85-95, 2015. Disponível em: <<http://seer.ufrgs.br/index.php/InfEducTeoriaPratica/article/view/51508>>. Acesso em: 31 jul. 2018

GABARDO, P.; QUEVEDO, S.; ULBRICHT, V.R., *Estudo Comparativo das Plataformas de Ensino-Aprendizagem*. Florianópolis: EGC/UFSC, 2010.

GIORDAN, M. *Computadores e linguagens nas aulas de ciências: uma perspectiva sociocultural para compreender a construção de significados*. Ijuí: Ed. Unijuí, 2008.

HORNINK, G. G. *Cartografando online: caminhos da informática na escola com professores que elaboram conhecimentos em formação contínua*. 2010. 296f. Tese (Doutorado em Ciências) - Instituto de Geociências, Universidade Estadual de Campinas, Campinas, SP, 2010. Disponível em: <<http://repositorio.unicamp.br/jspui/handle/REPOSIP/287024>>. Acesso em: 31 jul. 2018.

LUCCHESI, C. L., *Introdução à Teoria dos Grafos*. Rio de Janeiro: IMPA-CNPq, 1979.

MOODLE STATS. Disponível em: < <https://moodle.org/stats>>. Acesso em: 23 jan.2013

MOORE, M.; KEARSLEY, G. O ensino e os papéis do instrutor.

In: _____. *Educação a distância: uma visão integrada*. São Paulo: Thompson Learning, 2007, pp. 147-17.

ROCHA, H. V. et al. Avaliação Online: O modelo de suporte tecnológico do Projeto TelEduc. In: SILVA, M.; SANTOS, E. O. *Avaliação em Educação Online*, São Paulo: Edições Loyola, 2006. p. 347-368. p. 55

SIMÕES, J. A. A.; LINDEMANN, V. Editor de Textos Colaborativo Integrado ao Teleduc. *RENOTE - Revista Novas Tecnologias na Educação*, Porto Alegre, v. 5, n.1, 2007. p.

PEREIRA, C. S. et al. *A utilização de ambientes virtuais como ferramenta de apoio ao ensino presencial: estudos preliminares e proposta de metodologia de implantação no CEFET-BambuÍ*. Bambuí, MG, 2008. < Disponível em: <https://www.bambui.ifmg.edu.br/jornada_cientifica/str/artigos_aprovados/informatica/61-CO-5.pdf>. Acesso em: 06 set. 2018.

PETERS, O. *Didática do ensino a distância*. Ed. Unisinos, RS, 2006.

ROCHA, H. V. et al. Projeto TelEduc: Pesquisa e Desenvolvimento de Tecnologia para Educação a Distância. In: CONGRESSO INTERNACIONAL DE EDUCAÇÃO A DISTÂNCIA DA ABED, 9., 2002, São Paulo. Anais... São Paulo: ABED, 2002. p. 01-72. Disponível em: <http://www.teleduc.org.br/sites/default/files/publications/prmio_abed2002.pdf>. Acesso em 06 set. 2018.

SILVA, M. *Educação online*. São Paulo: Loyola, 2006.

TelEduc. Disponível em: <<http://teleduc.org.br/>>. Acesso em: 29 jan. 2013. <http://www.teleduc.org.br/?q=content/tutorial-intermap>

Capítulo 5 O uso de sistemas lúdicos de autoria no ensino-aprendizagem de programação

*Luis Eduardo de Matos
Augusto Márcio da Silva Júnior
Gabriel Gerber Hornink*

A crescente inserção das novas tecnologias digitais no cotidiano das pessoas tem modificado acentuadamente a forma de lidar com diversos problemas sociais, como por exemplo a utilização dessas ferramentas nos processos de ensino-aprendizagem. O cotidiano escolar traz, cada vez mais, estudantes com seus *smartphones* e desinteressados nos métodos tradicionais de ensino. Nesse contexto, apresenta-se a experiência da construção, aplicação e avaliação de uma unidade instrucional online, onde conceitos do pensamento computacional são postos em prática em atividades lúdicas a partir de ferramentas gratuitas, como as plataformas *code.org* e *Scratch*, para programação em blocos (*Blockly*). Para tal, a unidade instrucional online foi projetada no ambiente virtual de aprendizagem Moodle, realizando-se discussões em fóruns, desenvolvimento de projetos, sendo possível coletar informações referentes a participação dos estudantes por meio dos registros do Moodle, além de questionários pré e pós atividade. Identificou-se que as ferramentas utilizadas proporcionam boas alternativas para a inserção de mídias digitais no ensino. Dessa forma, busca-se estreitar a relação educador/educando favorecendo a autoria dos mesmos na produção de seus próprios materiais e conhecimentos.

1 Introdução

Os avanços no desenvolvimento de tecnologias digitais de comunicação e informação (TDICs), fruto material intrínseca do desenvolvimento da sociedade, oportunizou novas formas de lidar com as questões sociais, principalmente a educação, abrindo possibilidades para novos processos de interação humana (SILVA JR; HORNINK, 2017). Recentemente, com o crescente uso desses dispositivos eletrônicos, os estudantes têm a possibilidade de ter autoria sobre seu próprio material, de modo ativo, interativo e colaborativo na “palma de sua mão”.

As temáticas envolvendo ensino-aprendizagem e a relação educador/educando fazem parte da história da sociedade, ao ponto de levantar debates e discussões que englobam, em sua ideia central, como maximizar o aprendizado conjunto dos estudantes do ensino básico e diminuir a evasão causada por diversos motivos, dentre eles a falta de atratividades na escola.

Por muitos anos, o professor, como figura central da sala de aula, foi o detentor do conhecimento. Este era quem “transmitia” o conhecimento aos alunos de variadas formas, prioritariamente com métodos

tradicionais que na maioria das vezes não mostravam-se como uma alternativa interessante para uma parcela significativa dos estudantes (BULGRAEN, 2010).

Para mudar essa rotina, os profissionais da educação tentam compreender os motivos que levam a esse bloqueio, buscando novos modos de utilizar o tempo na sala de aula de forma mais produtiva, criativa e inclusiva, levando em consideração as dificuldades de cada estudante e sabendo que o mundo conectado em rede gerou novos meios de interação entre as pessoas na sociedade (BRAIT, 2010).

Com o avanço das tecnologias digitais e a ampliação do acesso aos dispositivos móveis, uma quantidade significativa dos estudantes possuem acesso à internet e a novas tecnologias, sendo que até o mês de maio de 2018 no Brasil, havia 220 milhões de *smartphones*, e 86 milhões de computadores portáteis, o que corresponde a cerca de 1,5 dispositivo portátil por habitante (MEIRELLES, 2018). Este valor significativo impacta, de certa forma, sobre o inter-relacionamento pessoal e conseqüentemente a forma de se aprender e ensinar novos conteúdos.

Com isso, percebe-se que o papel do professor na educação atual segue para caminhos diferentes. No

contraste entre a digitalização e as falhas nos métodos tradicionais, o professor se vê na necessidade de criar práticas mais interativas e eficazes, visto que as tecnologias transformaram vários setores da sociedade e, com isso, os processos de ensino-aprendizagem demandam se adequar às mudanças, gerando um impasse à educação, fazendo-se necessário

[...] evoluir para tornar-se relevante e conseguir que todos aprendam de forma competente a conhecer, a construir seus projetos de vida e a conviver com os demais. (MORAN, 2015, p.1)

Logo, sabendo-se que o acesso ao conhecimento está muito mais próximo da população nos dias de hoje, o objetivo do professor se torna, cada vez mais, o de ser um mediador da relação conhecimento/aprendizagem (BULGRAEN, 2010).

Apesar do alto acesso às informações, faz-se necessário destacar que não basta o acesso, mas, principalmente, saber como, onde e o que buscar, caso contrário o estudante poderá se perder em um mar de informações superficiais ou com erros.

É neste ponto que as novas tecnologias educacionais podem contribuir nas atividades de professores e estudantes, colocando-os de frente a uma nova realidade que pode potencializar os processos de

ensino-aprendizagem, podendo motivar os estudantes e ampliar a construção de conhecimentos de modo ativo, interativo e colaborativo, fazendo importante compreender como esses instrumentos culturais de mediação podem potencializar os processos educacionais, com o foco nas interações sociais (HORNINK; COMPIANI, 2017).

Além disso, as novas tendências para o uso da informática no ensino apontam para o estudante como autor de seu próprio conhecimento, marcando o caminho para a inclusão digital e para a autonomia do estudante (VON WANGENHEIM, 2014).

Experiências como de Alves et al., (2016) mostram que o estudante como autor de seus próprios aplicativos pode ocorrer nos mais diversos contextos, como no ensino de História, no ensino fundamental.

Dessa forma, é necessário ter uma visão progressista e também crítica diante dos métodos tradicionais que foram passados nos últimos anos e encarar uma nova realidade tecnológica digital.

Destaca-se que o simples uso da informática não garante um melhor processo de ensino-aprendizagem, mas que uma mudança na concepção do ensino se faz necessária, sendo que, nem sempre será necessária uma

tecnologia digital para garantir um bom processo educacional.

Aliado a isso, a inclusão digital é um fator de inclusão social, quebrando as barreiras socioeconômicas impostas por nosso sistema. Assim, é possível utilizar-se dos espaços que a internet proporciona agora e, desta forma, inserir os jovens como protagonistas da sociedade do conhecimento (DIAS, 2011).

Apresenta-se neste texto uma experiência do projeto de ensino-pesquisa-extensão “Pensando em Códigos”¹⁹, o qual foca em sistemas de autoria para construção de aplicações web, no sentido do desenvolvimento de habilidades sociais, afetivas e cognitivas envolvendo o pensamento computacional por meio da programação por blocos (*Blockly*), de forma lúdica e criativa, usando sistemas como code.org e Scratch.

Busca-se apresentar e discutir o modo de estruturação das atividades online no ambiente virtual Moodle, assim como apresentar indícios para compreender a aplicação das atividades, visando a formação do estudante autor.

19Website: <http://www.unifal-mg.edu.br/lme/projetos/pec>

2 Reflexões teóricas

2.1 Uso das TDICs no ensino

As Tecnologias digitais da informação e comunicação (TDICs) são ferramentas essenciais para as principais atividades sociais da chamada nova era digital. O computador é uma das ferramentas em que uma das funções é transmitir e receber informações e, portanto, servir como um comunicador (VALENTE, 1993), ou seja, instrumento sociocultural que medeia as relações humanas.

De acordo com Valente (1993, p. 9),

“O computador pode ser usado também como ferramenta educacional. Segundo esta modalidade o computador não é mais o instrumento que ensina o aprendiz, mas a ferramenta com a qual o aluno desenvolve algo, e, portanto, o aprendizado ocorre pelo fato de estar executando uma tarefa por intermédio do computador”.

O cotidiano dos estudantes está repleto de novas e variadas tecnologias, tornando essa interatividade digital um obstáculo às sessões tradicionais de ensino que mostram-se insuficientes para motivá-los ao estudo. Contudo o ensino apoiado pelo uso de dispositivos digitais pode sobrepor as abordagens obsoletas (MASCHIO, 2015).

Logo, o aprendizado adquirido por meio da

construção de ideias apresenta um favorável potencial lúdico, uma vez que o estudante se torna o autor do seu próprio material e conhecimento, promovendo o aprendizado, interesse, autonomia e despertando a criatividade do aluno (FREIRE 1996), ressaltando-se que há diversos recursos digitais que podem ser utilizados de forma gratuita.

Diversos autores (ALMEIDA, 2001; HORNINK, 2010; MERCADO, 2002; MORAES, 2000; SILVA JR, HORNINK, 2017) reconhecem as potencialidades da área de Informática, concordando que os computadores podem favorecer sobremaneira ao processo de educação escolar, visto que as tecnologias digitais estão cada vez mais presentes no cotidiano e podem ser usadas para potencializar os processos de ensino-aprendizado.

Aliado a isso, o ensino de computação impulsiona a criação de empregos por meio da inovação dos processos na nossa economia. O conhecimento em tecnologias da informação é um requisito básico para que os estudantes possam garantir a sua inserção no mercado de trabalho, independente da área de estudo ou do setor profissional escolhido (VON WANGENHEIM, 2014).

Além disso, de acordo com Silva Jr. e Hornink (2017), possibilita trabalhar com os estudantes o

desenvolvimento de raciocínio lógico-matemático, além de que, corroborando com Alves et al. (2016), indicam conceitos de abstração, recursão, interação, desenvolvendo habilidade de explorar o uso de linguagens de programação para resolução de problemas, comunicação, trabalho em equipe, além dos conhecimentos específicos das disciplinas trabalhadas no contexto de desenvolvimento.

Entretanto, para que as novas ferramentas sejam usadas de forma eficiente e o processo traga melhorias significativas, é necessário uma interação efetiva entre os envolvidos, tomando como base a figura do professor mediador.

Segundo Ramos (2014, p.7),

[...] os desafios da escola moderna face às aceleradas mudanças neste campo, deverá corresponder igual esforço das instituições de formação inicial assegurando que os novos professores estejam suficientemente preparados para as suas missões de ensino, incluindo uma formação científica e pedagógica sólida, moderna e atualizada.

Em contrapartida, um novo modelo educacional com uso de tecnologias digitais gera uma motivação à aprendizagem, tendo em vista que os aparelhos eletrônicos já se tornaram bastante populares e acessíveis mesmo a pessoas de baixa renda. De acordo

com Moran (2000, p. 59), “Com a internet podemos modificar mais facilmente a forma de ensinar e aprender tanto nos cursos presenciais como nos à distância”.

As ferramentas digitais apresentadas neste trabalho, Code.org e Scratch, indicam novas alternativas de se trabalhar os processos de ensino-aprendizado, oferecendo oportunidades de visualizar fenômenos científicos e conceitos abstratos de forma lúdica, colaborativa e interativa.

2.1 Inclusão digital

A globalização dos instrumentos tecnológicos digitais fez com que a sociedade desenvolvesse novas formas de relacionamento e comunicação que impactam a sua estrutura social/econômica (HORNINK, 2010). Nesse ponto, ressalta-se a importância da inclusão social por meio da utilização de dispositivos tecnológicos que fazem parte do cotidiano da nova sociedade digitalizada.

Inclusão digital é possibilitar ao indivíduo o uso das tecnologias para ampliar o desenvolvimento, potencializar sua autonomia, aumentar sua liberdade e aprimorar a democracia (SILVEIRA, 2011). É, portanto, uma forma de incluir socialmente o indivíduo, tendo em vista a nova imagem de uma sociedade informacional

que estabelece suas relações por meio do uso de tecnologias digitais e da internet.

Para Dias (2011, p. 62),

“Com o advento da internet e a popularização do computador, começaram a surgir na década de 1990, em países pobres, programas de inclusão digital para populações excluídas do acesso à comunicação mediada pelo computador”.

Contudo, apesar da crescente expansão tecnológica e das quedas consideráveis no preço dos equipamentos, as desigualdades socioeconômicas constituem a grande barreira para a comunicação em rede (SILVEIRA, 2008), além do modelo de banda larga no país, que acaba por não suprir a demanda de internet.

Segundo Silveira (2011, p. 49),

“No Brasil, a expressão “exclusão digital” passou a caracterizar o fenômeno das barreiras socioeconômicas, colocadas diante da maioria da população, para uso das tecnologias da informação, desde o final dos anos 1990”.

Logo, sabendo desse bloqueio imposto pela questão econômica, busca-se ampliar os limites da inclusão social/digital de forma a incluir também aqueles que não podem financiar o acesso às ferramentas.

De acordo com Dias (2011, p. 63), “Os laboratórios de informática são, para milhões de crianças e jovens, o

primeiro contato com o computador e a internet”. Se de fato os meios de comunicação são essenciais para a inclusão do indivíduo na sociedade, o poder comunicacional é cada vez mais realizado pelo acesso e uso pleno das tecnologias digitais.

Assim, tendo foco na cidadania e não somente na modernização, a rede torna-se um meio de participação ativa do indivíduo na sociedade. Castells (2009) sustenta que os meios de comunicação tornaram-se o espaço social onde o poder é decidido.

Por fim, o uso dos computadores em rede possibilita a distribuição do poder da informação entre as pessoas, favorecendo a ampliação da autonomia dos indivíduos e da sociedade.

Com isso, inclusão digital autônoma muda as exigências e o cenário político, uma vez que o espaço social dos debates é cada vez mais aberto e visível a todos indivíduos conectados (SILVEIRA, 2011).

2.2 Sistemas de autoria

O uso da informática na educação deve ser fundamentado em teorias e práticas que enfatizem o processo de construção do conhecimento pelo próprio estudante, buscando extrapolar a lógica da didática dos

instrumentos culturais não digitais.

O uso de dispositivos eletrônicos e sistemas de autoria na educação pode proporcionar, além do contato com novas tecnologias, a ampliação de estratégias para o desenvolvimento de habilidades cognitivas, expressão do pensamento, criação e imaginação, possibilitando a criança tornar-se criadora de conhecimento e não apenas receptora de informações (PEREIRA, 2005).

Os softwares de autoria devem ser pertinentes ao ambiente escolar, bem como a ideologia proposta, fácil de ser utilizado pelos envolvidos, servindo como um facilitador da construção do conhecimento por meio do desenvolvimento de projetos digitais lúdicos. Esses ambientes de aprendizagem devem fornecer conexões individuais e coletivas, permitindo o desenvolvimento de tarefas vinculadas à realidade do estudante, integrando diferentes áreas do conhecimento e favorecendo a multidisciplinaridade (PINTO, 2012).

Desta forma, Pinto (2012, p. 139) afirma que:

[...] o ambiente criativo deve propiciar ferramentas que viabilizem a autoria e a manifestação de ideias inovadoras permitindo o desenvolvimento de um espaço de aprendizagem que possibilite a manipulação de objetos e ideias e a interação dos estudantes entre si e com os professores.

Essa possibilidade propicia a manifestação da

criatividade, tendo como fator principal a relação educador/educando como forma de estimular e desafiar o estudante na busca do seu desenvolvimento.

Em Pereira (2005), apresenta-se o uso de um ambiente de autoria chamado LEGAL. Este ambiente permite que a criança possa expressar sua criatividade e imaginação para criar e contar histórias com funcionalidades simples e intuitivas.

A ferramenta foi aplicada em uma turma de crianças com idade entre quatro e sete anos, e se mostrou eficaz ao despertar o interesse e a criatividade dos envolvidos. Segundo Pereira (2005, p. 8):

“Esta ferramenta traz em si os benefícios de um ambiente de autoria, criando ambientes virtuais tridimensionais e a possibilidade de utilizar ambientes imersivos de múltiplas projeções”.

Portanto, o processo de aprendizagem é facilitado pela execução das tarefas propostas de forma lúdica, e pela busca por soluções eficientes e o conhecimento resultará da relação do sujeito com o objeto, assim como do sujeito com outros sujeitos, sendo que, mesmo na relação com o objeto, ele está se relacionando com outros sujeitos, entretanto, mediado pelo objeto que este criou, no entendimento deste como um instrumento cultural de mediação.

3 Materiais e métodos

Esta pesquisa foi de natureza aplicada, com objetivos exploratórios e descritivos, sendo caracterizada, principalmente, com procedimentos experimentais, com dados quanti-qualitativos. O objetivo principal foi o desenvolvimento, aplicação e avaliação de uma unidade instrucional online para o ensino de conceitos iniciais de programação.

Realizou-se um mapeamento prévio de trabalhos que utilizaram ferramentas lúdicas de autoria para auxílio ao ensino/aprendizagem e identificou-se que o Code.org e o Scratch apresentavam as melhores alternativas para a construção do curso, devido principalmente a facilidade no uso das mesmas, além da possibilidade de serem aplicadas no próprio navegador e serem gratuitas e com um grande número de tutoriais no youtube.

A unidade instrucional *online* foi ofertada por meio do Ambiente Virtual de Aprendizagem Moodle, de forma livre e gratuita, portanto, aberta a toda comunidade e sem necessidade de conhecimento prévio.

Construiu-se a unidade educacional com aumento crescente da dificuldade, inserindo gradativamente elementos da programação no Scratch, sendo que em

cada unidade os participantes tinham acesso à videoaulas, exercícios de correção automática e pequenos projetos para desenvolver no Scratch e postar na plataforma, além do fórum de dúvidas sobre o projeto e demais atividades. Ao fim do curso, o sistema de avaliação foi por pares, ou seja, cada aluno avaliou dois trabalhos de outros colegas para receber sua nota.

Os participantes eram das mais variadas idades, desde alunos do ensino fundamental até professores de escolas públicas e privadas. Contudo, verificou-se a maior parte dos concluintes foram educadores que buscavam adquirir conhecimento em novas tecnologias que pudessem ser utilizadas na sala de aula, auxiliando os processos de ensino.

Ofertou-se unidade instrucional em duas ofertas distintas, sendo que na primeira oferta (nov. 2016 - jan. 2017) se utilizaram as plataformas Code.org e Scratch, contando com 80 inscritos. Na segunda oferta se utilizou apenas da plataforma Scratch, sendo aplicada entre Abril e Maio de 2017, contando com 319 inscritos.

Os procedimentos para construção e avaliação das atividades desenvolvidas serão apresentadas a seguir.

3.1 Levantamentos prévios

Os levantamentos prévios tiveram como objetivo conhecer o que existe no Brasil sobre o uso de ferramentas digitais de autoria e, assim, identificar as possibilidades que cada uma poderia agregar a construção e aplicação da unidade instrucional online.

Segue no apêndice 5.A um quadro com alguns softwares de autoria encontrados por buscas feitas nos mecanismos de pesquisa do Google. Destaca-se que a um grande número de aplicativos da lista utiliza o sistema de blocos (blockly) para construção dos códigos.

Além disso, fez-se um levantamento dos cursos abordando o Scratch e Code.org, no período de 2008 à 2016 (Apêndice 5.B), encontrando-se 71 cursos e, com relação ao público-alvo (alguns cursos tinham mais de um público-alvo): 29 para o ensino fundamental; 21 para o ensino médio; 8 para o ensino técnico; 11 para o curso superior; 8 para educadores.

Destaca-se também que a maior parte dos cursos apresentaram propostas sociointerativas (96%), o que significam propostas nas quais os participantes interagem entre si, além da interação com as ferramentas, para o desenvolvimento dos conhecimentos sobre o Scratch ou Code.org.

Como exemplo de cursos abordando o Scratch, no período que se iniciou a primeira oferta, encontravam-se abertos os três cursos indicados no Quadro 5.1.

Quadro 5.1 - Cursos de Scratch abertos no período da realização do projeto (nov. 2016 - jan. 2017).

Instituição	Ferramenta	Público	Custo	Tipo	Horas
PUC-SP	Scratch/ Arduino	Educadores	Não	Presencial/ Interativo	8
IFES	Scratch	Graduação/ Técnico	Sim	EAD/ Moodle	45
Unifenas	Scratch	Geral	Sim	EAD/ Autoguiado	4

Fonte: Os autores

3.2 Scratch e Code.org

As mudanças sociais causadas pela expansão tecnológica do século atual apontam como uma nova tendência mundial a inclusão do ensino de computação nas fases escolares iniciais (ALVES, 2016).

Diversas iniciativas surgem nesse sentido, como, por exemplo, o uso do Code.org e do Scratch no ensino de crianças e adolescentes.

As ferramentas de desenvolvimento de aplicativos, como por exemplo o App Inventor²⁰ (Figura 5.1), criado pela Google em 2010 e vendido ao MIT em 2013, são de modo geral de fácil entendimento para aqueles que não estão familiarizados com os conceitos de programação, 20Website do sistema <http://appinventor.mit.edu>

uma vez que são construídos para uso intuitivo, baseado em cores, encaixes e semânticas.

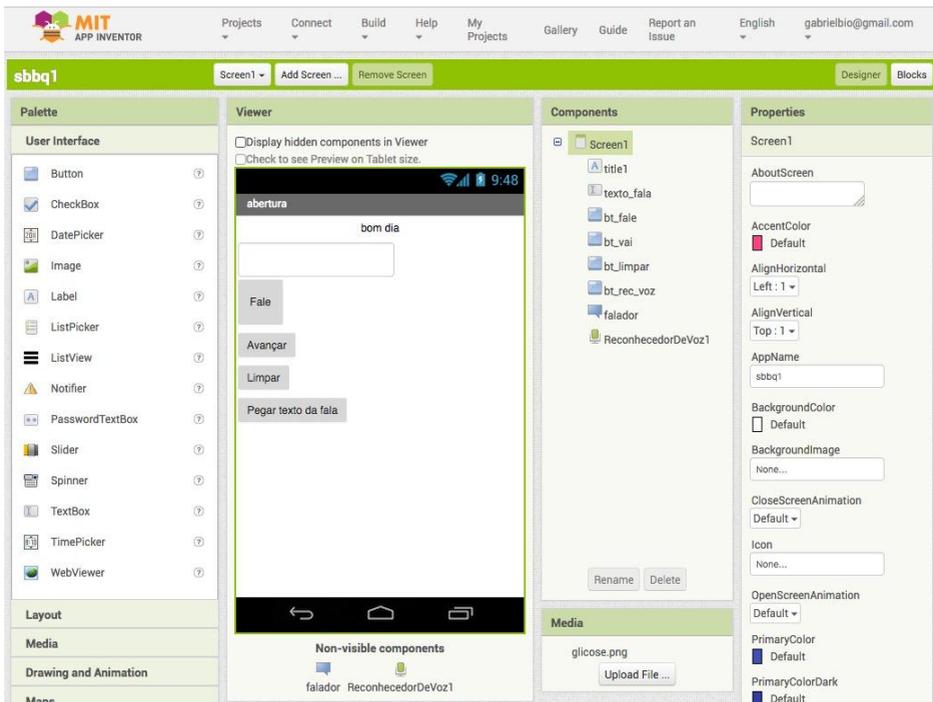


Figura 5.1 - Tela principal do App Inventor.

Fonte: Tela do App Inventor capturada pelos autores.

Para Finizola (2014, p. 337), “O ensino de programação é importante, e não deveria ser uma atividade restrita apenas a estudantes de computação”.

Diversos trabalhos apontam para a importância da inserção do pensamento computacional e da lógica de programação para o desenvolvimento de alunos em idade escolar, como dito por Scaico et al., (2013, p.93) “este tipo de educação permite o desenvolvimento de diversas capacidades que contribuem para melhorar o

raciocínio lógico dos estudantes”.

Além disso, o uso de ferramentas de desenvolvimento de aplicativos no ensino pode contribuir para apresentar ao estudante a oportunidade de usar a sua tecnologia disponível para outras atividades além daquelas que estava acostumado a realizar. Aliado a isso, apoia-se o desenvolvimento de habilidades de raciocínio lógico e resolução de problemas (FINIZOLA, 2014; SILVA JR, HORNINK, 2017).

O Code.org (Figura 5.2) é uma organização sem fins lucrativos criada em 2013 cujo objetivo é divulgar e disseminar o ensino de programação para todo tipo de pessoa e idade, dando oportunidade para que escolas tenham um material interativo disponível sobre a área.

A partir das informações do *website* (CODE.ORG, 2018), tem-se que a instituição acredita que o ensino de computação deve fazer parte do currículo escolar e a mesma possui parcerias com vários “gigantes da tecnologia”, tendo algumas das aulas ministradas por personalidades da área como Mark Zuckerberg e Bill Gates, assim como algumas das atividades envolvem desenhos e jogos conhecidos, como Frozen, Minecraft, Star Wars, Angry Birds, como forma de incentivo a realização das atividades.

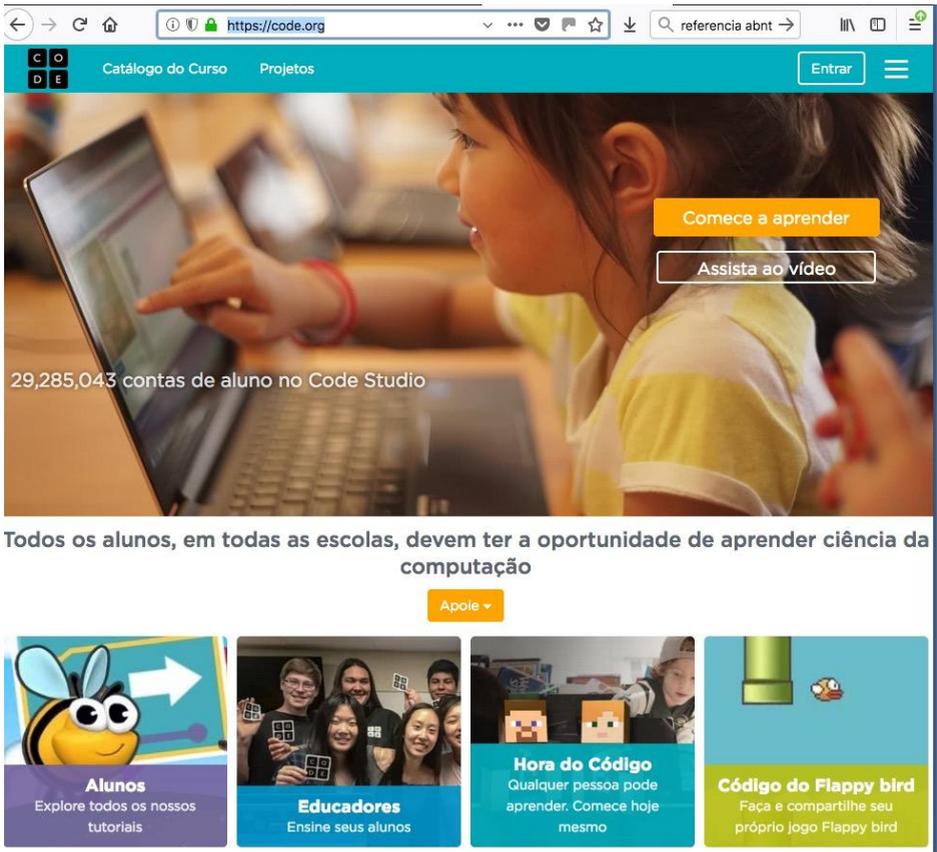


Figura 5.2 - Tela principal do website do projeto Code.org.
Fonte: Tela do Code.org capturada pelos autores.

Com uma plataforma de desenvolvimento online, disponível na página web²¹, a “Hora do Código” no Brasil pretende mudar a ideia tradicional de que programação seja algo muito avançado e difícil, permitindo que pais, professores e alunos aprendam programação de forma intuitiva e divertida.

Durante o primeiro módulo da primeira oferta do 21Página do projeto: <https://br.code.org>

curso, utilizaram-se algumas atividades da “Hora do Código” para introduzir os envolvidos aos conceitos básicos de programação, assim ao modo de se construir o código no code-org, que é similar ao Scratch, uma vez que ambos utilizam o *blockly*.

O Scratch foi desenvolvido pelo Lifelong Kindergarten Group no MIT Media Lab em 2007 e hoje se apresenta em vários idiomas, alguns distribuídos como software gratuito (Scratch) ²², outros como software livre (Snap)²³. É um sistema que pode ser utilizado para os primeiros contatos com conceitos iniciais de raciocínio lógico por intermédio da programação.

Com o Scratch (Figura 5.3) é possível criar personagens que dançam, cantem e interagem uns com os outros, apenas arrastando e soltando blocos que se encaixam para formar linhas de programação (sistema conhecido como *Blockly*).

Esses blocos representam os componentes do programa, como expressões booleanas, loops, condicionais e variáveis. A plataforma possui também uma rede social com projetos para compartilhamento.

22Página do Scrtach <http://scratch.mit.edu>

23Página do Snap: <http://snap.berkeley.edu>

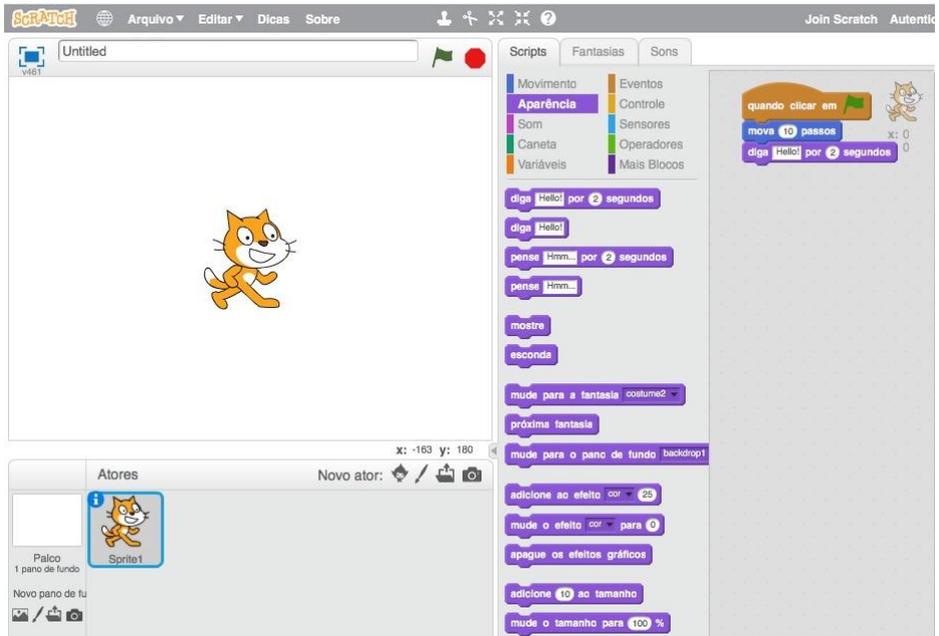


Figura 5.3 - Tela principal do Scratch.

Fonte: Tela do Scratch capturada pelos autores.

O Scratch possui uma comunidade de aprendizagem criativa presente em mais de 150 países com mais de 11 milhões de usuários e quase 4 milhões de projetos compartilhados.

Diversas pesquisas mostram que o uso do Scratch contribui positivamente no ensino de computação nas escolas (VON WANGENHEIM, 2014), assim como no desenvolvimento do raciocínio lógico-matemático (SILVA, JR; HORNINK, 2017).

Utilizou-se o Scratch no segundo tópico da primeira oferta do curso online e também durante toda a segunda oferta, na qual se trabalhou apenas este, para

aprofundar os conceitos de programação, sendo que os estudantes puderam construir seus próprios objetos, de acordo com as áreas de interesse do mesmo.

Uma das experiências estudadas apresenta-se em Finizola (2014), que em seu trabalho produziu um curso de ensino de programação para 20 alunos do 1º ano do ensino médio de uma escola pública.

A ferramenta escolhida foi a plataforma App Inventor, alegando-se que a mesma apresenta conceitos de programação de modo atraente e significativo para os alunos, além de abstrair a complexidade da programação tradicional.

Ainda segundo Finizola (2014, p. 340), "A turma mostrou-se motivada e ansiosa em ver o aplicativo desenvolvido executando em seus próprios tablets". Além disso, a plataforma App Inventor foi apropriada para atingir o objetivo proposto pelo curso, possibilitando um feedback rápido da construção dos algoritmos.

Em uma experiência, Wangenheim (2014) desenvolveu atividade usando o Scratch, aplicada em uma turma de 24 alunos do 1º ano do ensino fundamental de escola privada, sendo que, de acordo com este (2014, p. 117), "as crianças desenvolveram uma versão interativa da história infantil Chapeuzinho

Vermelho, de forma integrada ao conteúdo das disciplinas de Língua Portuguesa e Artes”.

Concluiu-se com os resultados da pesquisa (VON WANGENHEIM, 2014) que o uso do Scratch possibilitou a aprendizagem dos conceitos básicos de computação de forma efetiva, divertida e lúdica, despertando o interesse e motivação dos alunos pela tecnologia digital.

Dessa forma, destacando-se o quanto esse tipo de atividade contribui para motivação escolar, aprendizagem de conceitos escolares específicos, do estímulo ao raciocínio lógico, e outras habilidades cognitivas.

3.3 Ambiente virtual de aprendizagem - Moodle

Os ambientes virtuais de aprendizagem tem chamado atenção devido a grande importância que exercem na área da educação, principalmente quanto ao seu suporte à aprendizagem colaborativa. Esses ambientes focam no trabalho em grupo, com vistas a promover a aprendizagem coletiva, através da colaboração de cada indivíduo participante (BITTENCOURT, 2007).

O aspecto social desses ambientes possibilita a conectividade como forma de compartilhar ideias, criar

conteúdos dinâmicos e proporcionar o aprendizado coletivo (SERRÃO, 2016).

Ainda, segundo Serrão (2016, p. 23):

“o aprendizado pode ser visto como um processo de formação de redes que é o ato de representar, conectar e organizar nodos especializados para facilitar a informação e o fluxo de conhecimento”.

O Moodle foi desenvolvido para possibilitar a criação de cursos a distância publicados na web em 2001 (versão 1.0), possibilitando interação com os envolvidos de forma mais prática e fácil (TEODORO; ROCHA, 2007). É distribuído como open source, licenciado sob a *General Public License* (GNU GPL).

O sistema é utilizado por meio de um navegador web ou aplicativo e consiste em um conjunto de unidades funcionais denominadas módulos.

Cada módulo então é responsável pela execução de determinadas funções, tais como criar cursos online e fóruns para discussões, além de permitir a criação e execução de testes.

O ambiente Moodle é uma plataforma essencial para o desenvolvimento deste projeto pois, além de possibilitar a criação dos módulos do curso de forma interativa e eficaz, dispõe de diversas ferramentas que foram utilizadas para coletar dados importantes dos

participantes como o número de atividades realizadas, as temáticas que tiveram maior foco e a frequência de participação, dentre outros.

3.4 Organização da atividade de formação online

A atividade de formação *online*, que chamado simplificada de curso, foi desenvolvida e organizada de acordo com os principais temas e conceitos da programação computacional como operadores, variáveis, condicionais e repetição.

Destaca-se que é uma das ações do projeto de extensão da Unifal-MG “Pensando em códigos”, no qual se trabalharam os aplicativos Code.org, Scratch e App Inventor, virtualmente e presencialmente.

Fez-se a introdução aos conceitos computacionais utilizando atividades no Code.org, como exemplificada na Figura 5.4, nesse caso, com a temática do Angry Birds, tornando esta mais lúdica e atrativa aos participantes, principalmente na faixa de 8 a 16 anos.

Pode-se arrastar e juntar os blocos para criar uma sequência de ações para o personagem e, posteriormente, clicar em “executar” para verificar o resultado. A medida que se tem sucesso em realizar a ação, pode-se avançar para desafios mais complexos.

The image displays a game environment on a 10x10 grid. The grid contains various elements: wooden blocks, grey blocks, TNT blocks, and a red bird (Agent) and a green pig (Goal). Below the grid are two orange buttons: 'Executar' (Execute) and 'Avançar' (Advance). The bottom part of the image shows a code editor interface with a task: 'Leve o Agent para a placa de pressão.' (Take the Agent to the pressure plate.) The code editor has two panes: 'Blocos' (Blocks) and 'Área de trabalho:' (Workspace). The workspace contains a script starting with 'quando executar' (when executed), followed by 'vire à esquerda' (turn left), 'vire à direita' (turn right), 'repita 7 vezes' (repeat 7 times), and 'faça avance' (do advance).

Figura 5.4 - Exemplo de atividade introdutória utilizando a plataforma Code.org

Fonte: Telas do code.org capturada pelos autores.

A partir da Figura 5.4, observa-se que esta tarefa é relativamente simples, indutiva e intuitiva, semelhante à programação que se utiliza no Scratch.

Em cada tarefa no Code.org há um objetivo lúdico específico, por exemplo, o pássaro vermelho deve encontrar o porco verde.

Em cada módulo do curso, projetou-se uma sequência didática para ser seguida de acordo com o tema a ser trabalhado na presente semana.

Desta forma, cada módulo é relacionado com seu respectivo tema e apresentava instruções e atividades semelhantes aos módulos anteriores, fazendo com que o estudante criasse uma maior familiaridade com a estrutura do curso e assim tivesse maior facilidade para entender o que deveria ser feito em cada semana.

Na tela de trabalho do Code.org (Figura 5.4), encontra-se o botão "</> Mostrar código", a partir do qual pode-se visualizar como seria o código, em seu princípio:

```
turnRight();  
moveForward();  
turnLeft();  
moveForward();  
moveForward();  
moveForward();  
turnLeft();  
moveForward();
```

Dessa forma, o estudante pode, para cada execução, verificar a sequência dos comandos que resultou a ação indicada pelos blocos.

Apresenta-se um exemplo dessa sequência, com o conceito de repetição, no Quadro 5.2.

Quadro 5.2 - Exemplo de sequência didática a ser seguida em determinado módulo.

TEMA: Repetição SUBTEMA: Variantes das estruturas de repetição
ATIVIDADES: Realizar a leitura do material auxiliar introdutório ao conceito. Assistir as vídeo-aulas de cada subtema. Responder ao Quiz sobre a dificuldade encontrada em cada subtema. Responder aos questionários de avaliação sobre o conteúdo dos subtemas. Participar obrigatoriamente do fórum de discussão sobre o tema.

Fonte: Os autores

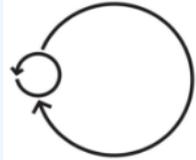
Exemplifica-se na Figura 5.5 um recorte de uma das telas do curso online no Moodle (2ª semana). Cada semana fica visível somente nas datas estipuladas no Moodle, assim como algumas atividades eram obrigatórias para liberação das atividades seguintes, criando um caminho para o participante que aumentava a dificuldade gradativamente.

Os participantes que dominava, alguns conceitos previamente teriam condições de avançar por essas rapidamente até chegar às que não conhecia.

Boas vindas | 1ª Semana | **2ª Semana** | 3ª Semana | 4ª Semana | 5ª Semana | Projeto Final

2ª Semana

11 até 25 de Abril - Repetição



Neste módulo trabalharemos a estrutura de repetição simples e composta no scratch, pratiquem bastante e utilizem a criatividade.

Vamos juntos?

-  Leia e siga as orientações.
-  AULA 04 - Repetição no Scratch
-  4.1 - Leitura - Estrutura de Repetição no Scratch
-  4.2 QUIZ- Após utilizar a estrutura de repetição no scratch, responda
-  4.3 Avaliação - Questionário sobre o vídeo 04

Restringido Disponível se: A atividade **AULA 04 - Repetição no Scratch** esteja marcada como concluída

-  AULA 05 - Criando animações com novos atores

Figura 5.5 - Recorte de uma da tela da 2º semana, que tratou do conceito de Repetição

Fonte: Tela capturada do curso pelos autores.

Seguindo a sequência didática estabelecida, após cada videoaula (Figura 5.6), o participante realizava uma atividade avaliativa sobre o conteúdo estudado anteriormente, com o intuito de fixar o tema visto.

Esse tipo de atividade foi configurado no ambiente Moodle para que ficasse disponível apenas quando a restrição fosse cumprida, nesse caso, a visualização prévia da videoaula.

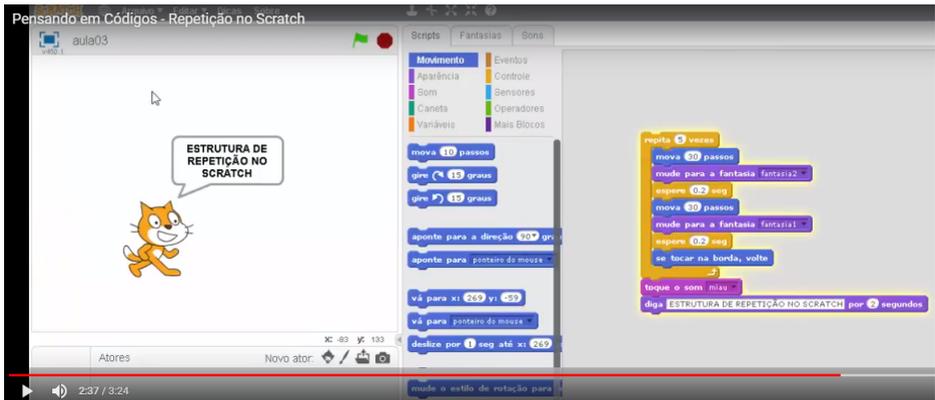


Figura 5.6 - Recorte de uma videoaula sobre Repetição.

Fonte: Tela capturada do curso pelos autores.

As principais atividades do curso online se desenvolveram, resumidamente, da seguinte forma:

- Orientações: Descreve instruções que o participante deve seguir para concluir as atividades do presente módulo;
- Videoaula: Explica o conceito computacional a ser estudado no presente módulo utilizando a plataforma e exemplos práticos;
- Leitura: Explicações teóricas acerca do conceito computacional a ser estudado no presente módulo.
- Quiz: Perguntas acerca da dificuldade encontrada em trabalhar o tema estudado no presente módulo;
- Questionário: Atividades extras propostas de acordo com o tema estudado, fazendo o

participante criar algo diferente ou dar continuidade ao que foi visto na videoaula e, depois, responder o questionário acerca do que foi feito;

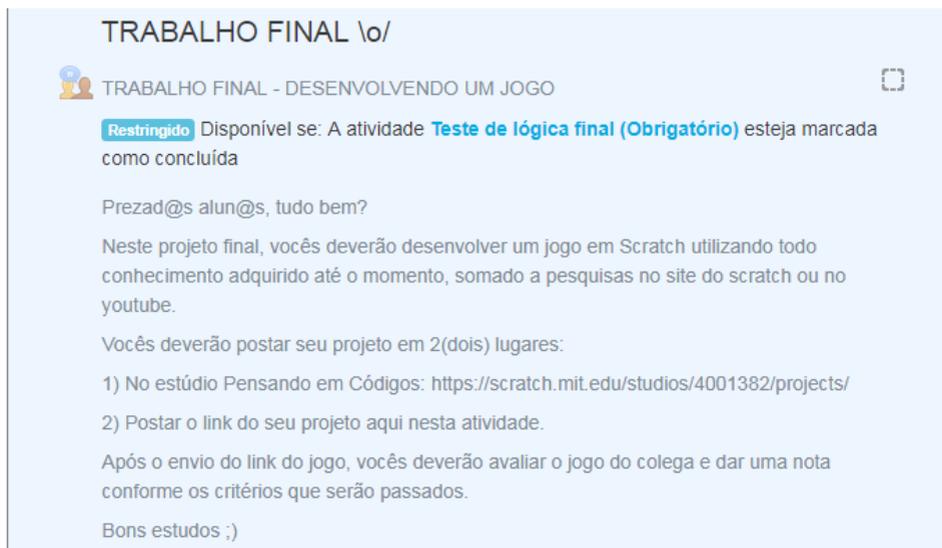
- Fóruns de dúvidas: em cada tópico havia um fórum onde os participantes compartilham suas dúvidas, sendo que tanto participantes quanto os formadores poderiam responder;
- Atividade prática: Indicavam-se pequenas metas de desenvolvimento no Scratch, relacionadas aos conceitos trabalhados, para que os participantes construísse e compartilhasse no fórum.

Após uma breve introdução aos conceitos computacionais utilizando Code.org, na sequência da primeira oferta do curso, utilizou-se o Scratch dando mais liberdade a criatividade do estudante nas criações que as atividades exigiam.

Ao fim do curso, propôs-se um trabalho (Figura 5.7), no qual o participante deveria desenvolver um jogo simples, utilizando todos os conceitos estudados no decorrer do curso, postando este no “Laboratório de avaliação” (ferramenta do Moodle).

Possibilita-se por meio desta ferramenta possibilita

que o estudante poste seu trabalho e depois avalie o trabalho de outros alunos como parte de sua avaliação, criando um sistema colaborativo de avaliação.



The screenshot shows a Canvas LMS activity page. At the top, the title is "TRABALHO FINAL \o/". Below the title, there is a sub-header "TRABALHO FINAL - DESENVOLVENDO UM JOGO" with a lock icon on the right. A blue box labeled "Restringido" (Restricted) indicates that the activity is only available if the "Teste de lógica final (Obrigatório)" activity is marked as completed. The main text of the activity is as follows:

Prezad@s alun@s, tudo bem?

Neste projeto final, vocês deverão desenvolver um jogo em Scratch utilizando todo conhecimento adquirido até o momento, somado a pesquisas no site do scratch ou no youtube.

Vocês deverão postar seu projeto em 2(dois) lugares:

- 1) No estúdio Pensando em Códigos: <https://scratch.mit.edu/studios/4001382/projects/>
- 2) Postar o link do seu projeto aqui nesta atividade.

Após o envio do link do jogo, vocês deverão avaliar o jogo do colega e dar uma nota conforme os critérios que serão passados.

Bons estudos ;)

Figura 5.7 - Recorte da tela do projeto final - criação do jogo.

Fonte: Tela capturada do curso pelos autores.

Na segunda oferta, mantiveram-se as características gerais da primeira oferta, entretanto, retirando-se a parte introdutória do Code.org e aperfeiçoando algumas atividades e tarefas.

Dessa forma, dando um foco maior no Scratch, o curso apresentou um cenário mais atrativo e objetivo, o que facilitou uma maior participação dos estudantes em tarefas mais desafiadoras e abertas a criatividade dos mesmos.

3.5 Obtenção, sistematização e análise dos dados

Para análise e compreensão das ofertas dos cursos, triangularam-se alguns dados, incluindo análise estatística descritiva dos dados quantitativos e correlação dos dados dos questionários com os resultados dos projetos.

Buscou-se na análise estatística identificar alguma melhoria nos dados pós curso em relação ao conhecimento de programação e Scratch.

Os dados foram obtidos prioritariamente utilizando as funcionalidades disponíveis no ambiente Moodle, incluindo:

- Registro de Log do Moodle (Nº de atividades concluídas e visualizações);
- Registro dos Fóruns (Participação em dúvidas, respostas e discussões).

Além disso, elaboraram-se questionários de avaliação pré e pós curso para identificar os conhecimentos prévios e posteriores ao curso em relação programação e aos aplicativos utilizados. Ressalta-se que se coletaram os resultados apenas daqueles que concluíram no mínimo 75% das atividades do curso, incluindo-se:

- Questionário pré-curso: Conhecer o nível de conhecimento em programação e em Scratch, através de avaliação dos próprios participantes, numa escala de 1 a 5;
- Questionário pós-curso: Avaliação da melhora do conhecimento em programação e em Scratch, segundo os próprios participantes, numa escala de 1 a 5.

4 Resultados e Discussões

Por meio das funcionalidades que a ferramenta Moodle oferece, registraram-se diversos dados que permitiram avaliar e discutir a participação e a evolução dos estudantes que se envolveram com o curso. Por meio desses dados, buscou-se entender os pontos positivos e negativos presentes na construção do curso, na interação com os estudantes, assim como no formato utilizado para a realização das atividades.

4.1 Medida de participação geral

A primeira oferta do curso “Pensando em códigos: Scratch básico” ocorreu com 87 atividades sobre o Scratch e Code.org, incluindo fóruns, quizzes, vídeo-aulas, materiais didáticos, e questionários avaliativos.

Dividiram-se os participantes em grupos, com o intuito de facilitar a atuação dos formadores e propiciar melhores diálogos nos fóruns.

Destaca-se que os grupos não foram aleatório, mesclando-se participantes com experiência prévia em programação e ou Scratch e de idades distintas, de tal forma que se potencializasse a colaboração entre os próprios participantes (os participantes menores de 18 anos ficaram em grupo separado dos demais).

Analisando-se os registros de log no Moodle, identificando-se, para cada grupo de participantes, o percentual de atividades concluídas, possibilitando-se agrupar os mesmos em quatro faixas de percentual de realização de atividades (Tabela 5.1).

Tabela 5.1 - Número de concluintes por etapa na 1º oferta da unidade instrucional.

Grupo	Menos de 25%	Entre 25 e 50%	Entre 50 e 75%	Mais de 75%	Total
Alpha	11	3	0	0	14
Beta	23	2	1	8	34
Gama	21	3	3	5	32
Total	55	8	4	13	80

Fonte: Os autores

A partir desses dados, percebe-se que de um total de 80 participantes, apenas 13 (16,25%) chegaram a concluir mais de 75% do curso, valor considerado

adequado para abranger todos os temas propostos.

Esta quantidade de concluintes está dentro do esperado, tendo em vista que a evasão em MOOCs (*Massive Open Online Course*²⁴) é de 75% à 95% dos inscritos (POY; GONZALES-AGUILAR, 2014).

Para segunda oferta, construíram-se 70 atividades abordando somente o Scratch, sendo que esta se iniciou com 319 inscritos, divididos em 11 grupos.

Apresenta-se na Tabela 5.2 a distribuição das frequências de conclusão do curso por grupo, sendo que 55 pessoas concluíram mais que 75%, o que corresponde a 17.25%.

Esse percentual está dentro do esperado em cursos MOOCs, tendo em vista a taxa média de conclusão e evasão de MOOCs. A questão da evasão se agrava, principalmente, em MOOCs abertos/ gratuitos.

Muitos participantes concluíram menos de 25%, sendo que a grande maioria destes não iniciou o curso, fato que ocorre por vários motivos, entre eles: falta de tempo; subestimação do tempo demandado; falta de interesse; falta de motivação para iniciar o curso.

Tabela 5.2 - Número de concluintes por etapa na 2º fase do curso. 24Cursos online Abertos Massivos (tradução nossa).

Grupo	Menos de 25%	Entre 25 e 50%	Entre 50 e 75%	Mais de 75%	Total
Alpha	18	6	1	4	29
Beta	16	3	3	7	29
Capa	19	4	2	4	29
Delta	14	4	5	6	29
Epsilon	18	4	1	6	29
Eta	20	3	1	5	29
Gama	18	1	5	5	29
Iota	17	4	3	5	29
Lambda	22	0	4	3	29
Teta	20	3	2	4	29
Zeta	18	3	2	6	29
TOTAL	200	35	29	55	319

Fonte: Os autores

4.2 Medida de participação nos fóruns de dúvidas

Por meio do número de interações nos fóruns de dúvidas de cada módulo, é possível identificar aqueles que tiveram maior participação. Dessa forma, pode-se dizer que os temas mais discutidos foram aqueles que causaram maior dúvida ou despertaram maior interesse dos participantes.

Observa-se na Tabela 5.3 que os fóruns que apresentaram maior número de interações na primeira oferta da unidade instrucional foram o de "Dúvidas - Módulo 1", que abordou as dúvidas iniciais, como cadastro na plataforma Code.org e atividades prévias, e

o de “Dúvidas - Módulo 5”, no qual se discutiu o conceito de repetição no Scratch.

Tabela 5.3 - Nº de interações por fórum 1º oferta.

Tópico	Respostas
Dúvidas - Módulo 1	7
Dúvidas - Módulo 2	3
Dúvidas - Módulo 3	3
Dúvidas - Módulo 4	1
Dúvidas - Módulo 5	6
Dúvidas - Módulo 6	0
Dúvidas - Módulo 7	2

Fonte: Os autores

Na segunda oferta, estas informações podem ser consideradas ainda mais consistentes, visto que a amostra abrange maior número de participantes. Observa-se na Tabela 5.4 que os fóruns que geraram maior discussão foram os de “Dúvidas - Módulo 4”, que abordou dúvidas sobre criação de listas e do jogo exemplo e o de “Dúvidas - Trabalho Final”, no qual os participantes desenvolveram um jogo próprio utilizando os conceitos estudados. Destaca-se que a participação nos fóruns não era obrigatória, assim, não sendo contabilizados para nota ou frequência.

Tabela 5.4 - N° de interações por fórum 2° oferta.

Tópico	Respostas
Dúvidas - Módulo 1	31
Dúvidas - Módulo 2	36
Dúvidas - Módulo 3	26
Dúvidas - Módulo 4	38
Dúvidas - Módulo 5	19
Dúvidas - Trabalho final	46

Fonte: Os autores

Destaca-se que em edições posteriores deste curso (2017 - 3ª oferta), alterou-se a forma de postagem dos trabalhos semanais para postagens nos fóruns, com obrigatoriedade de cada participante avaliar o trabalho de mais dois participantes, desta forma, tem-se uma estrutura didática que intencionalmente se estimula a aprendizagem colaborativa.

4.3 Medida de aprendizado em programação e em Scratch na 1° oferta

Obtiveram-se no início de cada oferta do curso informações dos participantes referentes ao nível de conhecimento e domínio nos conceitos de programação e desenvolvimento na plataforma Scratch.

Nos questionários pré e pós curso coletou-se a opinião dos próprios participantes, que deram notas de 1

a 5 (1 - Muito baixo, até 5 - Muito alto), referente aos dois parâmetros estudados.

No apêndice 5.C, apresenta-se tal cenário para a primeira oferta do curso, com uma tabela formada pelos valores que representam os 2 parâmetros estudados, sendo analisado somente os resultados daqueles que concluíram mais de 75% das atividades, ou seja, um total de 13 participantes.

Para analisar e comprovar se houve diferença significativa entre os dados iniciais e finais, aplicaram-se testes estatísticos. Inicialmente, aplicou-se o teste Shapiro-Wilk, indicado para amostras com $2 < n < 51$, para verificar a normalidade dos dados (LEOTTI, 2005).

O teste de Shapiro-Wilk é obtido dividindo-se o quadrado da combinação linear apropriada dos valores ordenados da amostra pela variância simétrica estimada (LEOTTI, 2005).

Apresenta-se no apêndice 5.D os resultados dos testes estatísticos referentes aos dados obtidos (prévios e posteriores) na primeira oferta da unidade instrucional.

No teste de Shapiro-Wilk, com um $\alpha = 0,05$, se o valor de $p < \alpha$, então nega-se a hipótese nula que indica para a normalidade dos dados. Portanto, como os valores de p mostrados no quadro 5.5 do apêndice 5.C

são sempre menores que alfa, os dados não apresentam uma distribuição normal, desta forma, aplicaram-se testes não-paramétricos (para dados não normais).

Optou-se pelo teste de Wilcoxon, este teste é uma prova não-paramétrica destinada a comparar dados pareados de uma amostra que foram obtidos antes e depois de algum evento (NETO, 2003).

Primeiro, analisaram-se os dados relacionados ao nível de conhecimento de Scratch, avaliado pelos próprios participantes antes e após o curso.

Neste teste, a hipótese de nulidade sempre aponta que não há diferença significativa entre os valores, enquanto que a hipótese alternativa aponta para a ocorrência de diferença entre os resultados prévios e posteriores (nível de decisão: $\alpha = 0,01$).

Observa-se no quadro 5.6 do apêndice 5.D que o valor de $p=0.0015$ é altamente significativo em relação a alfa, logo rejeita-se a hipótese de nulidade e aceita-se a alternativa. Portanto, como o saldo é positivo, os valores que indicam conhecimento em Scratch são maiores após a aplicação do curso. Isso mostra que, segundo os participantes, o curso auxiliou significativamente no aumento do conhecimento da plataforma Scratch.

Da mesma forma, agora analisando em relação ao

nível de conhecimento em programação, observa-se que no quadro 5.7 do apêndice 5.D como $p=0,0033$ é altamente significativo em relação a alfa, rejeita-se a hipótese de nulidade e aceita-se a alternativa.

Desta forma, infere-se que os valores atribuídos ao conhecimento de programação após a realização do curso são maiores, indicando que houve um aumento significativo na autopercepção da aprendizagem de programação pelos participantes envolvidos na unidade instrucional.

Dados de Silva Jr. (2018) indicam que além destes dados, que as notas da avaliação pré e pós curso em testes de raciocínio lógico-matemático também indicaram uma diferença significativa.

4.4 Medida de aprendizado em programação e em Scratch na 2ª oferta

Na segunda oferta do curso, fez-se as mesmas análises realizadas com os dados da primeira com os 55 estudantes que concluíram pelo menos 75% das atividades. De forma semelhante, aplicaram-se os questionários prévios e posteriores que identificaram a opinião dos participantes quanto ao seu nível de conhecimento em programação e em Scratch (Apêndice 5.E).

Desta forma, aplicou-se um teste estatístico para analisar os resultados obtidos. Para determinar qual teste de verificação de diferença deveria ser utilizado, utilizou-se o teste de Lilliefors com o intuito de identificar se os dados apresentam ou não uma distribuição normal. O teste de Lilliefors é uma adaptação do teste de Kolmogorov-Smirnov, calculando a média e o desvio no próprio conjunto de dados, sem necessidade do cálculo prévio (TORNAN, 2012).

Neste caso, não foi usado o teste de Shapiro-Wilk pois a amostra é referente a um $n = 55$, valor maior do que o indicado por esse teste.

Logo, preparadas as amostras a serem estudadas, definiram-se as hipóteses para o teste de normalidade de Lilliefors, com um nível de decisão $\alpha = 0,05$.

No quadro 5.8 do apêndice 5.F observa-se que em todas as colunas o valor de p corresponde a $p < 0.01$, pelo teste de Lilliefors nega-se a hipótese de nulidade. Portanto, os dados apresentam uma distribuição anormal, dessa forma, aplicaram-se testes estatísticos não-paramétricos (para dados não normais).

A partir desse resultado, optou-se pelo teste de Wilcoxon para analisar o significado dos dados da 2ª oferta. Inicialmente, aplicou-se o teste para as duas

primeiras colunas do apêndice 5.E, que representam a indicação do grau de conhecimento da plataforma Scratch.

Assim como apresentado no teste anterior, a hipótese de nulidade sempre aponta que não há diferença significativa entre valores e a hipótese alternativa indica diferença significativa entre os valores estudados (nível de decisão $\alpha = 0,01$).

A partir dos resultados indicados no quadro 5.9 do apêndice 5.F, como o valor de $p < 0.0001$ é bastante significativo em relação ao nível de decisão α , rejeita-se a hipótese de nulidade. Logo, como os valores que indicam o grau de conhecimento em Scratch pós curso são maiores que os valores indicados no pré curso, houve uma melhora significativa na autopercepção sobre os conhecimentos dos participantes em relação a plataforma Scratch.

Em mais uma aplicação do teste, agora tendo vista a evolução no quesito conhecimento de programação, observa-se no quadro 5.10 do apêndice 5.F que o valor de $p < 0.0001$ indica que deve-se rejeitar a hipótese de nulidade.

Portanto, os valores que representam o nível de conhecimento em programação após a realização do

curso são maiores que os valores indicados no pré curso, mostrando que houve uma melhoria significativa na auto percepção sobre o entendimento dos participantes no conceito de programação.

Da mesma forma que na primeira oferta, Silva Jr. (2018) testou se havia diferença significativa entre as notas da avaliação pré e pós curso em testes de raciocínio lógico-matemático e, da mesma forma, também encontrou diferença significativa, reforçando que, a partir do curso, os participantes obtiveram uma melhora significativa no raciocínio lógico-matemático, advindo do aprendizado dos princípios de programação.

4.5 Comentários dos participantes sobre o curso

No final das atividades aplicaram-se os questionários sobre conhecimento em programação, Scratch e lógica, nos quais os participantes do curso deixaram seus comentários a respeito dos pontos positivos e negativos que tiveram durante a aplicação da unidade instrucional.

Segue no Quadro 5.3 alguns desses comentários que podem contribuir para uma avaliação geral e para uma possível melhora da unidade em futuras aplicações.

Quadro 5.3 - Comentários dos participantes sobre o curso.

Tópico	Comentários
Pontos fortes da experiência	<p>“Os blocos e a quantidade de conteúdo está perfeito para quem não tem muito tempo. O resumo do conteúdo está condizente com um aluno que não conhece programação e a didática de vocês, as videoaulas deram conta do recado.”</p> <p>“Gostei muito de participar do curso, bastante interessante, tutores muito atenciosos.”</p> <p>“O conteúdo do curso foi de alto nível para estimular a motivação e auto conhecimento de soluções e atitudes em diversas situações na área da lógica de programação e para desenvolver habilidades nos problemas do cotidiano”</p> <p>“Amei o curso, acredito que eu tenha aumentado e muito meus conhecimentos sobre iniciação de programação.”</p> <p>“Ótimo curso, completíssimo, com ótimas videoaulas com explicações fáceis para quem não sabe do assunto, faria outra vez!”</p> <p>“Adorei o curso. Já tinha conhecimento e realizado formações do Scratch anteriormente, mas esse foi bem didático possibilitando quem não tinha conhecimento de acompanhá-lo.”</p> <p>“Aprendi bastante com este curso e aprimorei minha lógica.”</p> <p>“Foi muito importante esse curso pra que eu pudesse ter contato com o scratch e a partir dele começar a aplicar com meus alunos.”</p> <p>“Gostaria de agradecer imensamente pela oportunidade. O curso é bárbaro! Simples e objetivo.”</p> <p>“Esse curso me ajudou muito a realizar um sonho de</p>

	criar jogos eu agradeço a todos os professores deste curso.”
Pontos fracos da experiência	<p>“Poderia ter mais tempo para as postagens”</p> <p>“Poderia ter um tempo maior para desenvolver o projeto”</p> <p>“Atores e cenários limitados. Desenho ou modificação dos atores bastante complexa”</p> <p>“Adorei, só o tempo para realização das atividades que foram muito apertados.”</p> <p>“Senti falta da participação do restante da turma. Gostaria de ter tido mais troca de experiências”</p> <p>“Acho que dentro das atividades propostas poderiam propor a leitura de materiais mais teóricos sobre os conceitos de pensamento computacional, algoritmos e outros, para complementar a parte prática do curso que já é muito boa.”</p>

Fonte: Os autores

Em geral, percebe-se pelos comentários que o que mais atraiu e gerou uma avaliação positiva dos participantes foi a estrutura utilizada na construção da unidade educacional online. Os tópicos seguiram uma sequência didática organizada onde os estudantes tiveram facilidade em entender e saber o que deveria ser feito.

Por outro lado, o que chamou mais atenção nas avaliações negativas foi que muitos reclamaram da quantidade de tempo dada para a realização das

atividades. Esse ponto pode ser levado em consideração em futuras aplicações da unidade educacional online tendo em vista que alguns estudantes podem demorar mais tempo para dominar alguns conceitos computacionais.

5 Considerações finais

Diante dos novos contextos da relação humano-computador, em que a tecnologia está sendo cada vez mais inserida no cotidiano das pessoas, as discussões e avaliações sobre o envolvimento das tecnologias digitais nas atividades da sociedade são extremamente importantes para que não entremos cegos em um mundo ainda pouco conhecido.

Por meio deste trabalho pode-se estudar e conhecer, diante de pesquisas e aplicações, vários cenários que ao serem debatidos levantam novas reflexões e possibilidades. A partir do desenvolvimento da unidade instrucional online, percebe-se que as plataformas Scratch e Code.org são ótimas alternativas de ferramentas para ensino de programação à estudantes que nunca tiveram contato com o tema.

Estas ferramentas se mostraram potenciais desenvolvedoras das capacidades lúdicas dos

estudantes, além de ampliar a autoria e a criatividade dos participantes na construção de seus próprios materiais. Isso mostra que é possível inserir tecnologias digitais nos novos processos de ensino-aprendizagem, uma vez que a multidisciplinaridade é favorecida no uso desses ambientes.

Por fim, pode-se inferir que a partir dos resultados obtidos que as atividades desenvolvidas pelos estudantes auxiliaram significativamente na obtenção de conhecimento do conceito de programação. Além disso, com os resultados e comentários dos participantes da unidade instrucional concluímos que as ferramentas utilizadas no projeto contribuem, auxiliam e facilitam os processos de ensino-aprendizagem, o que mostra a importância do ensino de programação e softwares de autoria para todos.

Com o sucesso obtido no uso das ferramentas utilizadas no trabalho quanto ao aumento no aprendizado da lógica de programação, um possível trabalho futuro é a construção de uma unidade instrucional para o acompanhamento de estudantes de Tecnologias da informação (TI) nas fases iniciais de seus cursos. As disciplinas de início de programação, como Algoritmos, possuem uma grande taxa de reprovação.

Logo, é possível trabalhar softwares lúdicos nessas fases visando a construção de uma base mais sólida no aprendizado desses estudantes.

Agradecimentos

Ao CNPq pela bolsa de Iniciação Científica por meio do programa de Iniciação Científica (PIBIC/UNIFAL), à Proex/ Unifal-MG, pela bolsa de Extensão.

Referências

ALMEIDA, M. E. B. *Informática e formação de professores*. Brasília: Ministério da Educação/Proinfo, 2001.

ALVES, N. C. et al. Ensino de Computação de Forma Multidisciplinar em Disciplinas de História no Ensino Fundamental – Um Estudo de Caso. *Revista Brasileira de Informática na Educação*, [S.l.], v. 24, n. 3, 2016, p. 31-46. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/6481/4970>>. Acesso em: 30 jul. 2018.

BITTENCOURT, I. et al. Sistemas de Autoria para Construção de Ambientes Interativos de Aprendizagem Baseada em Agentes. *Revista Brasileira de Informática na Educação*, [S.l.], v. 15, n. 1, abr. 2007. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/59>>. Acesso em: 31 jul. 2018.

BRAIT, L. F. R. et al. A relação Professor/Aluno no processo de ensino e aprendizagem. *Itinerarius Reflectionis*, v. 6, n. 1, 2010.

BULGRAEN, V. C. O papel do professor e sua mediação nos processos de elaboração do conhecimento. *Revista conteúdo*,

v. 1, n. 4, p. 30-38, 2010.

CASTELLS, M. *Comunicación y poder*. Madrid: Alianza Editorial. 2009.

DIAS, L. R. Inclusão digital como fator de inclusão social. In: BONILLA, M. H. S.; PRETTO, N. D. L. (orgs). *Inclusão digital: polêmica contemporânea*. Salvador: EDUFBA, 2011, p. 61-90. Disponível em: <<https://pt.calameo.com/read/0051600936ee1e189a033>>. Acesso em 20 jun. 2018.

FINIZOLA, A. B. et al. O ensino de programação para dispositivos móveis utilizando o MIT-App Inventor com alunos do ensino médio. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 20, Dourados: Anais ... Universidade Federal da Grande Dourados, 2014, p. 337. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/3116>>. Acesso em 11 jul. 2018.

FREIRE, P. . *Pedagogia da Autonomia: saberes necessários à prática educativa*. São Paulo. Paz e Terra, 1996.

HORNINK, G. G. *Cartografando online: caminhos da informática na escola com professores que elaboram conhecimentos em formação contínua*. 2010. 296f. Tese (Doutorado em Ciências) - Instituto de Geociências, Universidade Estadual de Campinas (Unicamp), Campinas/SP, 2010.

HORNINK, G. G.; COMPIANI, M. Reflexões das mediações online no processo de formação continuada de professores com foco nos problemas ambientais. *Educação Temática Digital*, [S.I.] v. 19, n. 4, p. 773-794, 2017.

LEOTTI, V. B.; BIRCK, A. R.; RIBOLDI, J. Comparação dos Testes de Aderência à Normalidade Kolmogorov Smirnov, Anderson-Darling, Cramer-Von Mises e Shapiro-Wilk por Simulação. In: SIMPÓSIO DE ESTATÍSTICA APLICADA À EXPERIMENTAÇÃO AGRONÔMICA, 2005, Londrina. *Anais...*

Londrina: RBRAS, 2005. p. 192-196.

MASCHIO, E.; DIRENE, A. I. Múltiplas Representações Externas no Suporte à Aquisição de Conhecimento em Programação de Computadores. *Revista Brasileira de Informática na Educação, Local*, v. 23, n. 3. 2015.

MEIRELLES, F. S. 29ª Pesquisa Anual do Uso de TI. São Paulo. Fundação Getúlio Vargas, 2018. Disponível em: <https://eaesp.fgv.br/sites/eaesp.fgv.br/files/pesti2018gvciap_pt.pdf>. Acesso em 11 jun. 2018.

MERCADO, L. P. L. (Org.). *Novas tecnologias na educação: reflexões sobre a prática*. Maceió: Edufal, 2002.

MORAES, R. A. *Informática na educação*. Rio de Janeiro: DPA, 2000.

MORAN, J. M. Ensino e aprendizagem inovadores com tecnologias. *Informática na educação: teoria & prática*, Rio Grande do Sul, v. 3, n. 1. 2000.

MORAN, J. Mudando a educação com metodologias ativas. In: SOUZA, C. A.; MORALES, O. E. T. (orgs). Coleção Mídias Contemporâneas. *Convergências Midiáticas, Educação e Cidadania: aproximações jovens*. Ponta Grossa: UEPG/PROEX, 2015. v. 2, p. 15 - 33. Disponível em: <<https://uepgfocafoto.wordpress.com/2774-2/colecao-midias-contemporaneas-convergencias-midiaticas-educacao-e-cidadania-aproximacoes-jovens-vol-ii>>. Acesso em: 13 fev. 2018.

NETO, A. A. H.; STEIN, C. E. *Uma abordagem dos testes não-paramétricos com utilização do Excel*. 2003. Disponível em: <http://www.mat.ufrgs.br/~viali/estatistica/mat2282/materia/l/textos/artigo_11_09_2003.pdf>. Acesso em 10 jun. 2018.

PEREIRA, Andréia Regina; DE DEUS LOPES, Roseli. (2005). Legal: Ambiente de Autoria para Educação Infantil apoiada em Meios Eletrônicos Interativos. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 26. 2005. Juiz de Fora.

Anais ... Juiz de Fora, UFJF: 2005. p. 23-30. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/389>>. Acesso em: 11 jul. 2018.

PINTO, I. M.; BOTELHO, S. S. C. Ambientes Tecnológicos Lúdicos de Autoria (ATLA): criando espaços de ensino e aprendizagem. *Revista Brasileira de Informática na Educação*. Porto Alegre, v. 20, n. 3, 2012. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/1393>>. Acesso em: 11 jul. 2018.

POY, R.; GONZALES-AGUILAR, A. Factores de éxito de los MOOC: algunas consideraciones críticas. *Revista Ibérica de Sistemas e Tecnologia da Informação*, Portugal, v. 03, n. E1, p. 95-118, 2014

RAMOS, J. L.; ESPADEIRO, R. G. Os futuros professores e os professores do futuro. Os desafios da introdução ao pensamento computacional na escola, no currículo e na aprendizagem. *Educação, Formação & Tecnologias*, v. 7, n. 2, p. 4-25, 2014.

SCAICO, P. D. et al. Ensino de Programação no Ensino Médio: uma abordagem orientada ao design com a linguagem scratch. *Revista Brasileira de Informática na Educação*, Rio de Janeiro, v. 21, 2012

SERRÃO, T. et al. Construção automática de redes sociais móveis no ambiente moodle. *Revista Brasileira de Informática na Educação*, Porto Alegre, v. 24, n. 1, 2016. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/3151>>. Acesso em 30 jul. 2018.

SILVA JR, A. M. *Microgênese do desenvolvimento sociocultural do raciocínio lógico-matemático mediado por tecnologias educacionais*. 2018. 135f. Dissertação (Mestrado em Educação)- Universidade Federal de Alfenas, Alfenas, 2018. Disponível em: <<https://btdt.unifal-mg.edu.br:8443/handle/tede/1117>>. Acesso em 31 jul. 2018.

SILVA JR, A. M. ; HORNINK, G. G. Desenvolvendo o raciocínio lógico-matemático a partir da aprendizagem de princípios de programação em blocos. In: ENCONTRO VIRTUAL DE DOCUMENTAÇÃO EM SOFTWARE LIVRE (EVIDOSOL), 14; CONGRESSO INTERNACIONAL DE LINGUAGEM E TECNOLOGIA ONLINE, 10 (CILTEC-online), 2017, Belo Horizonte (Online). *Anais ...* Belo Horizonte: Texto Livre, 2017. p. 1-6.

SILVEIRA, S. A. Inclusão digital, software livre e globalização contra-hegemônica. *Software Livre e Inclusão Digital*-Organizadores: Sergio Amadeu de Silveira e João Cassino, v. 7, p. 11, 2003.

SILVEIRA, S. A. A noção de exclusão digital diante das exigências de uma cibercidadania. In: HETKOWSKI, T. M. (Org.). *Políticas públicas & inclusão digital*. Salvador: EDUFBA, 2008, p. 46-56.

SILVEIRA, S. A. Para além da inclusão digital: poder comunicacional e novas assimetrias. In: BONILLA, M. H. S.; PRETTO, N. D. L., orgs. *Inclusão digital: polêmica contemporânea* [online]. Salvador: EDUFBA, 2011, p. 49-59. Disponível em <<http://books.scielo.org/id/qfgmr/pdf/bonilla-9788523212063-04.pdf>>. Acesso em: 30 jul. 2018.

TEODORO, G. L. M.; ROCHA, L. C. D. *Moodle – Manual do Professor*. Belo Horizonte: UFMG, 2007. (Obtido via Internet, em dezembro de 2007, a partir do Centro de Apoio à Educação à Distância. Disponível em: <http://mambo.grude.ufmg.br/paginas/caed/index.php?option=com_remository&Itemid=33&func=fileinfo&id=6>. Acesso em: 10 jun 2018.

TORMAN, V. B. L.; COSTER, R.; RIBOLDI, J. Normalidade de variáveis: métodos de verificação e comparação de alguns testes não-paramétricos por simulação. *Clinical & Biomedical Research*, Rio Grande do Sul, v. 32, n. 2, 2012.

VALENTE, J. M. Diferentes usos do computador na Educação. *Em aberto*, Brasília, v. 12, n. 57, 1993. Disponível em <<http://emaberto.inep.gov.br/index.php/emaberto/articli>

[e/view/1876](#)>. Acesso em: 11 jun. 2018.

VON WANGENHEIM, C. G; NUNES, V. R.; DOS SANTOS, D. D. Ensino de Computação com SCRATCH no Ensino Fundamental–Um Estudo de Caso. *Revista Brasileira de Informática na Educação*, Porto Alegre, v. 22, n. 03, p. 115. 2014. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/2885>>. Acesso em 11 jun. 2018.

Apêndices

Apêndice 3.A Proposta de projeto final do curso de Scratch.

Proposta

Para o trabalho final do curso, vocês terão que desenvolver um projeto de tema livre em dupla ou individual, que obedecerá aos requisitos mínimos listados abaixo.

O projeto deverá:

- Ser interativo, ou seja, ter a interação do usuário com o computador. Para isso, será necessário utilizar, pelo menos, dois tipos de Sensores;
- Utilizar os seguintes comandos de controle: "sempre", "se e senão" e "espere até";
- Utilizar, no mínimo, quatro sprites, com troca de trajes;
- Ter sons ativados pelas interações (ex: quando o gatinho encostar em algum objeto, emitir o som de batida);
- Utilizar, ao menos, um Operador e duas Variáveis;
- Ter caráter didático;

Entrega

A entrega deverá ser feita até o dia 2 de outubro ao meio dia, através do próprio site do Scratch e pelo Moodle. Para isso, o usuário deverá criar um cadastro, caso já não tenha, no menu Inscreva-se no site do Scratch. Após o cadastro, você poderá disponibilizar seu projeto tanto pelo editor *offline* quanto pelo editor *online*.

Para exportar no editor *offline*, abra o projeto e vá em Arquivo > Compartilhar no site, digite as informações necessárias e aperte Ok. Para abrir seu projeto no editor online, vá em Arquivo > Carregar a partir do seu computador. Depois de aberto, clique em Compartilhar. No Moodle deverá ser entregue um arquivo de texto contendo o nome da dupla, o nome do projeto e o link para acessar o projeto na página do Scratch.

Sejam criativos :)

Apêndice 3.B Questionário Inicial (QI) do curso.

- 1) Desde quando você utiliza o computador?
 - a. Desde a infância
 - b. Desde a adolescência
 - c. Desde a faculdade
 - d. Após a faculdade
- 2) Você já utilizou o Scratch?
 - a. Uso frequentemente
 - b. Usei algumas vezes
 - c. Usei uma vez ou outra
 - d. Nunca usei, mas já conhecia
 - e. Nunca usei e não conhecia até a divulgação desse curso
- 3) Você já programa em alguma linguagem?
 - a. Programo frequentemente
 - b. Programei algumas vezes
 - c. Tentei programar uma vez ou outra
 - d. Nunca programei, mas sei o que é programar
 - e. Nunca programei e nem sei o que é isso
- 4) Você está familiarizado com quais conceitos de programação?
(É possível marcar mais de uma alternativa)
 - a. Tipos primitivos
 - b. Variáveis
 - c. Constantes
 - d. Laços de repetição
 - e. Desvios condicionais
 - f. Funções e procedimentos
 - g. Não estou familiarizado com nenhum conceito de programação
- 5) Qual a sua motivação?
- 6) O que espera aprender com o curso?
- 7) Com base na figura abaixo, responda:
Se o conjunto de instruções do Pseudocódigo 1 for aplicado na Figura 1, qual será a posição final do círculo?
 - a. Linha: d, coluna: 6
 - b. Linha: d, coluna: 2
 - c. Linha: b, coluna: 5
 - d. Linha: b, coluna: 6
 - e. Linha: a, coluna: 6

Figura 1

	1	2	3	4	5	6
a						
b						
c						
d		●				
e						

Pseudocódigo 1

```
circulo.movaParaDireita(4);  
circulo.movaParaCima(3);
```

Figura 3.27 - Exercício 7 do QI.

Fonte: Os autores

8) Com base na figura abaixo, responda:

Figura 2

	1	2	3	4	5	6
a						
b						
c						
d			⇒			
e						

Pseudocódigo 2

```
repita(7)  
  seta.mova(1);  
  se(está na borda)  
    gire.horario(90°);  
fimRepita;
```

Figura 3.28 - Exercício 8 do QI

Fonte: Os autores

Se o conjunto de instruções do Pseudocódigo 2 for aplicado na Figura 2, qual será a posição final da seta?

- a. Linha: d, coluna: 3
- b. Linha: c, coluna: 3
- c. Linha: e, coluna: 3
- d. Linha: e, coluna: 4
- e. Linha: d, coluna: 2

Apêndice 3.C Questionário final (QF) do curso.

- 1) Em relação ao minicurso oferecido, como você se sente:
 - a. Alcançou além das minhas expectativas
 - b. Alcançou minhas expectativas
 - c. Alcançou em partes as minhas expectativas
 - d. Alcançou pouco as minhas expectativas
 - e. Não alcançou em nada as minhas expectativas
- 2) Como você classificaria as atividades realizadas em sala? (É possível marcar mais de uma alternativa)
 - a. De fácil entendimento
 - b. Pedagógicas
 - c. Interessantes
 - d. Conteúdo eficiente
 - e. Complicadas de entender
 - f. Não pedagógicas
 - g. Desinteressantes
 - h. Conteúdo ineficiente
 - i. Um pouco complicadas
 - j. Pouco pedagógicas
 - k. Pouco interessantes
- 3) Quais aspectos do minicurso poderia ser melhorado, a sua opinião?
- 4) Quais aspectos positivos você manteria?
- 5) Agora, ao final do minicurso, você poderia dizer que:
(É possível marcar mais de uma alternativa)
 - a. Conhece e sabe utilizar conceitos envolvendo *sprites* e palco (ex: editar, importar, etc)
 - b. Compreende e sabe utilizar as diferentes categorias de blocos (ex: controle, som, movimento, etc)
 - c. Compreende qual a utilidade da aba Trajes
 - d. Consegue fazer a interação entre usuário e computador através do bloco "sensores"
 - e. É capaz de utilizar efeitos visuais e sonoros
 - f. Entende conceitos básicos de programação como: laços de repetição (ex: repita, sempre, se e senão, etc)

- g. Entende o conceito de variável e sabe utilizá-la
 - h. Entende conceitos básicos de programação como os operadores lógicos (e, ou, maior, etc)
 - i. É capaz de construir pequenos projetos
- 6) Antes de realizar o minicurso, você tinha conhecimentos prévios sobre lógica de programação?
- a. Sim
 - b. Um pouco
 - c. Nenhum
- 7) Você realmente acredita que utilizar o Scratch pode proporcionar um maior entendimento sobre conceitos de lógica de programação?
- a. Sim
 - b. Não
- 8) Durante o pouco tempo que você utilizou o Scratch, foi possível sentir um maior entendimento sobre os conceitos envolvendo lógica de programação?
- a. Sim, além de clarear algumas dúvidas que tinha sobre esse assunto
 - b. Sim, melhorou minha visão como programador
 - c. Foi indiferente
 - d. Pelo pouco tempo que utilizei, não senti diferença, mas, talvez, se eu utilizar mais vezes, sinta alguma melhora.
- 9) A ferramenta de programação que o Scratch oferece é, realmente:
- a. Intuitiva
 - b. De fácil acesso
 - c. Simples
 - d. Lúdica (se aprende brincando)
 - e. Nenhuma das alternativas
- 10) Com que frequência você pretende utilizar o Scratch futuramente?
- a. Pretendo incluir o Scratch na minha rotina e utilizá-lo sempre
 - b. Pretendo utilizar às vezes
 - c. Não pretendo utilizar o Scratch futuramente
- 11) (Somente para estudantes de ensino médio)
Você acredita que esse minicurso ajudará em uma

decisão futura, como por exemplo, auxiliar na hora de escolher um curso universitário?

- a. Sim
- b. Não
- c. Não tenho interesse em áreas relacionadas à computação

12) (Somente para estudantes da graduação ou superior e professores) Você acredita que o Scratch tem potencial para ser utilizado em sala de aula?

- a. Sim
- b. Não

13) (Somente para estudantes da graduação ou superior e professores) Você acredita que é possível fazer aplicativos educativos usando o Scratch?

- a. Sim, mas é muito limitado
- b. Sim, tem o necessário
- c. Não

Apêndice 5. A Dados pré e pós curso - 1ª oferta.

Quadro 5.4 - Mapeamento prévio dos softwares de autoria usando linguagens de programação.

Nome	Acesso (pago ou livre)	Público-alvo	Tipo de mídia	Execução do produto
Scratch MIT	Livre	Acima de 4 anos	animação/ap p	web/desktop
Code.org	Livre	Acima de 4 anos	animação	web
Ardora	Livre	Adultos/Jovens	recursos web	desktop
CourseLab	Livre	Professores	curso e- learning	desktop
MyUdutu	Livre	Professores	curso e- learning	web
Xerte	Livre	Professores	curso e- learning	web/desktop
HagáQuê (HQ)	Livre	Acima de 4 anos	animação	desktop
KompoZer	Livre	Adultos/Jovens	páginas web	desktop
Mblock	Livre	Adultos/Jovens	Prog Arduino	desktop
Scratch4A	Livre	Adultos/Jovens	Prog Arduino	desktop
AppInventor	Livre	Adultos/Jovens	Apps celular	web
Kturtle	Livre	Acima de 10 anos	animação/ap p	desktop
Hot Potatoes	Livre	Professores	mat. e- learning	web
Yenka	Livre	Professores	mat. e- learning	desktop
RoboMind	Livre	Acima de 9 anos	Prog Robô Legó	Robô Legó
GameMaker	Livre	Adultos/Jovens	Jogo Arcade	desktop/celular
GameSalad	Livre	Adultos/Jovens	Jogo Arcade	web/celular

Macromedia Flash MX	Livre	Adultos/Jovens	Anim/app/jogo	web
GreenFoot	Livre	Acima de 14 anos	Anim/app/jogo	desktop
BlueJ	Livre	Acima de 14 anos	aplicação	desktop
VisualClass	Pago	Professores	mat. e-learning	desktop
Everest	Pago	Acima de 9 anos	animação/mat.	desktop
Imagine	Pago	Acima de 9 anos	animação/mat.	desktop
CodeCombat	Livre	Acima de 4 anos	animação	web
LightBot	Pago	Acima de 4 anos	animação	web/desk/cel
Alice	Livre	Jovens	Anim/app/jogo	desktop
Kid Pix	Pago	Acima de 4 anos	Animações	desktop
Asymetrix Toolbook	Livre	Professores	App/e-learning	web/desktop
Blocky Games	Livre	Acima de 7 anos	Animação	web
OzoBlockly	Livre	Acima de 10 anos	Animação	web
Micro:bit	Livre	Acima de 7 anos	Animação/app	web/desktop
BlocklyProp	Livre	Jovens	Anim/Prog Robo	web/robo
Wonder	Pago	Acima de 7 anos	Prog Robo/apps	web/robo

Fonte: os autores.

Apêndice 5.B Levantamento prévio.

Quadro 5.5 - Síntese da busca sobre o uso do Scratch e code.org no Brasil a partir de levantamento de artigos.

Instituição	Aplicativo	Público	Ano	Tipo
Colégio Santo Agostinho RJ	Scratch	Fund	2008	SI
FAESA - ES	Scratch	Grad	2008	SI
Educação Matemática em Revista - RS	Scratch	Educ	2009	SI
Universidade Federal do Amazonas	Scratch	Fund	2010	SI
PUCRS	Scratch	Med	2010	SI
Instituto Federal de Santa Catarina	Scratch/ S4A	Grad	2011	SI
Universidade Federal de Itajubá	Scratch	Med	2011	SI
Universidade Federal do Rio de Janeiro	Scratch	Med	2011	SI
Universidade Federal da Paraíba	Scratch	Med	2012	SI
Universidade de Pernambuco	Scratch	Fund	2012	SI
Universidade Federal do Espírito Santo	Scratch	Educ	2012	SI
Instituto Federal de Farroupilha	Scratch	Grad	2012	SI
Faculdade Meridional IMED	Scratch	Fund	2012	SI
Faculdade Pitágoras	Scratch	Tec	2012	SI
Secretaria Municipal de Educação	Scratch	Fund	2012	SI
Universidade Federal da Paraíba	Scratch	Med	2013	SI
Universidade Federal de Pernambuco	Scratch	Tecn/Grad	2013	SI
Universidade Federal de Pernambuco	Scratch	Fund	2013	SI

Universidade Fed. Rural da Amazônia	Scratch	Grad	2013	SI
Universidade Fed Rural de Pernambuco	Scratch	Fund	2013	SI
Universidade Est. de Feira de Santana	Scratch	Grad	2013	SI
Instituto Federal de São Paulo	Scratch	Tec	2013	SI
Instituto Federal de Minas Gerais	Scratch	Grad	2013	SI
Universidade Estadual de Londrina	Scratch	Fund	2013	SI
Instituto Federal do Paraná	Scratch/ S4A	Tec	2013	SI
Universidade Federal de Pernambuco	Scratch	Grad/ Edu	2014	SI
Universidade de Pernambuco	Scratch	Fund	2014	SI
Instituto Federal do Amazonas	Scratch	Educ	2014	SI
Universidade de São Paulo USP	Scratch	Educ	2014	SI
USP / LSI-TEC	Scratch	Med	2014	SI
Universidade Fed. Rural da Amazônia	Scratch	Fund	2014	SI
Instituto Federal do Sertão PE	Scratch	Med/Téc n	2014	SI
UNICAMP	Scratch	Med	2014	SI
Instituto Federal Fluminense	Scratch	Fund	2014	SI
Instituto Federal Fluminense	Code.org	Fund/Me d	2014	SI
Universidade de Passo fundo	Scratch	Educ	2014	SI
Universidade de Passo Fundo	Scratch	Fund	2014	SI
Universidade Federal de Santa Catarina	Scratch	Fund	2014	SI
SENAI / Universidade do Extremo Sul Catarinense UNESC	Scratch/Co de.org	Tec	2014	SI

Universidade Federal do Rio Grande	Scratch	Med	2014	SI
Universidade Fed. Rio Grande do Norte	Scratch	Educ	2014	SI
Universidade do Estado da Bahia	Scratch	Fund	2014	I
Universidade Federal do Pará	Code.org	Med	2014	SI
Universidade Federal do Oeste do Pará	Scratch/ Code.org	Fund	2014	SI
Universidade Fed de Campina Grande	Scratch	Fund	2014	SI
Universidade Federal do Sergipe	Scratch	Med	2014	I
Universidade Federal da Paraíba	Scratch	Fund/Me d	2015	SI
Universidade Federal da Paraíba	Scratch	Med	2015	SI
Universidade de Pernambuco	Scratch	Fund	2015	SI
Instituto Federal do Amazonas	Scratch	Tec	2015	SI
Universidade de São Paulo USP	Scratch	Med	2015	I
Instituto Federal de Educação Ciência e Tecnologia do Sertão Pernambucano	Code.org	Fund/Me d	2015	SI
Universidade Fed Rural de Pernambuco	Scratch	Fund	2015	SI
Universidade Federal do Amazonas	Scratch	Grad	2015	SI
Universidade Est. de Feira de Santana	Scratch	Fund	2015	SI
UNICAMP	Scratch	Fund	2015	SI
Universidade Federal do Espírito Santo	Scratch/ Code.org	Med	2015	SI
Instituto Federal de Educação Ciência e Tecnologia Farroupilha	Scratch/ Code.org	Fund	2015	SI

UDESC/Universidade do Estado de SC	Scratch	Fund	2015	SI
Universidade Fed. Rio Grande do Sul	Scratch	Fund	2015	SI
Universidade Federal de Santa Maria	Scratch/ Code.org	Med	2015	SI
Universidade Federal do ABC	Scratch	Fund	2015	SI
Instituto Mauá de Tecnologia	Scratch	Med	2015	SI
FHO-UNIARARAS	Scratch/ Code.org	Grad	2015	SI
Instituto Federal Baiano	Scratch	Med	2015	SI
Universidade Fed da Grande Dourados	Scratch	Grad	2015	SI
Universidade Fed Mato Grosso do Sul	Scratch	Educ	2015	SI
Universidade do Estado do Amazonas	Scratch	Educ	2015	SI
Universidade Federal do Pampa	Scratch	Fund	2015	SI
Instituto Federal do Amazonas	Scratch	Fund/Tec	2016	SI
Universidade Est do Norte Fluminense	Scratch	Fund	2016	SI

Fund: Fundamental; Med: Médio; Tec: Técnico; Grad: Graduação
Fonte: Os autores

Apêndice 5.C Respostas dos questionários pré e pós curso na 1ª oferta.

Tabela 5.5 - Notas dos questionários pré e pós curso e avaliação dos alunos sobre grau de conhecimento na 1º oferta do curso (somente alunos concluintes).

P	Conhecimento Scraeth (pré-curso)	Conhecimento Scraeth (pós-curso)	Saldo	Conhecimento programação (pré-curso)	Conhecimento programação (pós-curso)	Saldo
1	1	5	4	3	3	0
2	1	5	4	1	5	4
3	1	5	4	1	4	3
4	1	5	4	3	5	2
5	4	5	1	5	5	0
6	3	5	2	2	4	2
7	1	5	4	1	5	4
8	1	5	4	2	5	3
9	1	4	3	1	3	2
10	1	4	3	2	4	2
11	1	5	4	3	5	2
12	1	5	4	1	5	4
13	1	5	4	1	3	2

P: Participante

Fonte: Os autores.

Apêndice 5.D Resultados estatísticos - 1º oferta.

Quadro 5.6 - Resultados do teste de normalidade Shapiro-Wilk.

Resultados	- 1 -	- 2 -	- 3 -	- 4 -
Tamanho da amostra =	13	13	13	13
Média =	1.3846	4.8462	2.0000	4.3077
Desvio padrão =	0.9608	0.3755	1.2247	0.8549
W =	0.4663	0.4456	0.8006	0.7437
p =	0.0057	0.0055	0.0098	0.0091

Fonte: Resultados dos cálculos dos autores no Software BioEstat 5.1.

Quadro 5.7 - Resultados do teste de Wilcoxon em relação ao conhecimento de Scratch

	Resultados Cols. 1 e 2
T =	0
Número de pares =	13
Z =	3.1798
p-valor (unilateral) =	0.0007
p-valor (bilateral) =	0.0015

Fonte: Resultados dos cálculos dos autores no Software BioEstat 5.3.

Quadro 5.7 - Resultados do teste de Wilcoxon em relação ao conhecimento em programação.

	Resultados Cols. 3 e 4
T =	0
Número de pares =	11
Z =	2.9341
p-valor (unilateral) =	0.0017
p-valor (bilateral) =	0.0033

Fonte: Resultados dos cálculos dos autores no Software BioEstat 5.3.

Apêndice 5.E Respostas dos questionários pré e pós curso na 2ª oferta.

Tabela 5.6 - Notas dos questionários pré e pós curso e avaliação dos alunos sobre grau de conhecimento na 2º oferta do curso (somente alunos concluintes).

Participantes	Conhecimento Scraeth (pré-curso)	Conhecimento Scraeth (pós-curso)	Saldo	Conhecimento programação (pré-curso)	Conhecimento programação (pós-curso)	Saldo
1	2	3	1	4	5	1
2	2	3	1	3	4	1
3	3	4	1	3	5	2
4	2	3	1	2	4	2
5	1	3	2	1	4	3
6	2	4	2	3	4	1
7	3	3	0	2	3	1
8	3	3	0	3	5	2
9	1	3	2	1	3	2
10	3	4	1	1	3	2
11	1	4	3	1	4	3
12	2	4	2	3	4	1
13	3	5	2	2	5	3
14	3	3	0	4	4	0
15	2	3	1	1	3	2
16	4	5	1	3	5	2
17	1	2	1	1	2	1

18	2	5	3	4	5	1
19	1	3	2	4	4	0
20	3	5	2	2	5	3
21	3	4	1	3	4	1
22	5	5	0	3	4	1
23	1	3	2	2	3	1
24	2	3	1	2	4	2
25	3	4	1	1	4	3
26	2	4	2	3	4	1
27	3	3	0	2	3	1
28	3	3	0	4	5	1
29	3	3	0	3	3	0
30	3	3	0	3	4	1
31	1	2	1	1	3	2
32	5	5	0	2	5	3
33	4	5	1	2	5	3
34	1	3	2	1	3	2
35	2	3	1	2	3	1
36	4	5	1	5	5	0
37	1	2	1	1	2	1
38	2	4	2	3	5	2
39	3	3	0	4	5	1
40	3	4	1	4	4	0
41	3	5	2	3	5	2

42	1	4	3	1	4	3
43	1	2	1	1	2	1
44	2	4	2	1	5	4
45	1	4	3	1	5	4
46	1	1	0	1	3	2
47	1	1	0	1	3	2
48	1	3	2	1	3	2
49	2	4	2	1	5	4
50	3	5	2	3	5	2
51	3	4	1	4	4	0
52	1	3	2	1	4	3
53	2	5	3	1	5	4
54	3	4	1	3	4	1
55	5	5	0	2	5	3

Fonte: Os autores.

Apêndice 5.F Resultados estatísticos referentes a 2^o oferta.

Quadro 5.8 - Resultados do teste de normalidade de Lilliefors.

	- 1 -	- 2 -	- 3 -	- 4 -
Tamanho da amostra =	55	55	55	55
Desvio máximo =	0.1800	0.2047	0.2243	0.2226
Valor crítico (0.05) =	0.1195	0.1195	0.1195	0.1195
Valor crítico (0.01) =	0.1390	0.1390	0.1390	0.1390
p(valor)	< 0.01	< 0.01	< 0.01	< 0.01

Fonte: Resultados dos cálculos dos autores no Software BioEstat 5.3.

Quadro 5.9 - Resultados do teste de Wilcoxon em relação ao conhecimento de Scratch.

	Resultados Cols. 1 e 2
T =	21
Número de pares =	42
Z =	5.3828
p-valor (unilateral) =	<0.0001
p-valor (unilateral) =	<0.0001

Fonte: Resultados dos cálculos dos autores no Software BioEstat 5.1.

Quadro 5.10 - Resultados do teste de Wilcoxon em relação ao conhecimento de programação.

	Resultados Cols. 3 e 4
T =	0
Número de pares =	49
Z =	6.0927
p-valor (unilateral) =	<0.0001
p-valor (unilateral) =	<0.0001

Fonte: Resultados dos cálculos dos autores no Software BioEstat 5.1.

Autores

Gabriel Gerber Hornink (organizador e coautor)



Doutor em Ciências, mestre em Biologia Funcional e Molecular (área Bioquímica, subárea Ensino), especialista em gestão ambiental, bacharel e licenciado em Ciências Biológicas, todos cursos pela Universidade Estadual de Campinas (Unicamp). Tem experiência na área de tecnologias educacionais, atuando no desenvolvimento de softwares educacionais, educação a distância e formação de professores. Professor efetivo na Universidade Federal de Alfenas (Unifal-MG), atuando na graduação (diversos cursos) e pós-graduação em Educação. Atua como líder do grupo de pesquisa Inovações Tecnológicas no Ensino. <http://lattes.cnpq.br/7615930937088442>

Augusto Márcio da Silva Júnior (coautor)



Mestre em educação pela Universidade Federal de Alfenas (Unifal-MG); MBA Executivo Internacional em gestão estratégica de TI (FGV-SP); Especialista em E-Commerce (PUC-MG); Bacharel em Ciências da Computação pela Universidade José do Rosário Vellano (2000). Pesquisador na área de tecnologias educacionais, ambientes virtuais de aprendizagem, redes de computadores e segurança da informação. Professor efetivo no Instituto Federal Sul de Minas; Coordenador do Laboratório de Redes, Educação e Sistemas (LARES); Coordenador da Academia Cisco do Câmpus Muzambinho, onde recebeu os prêmios - Cisco Instructor Excellence Award, nos anos de 2016 e 2017. <http://lattes.cnpq.br/2136263841685333>

Daniele de Almeida Dias (coautora)



Bacharel em Ciência da Computação pela Universidade Federal de Alfenas (Unifal-MG). Desenvolveu, durante o período de graduação, projetos de extensão e de iniciação científica que focaram no uso de tecnologias educacionais abordando os princípios de lógica de programação à iniciantes, inclusão digital, por meio de jogos digitais, para pessoas com necessidades especiais. Atualmente trabalha na área de qualidade de software. <http://lattes.cnpq.br/1114978500197546>

Lucas Faloni Ferreira (coautor)



Analista de Sistemas atuando em empresas de telecomunicações no desenvolvimento de automações. Aluno de mestrado na Universidade Estadual de Campinas (UNICAMP). Bacharel em Ciências da Computação pela Universidade Federal de Alfenas - MG (2012-2015). Atua e possui experiência em programação mobile e web.

Luis Eduardo de Matos (coautor)



Bacharel em Ciências da Computação pela Universidade Federal de Alfenas (UNIFAL-MG). Participou do projeto Pensando em Códigos como bolsista de Iniciação Científica e voluntário. <http://lattes.cnpq.br/7872190541781792>

Paulo Alexandre Bressan (coautor)



Possui graduação em Ciência da Computação pela Universidade Estadual Paulista Júlio de Mesquita Filho (1994), mestrado em Ciência da Computação pela Universidade Federal de São Carlos (1997) e doutorado em Engenharia Elétrica pela Universidade de São Paulo (2004). Atualmente é professor de 3º grau da Universidade Federal de Alfenas, com experiência na área de Ciência da Computação, ênfase em Computação Gráfica, atuando principalmente nos seguintes temas: tecnologia educacional, objetos virtuais de aprendizagem e ensino de física. <http://lattes.cnpq.br/0231239253253477>

Raphael Biavati Silva (coautor)



Bacharel em Ciência da Computação pela Universidade Federal de Alfenas, possui MBA em Engenharia e Inovação pela Universidade Estácio de Sá UniSEB. Atua na área de TI há mais de 6 anos, com extenso conhecimento técnico e vivência em projetos de médio e grande porte. Ampla experiência em projetos nas plataformas móveis SAP com foco nas áreas de manutenção, ativos e materiais.

Profissional certificado SAP (System Administration (MS SQL DB) with SAP NetWeaver 7.02), Scrum Master e ITIL. Experiência na gestão de projetos de desenvolvimento de Sistemas utilizando SCRUM.
<http://lattes.cnpq.br/7143911543209559>

Raquel Ribeiro Balbino (coautora)



Bacharel em Ciência da Computação pela Universidade Federal de Alfenas (UNIFAL-MG). Atualmente trabalha na empresa Ci&t, em Campinas, como Engenheira de Software. Tem experiência em pesquisa na área de Engenharia de Software, com ênfase em requisitos e qualidade de softwares.

<http://lattes.cnpq.br/6688097731576410>

Contato



**Laboratório de
Mídias Educacionais**

Construindo ideias bit a bit

01001100 01001101 01000101



**Departamento de Bioquímica
Instituto de Ciências Biomédicas
Universidade Federal de Alfenas**

Endereço:

R. Gabriel Monteiro da Silva, 700, sala E209D

CEP: 37.130-001 Alfenas-MG

Website: <http://www.unifal-mg.edu.br/lme>

Fone: +55 35 3701-9560

Email: lme.unifal@gmail.com

Contribuições da Computação para as Tecnologias Educacionais

Apresentamos neste livro algumas experiências trabalhadas por alunos do curso de Ciência da Computação, da Universidade Federal de Alfenas, abordando as tecnologias educacionais, iniciando com uma visão histórica, apresentando os desenvolvimentos de aplicativos para dispositivos móveis e para o Moodle, além de vivências com cursos abordando o pensamento computacional.

ISBN: 978-85-63473-33-2



9 78 85 63 47 33 32