

PROGRAMAÇÃO NO-CODE NO ENSINO SUPERIOR: POSSIBILIDADES DE AVALIAÇÃO E APRENDIZAGEM ATIVAS

PROGRAMACIÓN SIN CÓDIGO EN EDUCACIÓN SUPERIOR: EVALUACIÓN ACTIVA Y POSIBILIDADES DE APRENDIZAJE

NO-CODE PROGRAMMING IN HIGHER EDUCATION: ACTIVE EVALUATION AND LEARNING POSSIBILITIES

Jardel Lucas Garcia*, **
jardelgarcia.ti@gmail.com

*Centro Universitário UNA, Sete Lagoas/MG, Brasil

**Laboratório de Educação à Distância e eLearning – Universidade Aberta, Lisboa, Portugal

Resumo

Em cursos de graduação em tecnologia existe um senso comum quanto à complexidade de se desenvolver as competências no âmbito dos algoritmos e programação de computadores. Dessa forma, o presente trabalho teve como objetivo analisar a experiência de uma turma de um curso superior de tecnologia com programação No-Code, isto é, com plataformas de desenvolvimento de software que necessitam da escrita de pouco ou nenhum código-fonte para implementação dos aplicativos. Além disso, também constituiu objetivo deste trabalho promover uma prática pedagógica interdisciplinar, numa perspectiva de Aprendizagem Baseada em Projetos/Problemas (ABP) e numa perspectiva de avaliação participativa e por rubricas. Tais práticas ocorreram com uma turma de um Curso Superior de Tecnologia em Redes de Computadores. Foi proposto que, em grupos, estudassem uma ferramenta No-Code e construíssem, pelo menos, parte desse software através da plataforma definida para o seu grupo. Ao final desse processo, apresentaram a proposta, a ferramenta e o software desenvolvido e cada um dos grupos avaliou um ao outro e a si próprios com uma mesma rubrica. Com base nesses pressupostos, verificou-se que os alunos conseguiram obter a proficiência necessária para utilizar as ferramentas propostas e conseguiram operá-las de maneira satisfatória e autônoma com base na modelagem dos projetos que propuseram. Ainda, puderam perceber a importância da avaliação e da percepção de seus pares, elemento pedagógico essencial neste processo. As avaliações e o nível de compreensão demonstrados com as ferramentas e processos mostraram-se coerentes e diretamente proporcionais, constituindo, assim, uma experiência efetiva de aprendizagem ativa.

PALAVRAS CHAVE: Programação; Algoritmos; No-Code; Projetos; Avaliação por rubricas.

Resumen

En los cursos de pregrado en tecnología hay un sentido común sobre la complejidad de desarrollar habilidades en el área de algoritmos y programación de computadoras. De esta forma, el presente trabajo tuvo como objetivo analizar la experiencia de una clase de un curso de tecnología superior con programación No-Code, es decir, con plataformas de desarrollo de software que requieren la escritura de poco o ningún código fuente para implementar las aplicaciones. Además, también fue objetivo de este trabajo promover una práctica pedagógica interdisciplinar, desde una perspectiva de Aprendizaje Basado en Proyectos/Problemas (ABP) y desde una perspectiva de evaluación participativa y por rúbrica. Tales prácticas se dieron con una clase de un Curso Superior de Tecnología en Redes de Computadores. Se propuso que, en grupos, estudien una herramienta No-Code y construyan, al menos, parte de este software a través de la plataforma definida para su grupo. Al final de este proceso, presentaron la propuesta, la herramienta y el software desarrollado

y cada uno de los grupos se evaluó entre ellos y ellos mismos con la misma rúbrica. Con base en estos supuestos, se encontró que los estudiantes lograron obtener la competencia necesaria para utilizar las herramientas propuestas y pudieron operarlas de manera satisfactoria y autónoma a partir de la modelación de los proyectos que propusieron. Además, pudieron percibir la importancia de la evaluación y la percepción de sus pares, elemento pedagógico fundamental en este proceso. Las valoraciones y el nivel de comprensión demostrado con las herramientas y procesos resultaron ser coherentes y directamente proporcionales, constituyendo así una experiencia efectiva de aprendizaje activo.

PALABRAS CLAVE: Programación; algoritmos; Sin código; Proyectos; Evaluación por rúbricas.

Abstract

In undergraduate technology courses there is a common sense about the complexity of developing skills in the area of algorithms and computer programming. In this way, the present work aimed to analyze the experience of a class of a higher technology course with No-Code programming, that is, with software development platforms that require the writing of little or no source code to implement the applications. Furthermore, it was also the objective of this work to promote an interdisciplinary pedagogical practice, from a Problem Based Learning (PBL) perspective and from a participatory and rubric evaluation perspective. Such practices occurred with a class of a Higher Course of Technology in Computer Networks. It was proposed that, in groups, they study a No-Code tool and build, at least, part of this software through the platform defined for their group. At the end of this process, they presented the proposal, the tool and the software developed and each of the groups evaluated each other and themselves with the same rubric. Based on these assumptions, it was found that the students were able to obtain the necessary proficiency to use the proposed tools and were able to operate them satisfactorily and autonomously based on the modeling of the projects they proposed. Furthermore, they were able to perceive the importance of evaluation and the perception of their peers, an essential pedagogical element in this process. The assessments and the level of understanding demonstrated with the tools and processes proved to be coherent and directly proportional, thus constituting an effective experience of active learning.

KEYWORDS: Programming; Algorithms; No-Code; Projects; Assessment by rubrics.

1. Introdução

Trabalhar com projetos em qualquer nível de ensino é sempre um desafio. São necessárias competências diversas e conhecimento de metodologias e ferramentas que contemplem cada parte do processo, da concepção à avaliação. Sobre essa última, muitas vezes ela é deixada para o final do design educacional, o que não é uma boa ideia já que a avaliação precisa ser parte integrante de todo o processo que será desenvolvido (PORTO, 2005).

No que diz respeito aos cursos em que a tecnologia é a matéria, é preciso ter uma atenção especial à abordagem dos conteúdos. Um bom exemplo disso é o ensino de algoritmos e programação, tido como difícil e desafiador e que requer que os docentes tragam estratégias que fujam do tradicionalismo

conteudista e das metodologias estáticas e pouco diversificadas (GOMES; HENRIQUES; MENDES, 2008) para garantir uma aprendizagem significativa em um processo mais descomplicado.

Assim, o presente trabalho teve como objetivo utilizar plataformas de programação No-Code para desenvolver projetos com alunos do ensino superior numa perspectiva de aprendizagem ativa e avaliação participativa. Por meio desse processo, foi possível obter perspectivas tanto sobre os usos recomendados dessas plataformas quanto sobre propostas de avaliação em trabalhos com projetos.

2. Aprendizagem de programação e algoritmos: das linhas de código ao No-Code

Aprender algoritmos e programação já é, por si só, um desafio nos cursos de tecnologia em nível superior (GARCIA, 2021), ainda mais que tais disciplinas normalmente ocorrem nos primeiros semestres dos cursos de graduação. Contudo, aprender algoritmos e programação é realmente difícil?

De acordo com Roveda (2021), a resposta para essa pergunta não deve ser nem sim, nem não. O autor chama a atenção para três fatores determinantes e subjetivos que interferem nessa definição: são eles as aptidões pessoais, o tempo dedicado aos estudos de programação e a frequência com que esse estudo acontece - fatores esses, inclusive, que não se distinguem das demais áreas de estudo.

Na visão do Código Fonte TV (2021), o aprendizado de algoritmos e programação não pode ser considerado necessariamente fácil, pois depende muito de características e comportamentos subjetivos que variam muito de pessoa para pessoa. Destacam também o fator da motivação para aprender, o que também é variável, já que questões como gosto pessoal, empregabilidade e ganhos financeiros costumam figurar como principais elementos que levam as pessoas a buscar tal aprendizado. Além disso, os autores citam o fato de que a curva de aprendizagem neste campo não é linear, isto é, pode ser bem variável, por vários caminhos e através do uso de diversos artefatos. Como desafio principal, citam a necessidade de atualização constante, já que as tecnologias evoluem cada vez mais rápido.

A escolha da linguagem de programação também pode ser determinante para a continuidade ou o abandono do estudos, embora o foco inicial deve ser sempre na lógica de programação ao invés do aprendizado de linguagens e plataformas específicas (ROVEDA, 2021). Nessa perspectiva, Gomes, Henriques e Mendes (2008) realizaram um levantamento sobre os principais problemas que impactam a aprendizagem de programação e algoritmos e ressaltaram, também, que programar é muito mais do que escrever linhas de código: se trata de uma ciência que perpassa inúmeras disciplinas e áreas. Os autores identificaram muitas complexidades em ensinar justamente os conceitos básicos e essenciais da lógica de programação, o que leva a altas taxas de evasão e reprovação em disciplinas que envolvem o aprendizado de algoritmos. Criticam, dessa forma, as práticas pedagógicas que se utilizam de materiais unicamente estáticos e tradicionais, pouco dinâmicos, e as matrizes curriculares dos cursos que se concentram muito

em detalhes técnicos ao invés da aplicabilidade prática, da resolução de problemas, dos testes e reflexões sobre as soluções e técnicas propostas.

Como alternativa e como forma de ampliar e facilitar o acesso às técnicas e tecnologias de programação, nos últimos anos têm se popularizado muitas plataformas chamadas de No-Code. Conforme definido pela NoCode.Tech (s.d.), a programação No-Code - muitas vezes chamada também de programação visual - refere-se a ferramentas e plataformas que permitem construir software sem precisar escrever código - ou escrevendo muito pouco. Essas plataformas geralmente são compostas de interfaces visuais baseadas em elementos do tipo clique e arraste (drag-and-drop), o que facilita a construção dos componentes de software. Programar utilizando plataformas No-Code pode ser como jogar um jogo, mas está longe de se resumir a isso: elas devem ser vistas, então, como ferramentas poderosas para aprender, empoderar e possibilitar gerações a não apenas utilizar dispositivos digitais, mas também para mostrar o que fazer com eles, partilhar conhecimento e boas práticas e, acima de tudo, gerar impactos positivos na sociedade em que vivem (CORRAL; FRONZA; PAHL, 2021).

Contudo, é preciso ter atenção para o fato de que programação No-Code também é programação, isto é, ainda é necessário saber como os computadores funcionam e como dizer o que devem fazer, bem como integrar tecnologias distintas (NOCODE.TECH, s.d.). Percebe-se, dessa forma, que há desafios em qualquer tipo de programação. Woo (2020) cita, como exemplo, o fato de utilizar somente o drag-and-drop pode dificultar a atualização do software quanto a novos requisitos técnicos, isto é, pode tornar mais difícil o desenvolvimento de componentes e funcionalidades específicas. Citam também a questão da performance, que pode ser reduzida em comparação com outros softwares desenvolvidos por código, e da integração, que pode ser dificultada dada a ausência do código.

Contudo, tendo em mente tanto as potencialidades quanto os desafios, há de se destacar o potencial educacional e a facilidade de adentrar o mundo da programação por meio do No-Code. Em seu trabalho, Corral, Fronza e Pahl (2021) demonstraram, ao longo de 10 anos, que, por meio do ensino de programação com o uso de plataformas No-Code, saíram da mera transferência de conhecimento para uma cultura de trabalho em equipe, colaboração e criação ativa e de aprendizagem constante.

Assim, as plataformas No-Code - e também as Low-Code, isto é, aquelas que baseiam-se no visual mas que ainda há algum código - são verdadeiras agentes de democratização do desenvolvimento de software em todos os níveis, conforme apontam Corral, Fronza e Pahl (2021). Por meio delas, não é necessário ser um grande expert em programação, o que abre portas para o seu ensino em diferentes níveis de escolaridade e idades. Por isso, os autores lembram também que o No-Code, embora considerado uma tendência contemporânea, não é algo novo: tecnologias como Lego, Scratch e LogoBlocks já estão no mercado há anos e implementam as mesmas ideias.

Nesse viés, Smiderle e Andreis (2019) citam como elemento essencial da educação contemporânea - pelos autores referida como Educação 4.0 - a própria programação, dado o seu potencial de desenvolver o raciocínio lógico, a organização pessoal e a resolução de problemas. Chamam a atenção, contudo, para a necessidade de que o ensino da programação não seja abstrato, isto é, que as práticas pedagógicas envolvam a concretização de ideias e não só a repetição de conceitos. Essa abordagem, já comum em várias instituições de ensino que possuem condições para tal, torna possível o trabalho com projetos integradores, isto é, com metodologias que privilegiam a criatividade ao propor problemas a serem resolvidos por meio da tecnologia e da programação, o que tem o potencial de promover uma aprendizagem significativa e ativa. Assim, percebe-se a importância do alinhamento entre técnica, ferramenta, metodologia e avaliação para garantir uma tal aprendizagem.

3. Exemplos de plataformas No-Code disponíveis atualmente

O *Adalo* é uma plataforma No-Code que possui uma interface simples baseada em drag-and-drop, muito semelhante a outros aplicativos de prototipação, isto é, para construção de protótipos de tela para outros softwares. Ele possui uma versão gratuita que permite construir um número ilimitado de aplicativos na plataforma e publicá-los no domínio da própria ferramenta. Contudo, essa versão limita a quantidade de registros que podem ser gravados em cada aplicativo desenvolvidos em apenas 50 registros. Isso faz com que a versão seja ideal para aprender sobre programação No-Code, para testar ideias para aplicativos e disponibilizar versões de teste de maneira rápida.

Já o *AppGyver* é uma plataforma totalmente gratuita (a não ser por grandes corporações que possuem elevado capital) e tem uma proposta parecida com a do *Adalo*. Contudo, fornece algumas dinâmicas de configurações a mais - como diversas fontes de dados das quais o usuário pode extrair seus dados, um editor de fórmulas e a possibilidade de publicar aplicativos para diversos dispositivos, como smartphones, computadores, televisões, etc. Também funciona com a lógica do drag-and-drop e pode ser acessado através do navegador.

Com uma proposta diferente, o *AppSheet* propõe transformar uma planilha eletrônica em um aplicativo. É uma proposta bem interessante e rápida de se construir software a partir de um banco de dados organizado previamente. Ele suporta integração com *Microsoft Office*, *Google Sheets*, *Dropbox*, entre outras plataformas populares e gera as telas com base nas planilhas contidas nos arquivos importados. Além de criar as telas, é possível customizá-las e adaptar a interface de cada uma. Contudo, o *AppSheet* é uma plataforma paga. Sua versão gratuita suporta criar aplicativos para até, no máximo, dez usuários.

E como uma das plataformas No-Code mais utilizadas atualmente, há o *Bubble*. Já com milhões de usuários ao redor do mundo, o *Bubble* é uma ferramenta muito completa com diversos recursos que

também se utiliza do drag-and-drop e disponibiliza uma série de configurações e recursos para gerenciar os softwares desenvolvidos, que vão desde sistemas web a aplicativos móveis. Possui também uma versão gratuita limitada a alguns recursos e versões pagas que ampliam muito as possibilidades de desenvolvimento. O site da plataforma demonstra uma série de softwares já desenvolvidos por meio dela..

4. Aprendizagem baseada em projetos e avaliação por rubricas

Retomando o que foi mencionado anteriormente através do trabalho de Smiderle e Andreis (2019), ensinar algo como programação exige um desenho metodológico que não seja abstrato e nem estático, que privilegie a prática significativa e que possua um modelo avaliativo coerente, formativo e perceptível. Neste caso, considera-se a Aprendizagem Baseada em Projetos (ABP) e a avaliação por rubricas como os dois conceitos adequados para o trabalho realizado com programação No-Code.

A Aprendizagem Baseada em Projetos/Problemas (ABP, ou Problem-Based Learning) é definida por Bender (2014) como uma metodologia de ensino e aprendizagem por meio da qual os estudantes devem analisar, selecionar e trabalhar sobre problemas do mundo real, propostos em sala de aula junto ao professor, a fim de propor soluções e contribuir tanto para seu conhecimento próprio quanto até mesmo para a sua comunidade. O autor reitera que a ABP pressupõe o desenvolvimento da criatividade, da colaboração, da cooperação e do aprendizado e motivação para resolução de problemas.

Nessa mesma linha, Bacich e Moran (2018) destacam que a ABP deve possuir um caráter transdisciplinar, já que deve instigar o aluno a problematizar, levantar hipóteses, investigar, discutir e mobilizar diferentes competências adquiridas de diversas fontes e disciplinas. Isso contribui para que o ensino não seja compartimentalizado em conteúdos isolados. Bender (2014) também lembra que a ABP geralmente começa com a proposição de uma questão problema - em geral, proposta pelo professor - que motivará os estudantes a buscar respostas em diversos âmbitos e combiná-las na forma de um projeto.

Inclusive, no contexto do ensino superior, Garcia, Barbosa e Mehlecke (2022) trazem a ABP como uma ferramenta para potencializar as práticas pedagógicas tanto no âmbito no ensino quanto da extensão. Em seu trabalho, combinaram a ABP com um modelo avaliativo formativo baseado em competências para inserir a extensão universitária no currículo de um curso de graduação. Concluíram que o trabalho com projetos possibilita a integração dos conteúdos numa perspectiva transdisciplinar sem direcioná-los para uma esfera extracurricular, como normalmente costuma acontecer.

Com isso, percebe-se que uma metodologia baseada em projetos precisa de uma sistemática de avaliação que seja coerente com essa proposta. Garcia, Barbosa e Mehlecke (2022) descrevem, conforme mencionado, uma metodologia de avaliação formativa, isto é, com critérios bem definidos para uma

avaliação contínua que não privilegia somente entregas pontuais e aspectos quantitativos, como muitas metodologias tradicionais. De acordo com Bender (2014), um trabalho com projetos gera entregas, atividades, momentos, experiências e circunstâncias distintas, o que demanda uma avaliação que considere todos esses aspectos e os combine de forma concisa.

Sobre isso, Irala, Blass e Junqueira (2021) trazem uma pergunta essencial sobre o processo decisório por uma ou outra metodologia de avaliação: o que realmente avaliamos? Os autores ressaltam que as instituições de ensino normalmente só permitem uma avaliação da performance do estudante, isto é, as suas flutuações de comportamento, de conhecimento e de desempenho que acontecem logo após adquirirem novas informações, o que geralmente é mensurado com critérios quantitativos. Com base nesse pressuposto, propõem, em seu trabalho, o conceito de avaliação por rubricas, sendo elas entendidas como dispositivos de meta-avaliação que explicitam intenções, pilares, dimensões, níveis que comunicam as expectativas dos professores quanto às aprendizagens dos alunos, sendo elas instrumentos que podem ser construídos junto deles.

Moreira et al (2020) descrevem uma rubrica como sendo uma descrição detalhada dos aspectos que serão avaliados. Essa descrição vem com escalas com diferentes níveis de desempenho que podem ser atingidos e com uma descrição objetiva de cada um desses níveis. Essa escala de níveis e descrições funciona, então, como uma sistematização para se chegar às notas de maneira mais detalhada do que simplesmente fornecer uma pontuação única, o que apenas daria uma impressão geral sem detalhar todo o processo avaliativo (PORTO, 2005).

Bender (2014) se utiliza justamente desses mecanismos para descrever formas de se avaliar numa perspectiva de aprendizagem baseada em projetos. De acordo com o autor, uma rubrica é adequada para a ABP pois fornece os níveis desejados de detalhamento para as soluções que devem ser propostas para o problema e servem como instrumento avaliativo para os vários artefatos produzidos pelos estudantes.

5. Metodologia

Com base em todos esses pressupostos, no contexto de um Curso Superior de Tecnologia em Redes de Computadores em uma instituição de ensino superior privada brasileira, foi desenvolvido um trabalho com projetos, avaliação participativa por rubricas e programação No-Code com uma turma de doze estudantes. Cabe ressaltar que as práticas aqui descritas ocorreram durante o período da pandemia da COVID-19 no ano de 2021, isto é, as aulas ocorreram em modo online.

Uma das motivações para essas práticas é o fato de que o curso de Redes de Computadores não possui como foco central a programação, como vários outros cursos de tecnologia, mas sim as telecomunicações. Isso quer dizer que a sua matriz curricular não se focou muito no ensino de programação

e algoritmos de maneira aprofundada. Contudo, como a área de tecnologia é extremamente abrangente e interconectada, houve um esforço dos docentes em inserir diversos tópicos relacionados à grande área em disciplinas da matriz que permitiam tal ação.

Uma dessas ações foi a inserção da programação No-Code. Essa escolha se deu por dois fatores: pela contemporaneidade do assunto, que se encaixava nos temas e intencionalidade da disciplina, e pela baixa curva de aprendizado que a programação No-Code requer, o que permitiu trabalhar os projetos de maneira mais ágil e prática. Ambos os fatores, combinados com a abordagem pouco voltada para a programação no curso, foram decisivos para a implementação pedagógica dessa parte da ementa da disciplina, o que compôs 30 horas dela.

Com base nessas definições, a turma de doze alunos foi dividida em quatro grupos de três estudantes cada. Em uma sessão síncrona, foram apresentados aos conceitos de programação No-Code, seu histórico, plataformas disponíveis e exemplos. Foi dado um prazo de uma semana para os alunos consumirem os materiais disponibilizados sobre o assunto, realizarem alguns exercícios e testarem alguns dos exemplos e plataformas demonstrados. Em seguida, o docente definiu uma plataforma No-Code para cada um dos quatro grupos, sendo elas *Adalo*, *AppGyver*, *AppSheet* e *Bubble*. Como tarefa, cada grupo ficou responsável por: 1) estudar e explorar a ferramenta e seus recursos; 2) desenvolver um aplicativo com base nas orientações dadas pelo professor; 3) construir uma apresentação sobre a ferramenta utilizada, demonstrando suas características; 4) apresentar o aplicativo desenvolvido; 5) avaliar os seus pares, isto é, cada grupo dar notas aos demais grupos após a apresentação, e autoavaliar-se com base em uma rubrica construída em conjunto.

Para isso, o docente propôs a seguinte questão-problema (de forma resumida) para nortear os projetos: “Você foi contratado(a) como programador(a) para criar um aplicativo com um prazo muito curto para uma determinada empresa. O(a) dono(a) do negócio não é versado em tecnologia e não tem noção da complexidade com que um software é produzido, mas ele(a) quer obter uma primeira versão do aplicativo pronto em três semanas. Proponha o desenvolvimento desse aplicativo utilizando a plataforma No-Code designada para o seu grupo. Quais recursos puderam ser desenvolvidos nesse período? Como foi o processo? Apresente o produto e o processo ao final do período.”.

Embora se trate de uma questão fictícia, o problema colocado pelo docente é algo bem comum no mundo da tecnologia e os alunos tiveram exatamente três semanas para desenvolver o trabalho. Ao longo desse período, o professor orientou e forneceu informações e recursos necessários para que os estudantes pudessem concluí-lo, sem, contudo, intervir nas suas produções. Reforçou que, na apresentação final dos projetos, cada grupo deveria explicar bem tanto o processo (assim como a plataforma utilizada) quanto o produto obtido.

Além disso, o professor propôs a criação de uma rubrica para avaliar os projetos. Introduziu o conceito de avaliação por rubricas, de avaliação por pares e de autoavaliação em aulas síncronas e, junto aos estudantes, foi construída a rubrica exibida na Figura 1.

Figura 1. Rubrica construída em conjunto para avaliar os projetos

Critério/Nível	Atende totalmente	Atende parcialmente	Não atende
Concepção	O aplicativo apresenta uma boa e criativa ideia para a sua concepção, embasado em situações reais do dia-a-dia (0,25)	O aplicativo apresenta elementos simples, de criatividade mediana, que representam parcialmente alguns elementos do dia-a-dia. (0,15)	O aplicativo não apresenta muita criatividade, não se embasou em situações comuns do dia-a-dia. (0,0)
Usabilidade	O aplicativo é de fácil utilização, intuitivo e claro, podendo ser utilizado por qualquer pessoa sem dificuldades. (0,25)	O aplicativo é razoavelmente fácil de utilizar, apresentando aparente facilidade de utilização mediante orientações. (0,15)	O aplicativo é confuso, sua interface não é intuitiva, não respondendo bem ao usuário conforme o esperado. (0,0)
Implementação	O aplicativo utiliza bem os recursos da plataforma em que foi desenvolvido, sendo o produto final adequado ao tempo e às condições que foram dadas para sua implementação. (0,25)	O aplicativo se utiliza de alguns dos recursos disponíveis na plataforma em que foi desenvolvido, apresentando uma implementação razoavelmente satisfatória em função do tempo e condições que foram dados para sua implementação. (0,15)	O aplicativo desenvolvido não utiliza as potencialidades da plataforma em que foi desenvolvido, poderia ser melhor, já que não corresponde com as expectativas de desenvolvimento em função do tempo e das condições que foram dadas para a sua implementação. (0,0)
Apresentação	A apresentação do grupo foi satisfatória, clara e criativa, preocupando-se com o público e demonstrando todos os aspectos necessários para conhecer tanto o produto quanto o processo. (0,25)	A apresentação do grupo foi boa, demonstrando o aplicativo desenvolvido sem detalhar muito sobre os demais aspectos da sua construção (0,15) .	A apresentação do grupo não demonstrou preocupação com o público que assistiu, não detalhando o percurso nem do produto e nem do processo. (0,0)

De acordo com a Figura 1, foi definido que a avaliação se daria sob quatro critérios básicos bem comuns do contexto da programação: a concepção, isto é, a percepção quanto à ideia base do software desenvolvido, se está bem estruturada e se corresponde ao problema apresentado; a usabilidade, que se refere ao nível de facilidade/dificuldade de utilização do aplicativo desenvolvido; a implementação, que diz respeito ao uso efetivo das potencialidades e recursos disponíveis para o desenvolvimento do software; e a apresentação, que nada mais é do que o momento em que o grupo apresentaria o processo e o produto. Cada um desses critérios foi avaliado em “atende totalmente”, “atende parcialmente” e “não atende”, sendo que esses níveis seriam, por sua vez, transpostos para um quantitativo final para compor a nota do grupo. Foi acordado, então, que a rubrica seria utilizada tanto para avaliação do professor (que correspondeu a 60% da nota final), avaliação dos pares (20% da nota final) e autoavaliação (também 20% da nota final).

5. Resultados e discussão

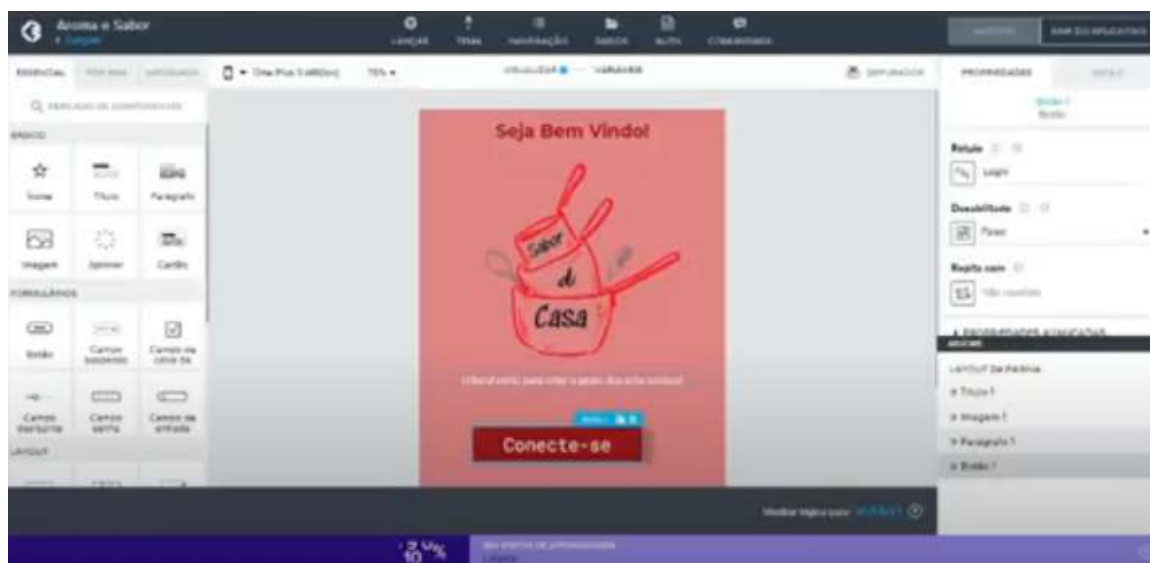
Conforme previsto nos procedimentos metodológicos utilizados, os estudantes concluíram o trabalho no período estipulado. Cada um dos grupos conseguiu desenvolver um aplicativo com a plataforma No-Code designada, apresentou o processo e o produto, bem como realizou as etapas de avaliação solicitadas.

O primeiro grupo utilizou o *Adalo*. Os estudantes desenvolveram um aplicativo cuja proposta foi gerenciar tarefas, isto é, um organizador de rotinas. Utilizaram a questão da pandemia da COVID-19 como fator motivador para que uma empresa solicitasse a criação de uma ferramenta como tal. Criaram as funções básicas de um gerenciador de tarefas: login, lista de tarefas com descrição, data e horário, e a possibilidade de inserção, alteração e deleção de tarefas (FIGURA 2).

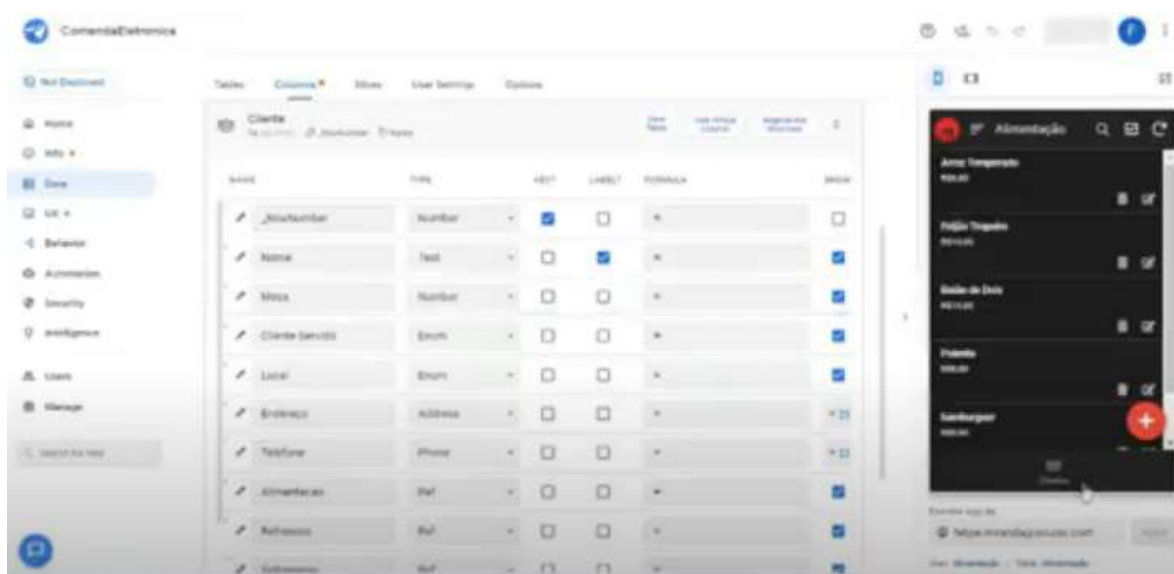
Figura 2. Aplicativo gerenciador de tarefas construído no *Adalo*



Já o segundo grupo realizou o seu trabalho com base na plataforma *AppGyver*. Neste caso, abordaram um segmento diferente de negócio: o gastronômico. Utilizaram como problema central os restaurantes que tiveram que se reinventar em decorrência também da pandemia da COVID-19, automatizando seus processos para atender com eficiência, com pessoal reduzido e com o mínimo de contato pessoal. O aplicativo consistiu, então, de um cardápio digital (chamado por eles de *Sabor di Casa*), que tem como funcionalidades: realizar pedidos no restaurante, chamar um garçom em pessoa (caso necessário), montar o seu prato e realizar o pagamento. Utilizaram vários recursos do *AppGyver* e disponibilizaram para teste da turma na forma de um aplicativo web, isto é, acessível por navegador (FIGURA 3).

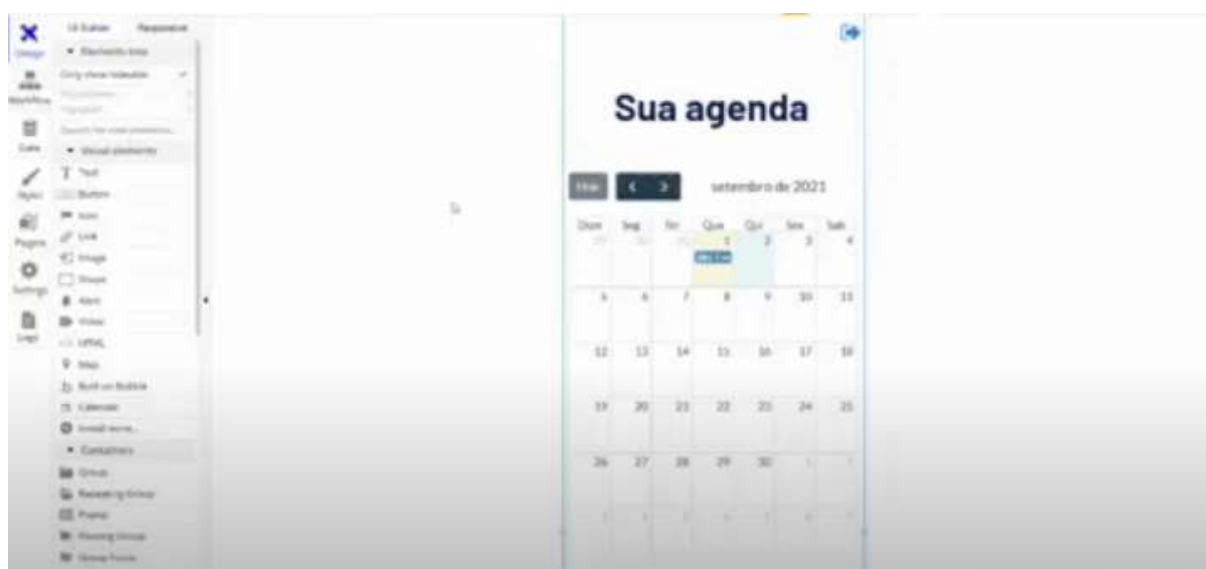
Figura 3. Tela de construção do aplicativo no *AppGyver*

De maneira semelhante, o terceiro grupo também abordou a temática dos restaurantes em tempos de pandemia com a plataforma *AppSheet*. No caso dessa ferramenta, a proposta de construção é diferente - o aplicativo é desenvolvido a partir de planilhas eletrônicas. Os estudantes assim o fizeram: criaram as planilhas no *Google Sheets* e as configuraram no *AppSheet*. O problema que buscaram resolver foi a questão das entregas de comida, uma modalidade de trabalho que muitos restaurantes tiveram que adotar em um curto espaço de tempo e em caráter emergencial. O aplicativo desenvolvido por eles envolveu as funcionalidades de realizar um pedido completo, calcular o preço final e fechar o pedido (FIGURA 4).

Figura 4. Tela de construção do aplicativo no *AppSheet*.

Por fim, o quarto grupo desenvolveu o trabalho utilizando o *Bubble*. De maneira parecida com o primeiro grupo, este também pensou em algo relacionado à organização de tarefas profissionais para colaboradores de uma empresa. A proposta foi criar uma agenda com um formato semelhante a outros aplicativos (como o *Google Agenda*) mas com uma interface diferente e exclusiva para grupos de trabalho individuais. Criaram as funções de login, de criar eventos e de visualizá-los (FIGURA 5).

Figura 5. Tela de construção do aplicativo no *Bubble*.



Os quatro aplicativos desenvolvidos foram apresentados em aula síncrona e, ao final de cada apresentação, foram realizadas as devidas avaliações de acordo com a rubrica descrita na Figura 1. Os resultados da avaliação do professor, da avaliação dos pares e da autoavaliação foram compilados pelo docente e divulgados em momento posterior à aula síncrona, acompanhada de um feedback para cada grupo.

Dado esse retorno, o professor também solicitou um feedback aos alunos sobre a sua experiência nesse percurso. Esse momento foi realizado em nova aula síncrona em que se fez uma gravação. De maneira unânime, os estudantes afirmaram ter conseguido utilizar as plataformas designadas com um esforço relativamente baixo - isto é, eles tiveram consciência de que já eram estudantes de tecnologia e que, por isso, o uso de uma ferramenta No-Code se torna ainda mais fácil. Afirmaram que, embora outros estudantes de outros cursos possam ter opiniões diferentes quanto à curva de aprendizado, tais plataformas de programação sem código são excelentes instrumentos educacionais sobretudo para introduzir as pessoas ao universo do desenvolvimento de software.

Uma outra afirmação importante de alguns dos estudantes foi que pretendem explorar essas e

outras plataformas No-Code em seus trabalhos subsequentes, pois possuem ideias e planos para futuros softwares e que seria muito mais complicado em termos de tempo de esforço construir as versões de teste em ferramentas de programação por linhas de código. Um dos alunos, inclusive, citou a possibilidade de construir protótipos mais assertivos e completos por meio delas em pouco tempo. Contudo, foi apontado por outros que o No-Code não pode se resumir à construção de protótipos pois a sua proposta é muito mais ampla.

Nesse viés, os estudantes também apontaram que sentem que essa área de programação No-Code ainda parece efervescente, isto é, cresce a cada dia, mas ainda precisa de consolidação. O professor perguntou sobre as razões que os levaram a pensar assim e responderam que sentiram um pouco de dificuldade de customizar elementos prontos das plataformas e de encontrar referências ou discussões na internet sobre o que precisavam fazer. Discutiu-se que tais problemas podem ser realmente decorrentes tanto do uso de versões gratuitas da maioria das ferramentas utilizadas quanto pelo fato de serem relativamente novas, assim como suas comunidades de uso, o que faz com que haja menos conteúdo em rede do que plataformas mais antigas e consolidadas.

Concluída essa primeira fase de feedback, o professor questionou-os quanto à metodologia de avaliação utilizada no trabalho. Um estudante salientou que se sentiu, ao mesmo tempo, intimidado e responsável, isto é, enxergou a avaliação como algo que gera um desconforto mas que confere responsabilidade àquele(a) que avalia. Uma outra aluna ressaltou que o trabalho avaliativo não é fácil, já que várias dimensões devem ser analisadas e não apenas uma - e, muitas vezes, quando vêm os números em suas notas, eles(as) não têm essa visão. Outros estudantes relataram que o ato de avaliar os trabalhos dos colegas refletiu exatamente o que irão fazer no mercado de trabalho: analisar, desenvolver e avaliar projetos que já estão em andamento ou iniciar outros. Além disso, alguns alunos identificaram nos critérios de avaliação utilizados na rubrica elementos estudados em outras disciplinas ligadas à programação, o que fez com que relacionassem imediatamente os conteúdos.

Assim, os pressupostos levantados por Bender (2014) na discussão sobre aprendizagem baseada em projetos se refletiram bem neste trabalho ao combinar a resolução de problemas, o trabalho em equipe e as metodologias de avaliação numa perspectiva de projetos interdisciplinares. Assim, a programação No-Code foi empregada tanto de modo a cumprir com a matriz curricular esperada quanto para desenvolver competências gerais e interdisciplinares.

7. Considerações finais

O presente estudo mostrou uma possibilidade de trabalho com programação de forma descomplicada através do uso de plataformas No-Code. O trabalho desenvolvido mostrou que é possível

utilizar tais tecnologias com pessoas com diferentes níveis de proficiência em tecnologia e que tais ferramentas se configuram tanto como plataformas comerciais quanto como artefatos educacionais.

Dessa forma, concluiu-se o percurso deste trabalho atribuindo às plataformas No-Code as seguintes características, do ponto de vista dos estudantes: são excelentes ferramentas educacionais para ensinar programação, com baixa curva de aprendizado; servem muito bem para construção de protótipos e versões iniciais de softwares maiores, embora não se resumam a isso; são necessários estudos e uma difusões maiores das plataformas No-Code para que sua popularidade, recursos e comunidades cresçam e se consolidem.

Em relação à avaliação, é preciso sempre planejar uma metodologia de avaliação coerente com a proposta pedagógica, com as ferramentas e com a intencionalidade das práticas a serem realizadas. No caso do trabalho com projetos, neste caso, a avaliação por rubricas se mostrou eficiente tanto para avaliar diferentes dimensões de diferentes projetos quanto para distribuir a responsabilidade do processo avaliativo entre os estudantes e não centrá-la apenas na figura do professor. Assim, tem-se metodologias de aprendizagem e avaliação ativas e participativas.

Referências

ADALO. Disponível em: <https://www.adalo.com>. Acesso em: 2 mar. 2022.

APPGYVER. Disponível em: <https://www.appgyver.com>. Acesso em: 2 mar. 2022.

APPSHEET. Disponível em: <https://about.appsheet.com/home>. Acesso em: 2 mar. 2022.

BACICH, Lilian; MORAN, José. **Metodologias ativas para uma educação inovadora**: uma abordagem teórico-prática. Penso Editora, 2018.

BENDER, Willian N. **Aprendizagem baseada em projetos**: educação diferenciada para o século XXI. Penso Editora, 2015.

BUBBLE. Disponível em: <https://bubble.io/>. Acesso em: 2 mar. 2022.

CÓDIGO Fonte TV. Programação é Fácil ou Difícil? A VERDADE! // Vlog #144. YouTube, 2021. Vídeo. Disponível em: <https://www.youtube.com/watch?v=Qr-7Be1wW2E>. Acesso em: 16 fev. 2022.

CORRAL, Luis; FRONZA, Ilenia; PAHL, Claus. Block-based Programming Enabling Students to Gain and Transfer Knowledge with a No-code Approach. In: SIGITE '21: PROCEEDINGS OF THE 22ST ANNUAL CONFERENCE ON INFORMATION TECHNOLOGY EDUCATION. 2021. Proceedings [...]. 2021, p. 55-56. Disponível em: <https://doi.org/10.1145/3450329.3478314>. Acesso em: 27 fev. 2022.

GARCIA, Jardel Lucas. Aprendizagem colaborativa de programação e algoritmos com Repl.it: potencialidades multimodais. In: CONGRESSO MOVIMENTOS DOCENTES. 2021. Anais [...]. 2021. 342 p. Disponível em: <https://www.vveditora.com/anaiscmd>. Acesso em: 20 fev. 2022.

GARCIA, Jardel Lucas; BARBOSA, Marcus Vinícius; MEHLECKE, Querte Teresinha Conzi. Extensão, projetos e avaliação: pilares para uma aprendizagem significativa no ensino superior. **Revista Valore**, v. 7, n. 2, p. 1-22, 2022. Disponível em: <https://revistavalore.emnuvens.com.br/valore/article/view/1108>. Acesso em: 25 mar. 2022.

GOMES, Anabela; HENRIQUES, Joana; MENDES, António. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. **Educação, Formação & Tecnologias**, v. 1, n. 1, p. 93-103, 2008. Disponível em: <http://eft.educom.pt/index.php/eft/article/view/23>. Acesso em: 21 fev. 2022.

IRALA, Valesca Brasil; BLASS, Leandro; JUNQUEIRA, Sonia Maria da Silva. Introduzindo o conceito de avaliação por rubricas por intermédio de oficinas: análise de uma experiência piloto. **Revista Contexto e Educação**, v. 36, n. 113, 2021. Disponível em: <https://www.revistas.unijui.edu.br/index.php/contextoeducacao/article/view/10083>. Acesso em: 01 mar. 2022.

MOREIRA, José António et al. **Educação digital em rede: princípios para o design pedagógico em tempos de pandemia**. 1 ed. Lisboa: Universidade Aberta, v. 1, 2020. 49 p. Disponível em: <https://repositorioaberto.uab.pt/handle/10400.2/9988>. Acesso em: 28 mar. 2022.

NOCODE.TECH: What is NoCode?. Vídeo. Disponível em: <https://www.nocode.tech/lessons/what-is-nocode>. Acesso em: 25 fev. 2022.

PORTO, Stella. A avaliação da aprendizagem no Ambiente On-line. In: CARNEIRO, Roberto; BORGES, Paula; SILVA, Anabela Vidigal da. **Educação, aprendizagem e tecnologias: um paradigma para professores do Século XXI**. 1 ed. Lisboa: Edições Sílabo, f. 111, 2005. 222 p. cap. 6, p. 139-161.

ROVEDA, Ugo. **Programação para iniciantes: é difícil aprender a programar?**. Kenzie. 2021. Disponível em: <https://kenzie.com.br/blog/programacao-para-iniciantes/#:~:text=%C3%89%20muito%20dif%C3%ADcil%20aprender%20a,aptid%C3%B5es%20pessoais%20para%20determinadas%20disciplinas>. Acesso em: 15 fev. 2022.

SMIDERLE, Eduardo; ANDREIS, André Augusto. Programação de Arduino por meio da Metodologia Ativa Aprendizagem Baseada em Projetos para uma Educação 4.0. In: 4º SALÃO DE PESQUISA, EXTENSÃO E ENSINO DO INSTITUTO FEDERAL DO RIO GRANDE DO SUL (IFRS). 2019. 2019. Disponível em: https://eventos.ifrs.edu.br/index.php/Salao_IFRS/4salao/paper/view/8233. Acesso em: 26 fev. 2022.

WOO, Marcus. The Rise of No/Low Code Software Development: No Experience Needed?. **Elsevier Public Health Emergency Collection**, v. 6, n. 9, p. 960-961, 2020. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7361109/>. Acesso em: 26 fev. 2022.

Recebido em: 27/10/2022

Aceito em: 06/05/2023

Endereço para correspondência:

Nome: Jardel Lucas Garcia

Email: jardelgarcia.ti@gmail.com



Esta obra está licenciada sob uma [Licença Creative Commons Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)