



DESENVOLVIMENTO DE UM SITE PARA O ENSINO DE PROGRAMAÇÃO

DEVELOPMENT OF A WEBSITE FOR TEACHING PROGRAMMING

Ismael M. A. de Souza

Rodrigo dos Santos

Thomas N. Henrique

Juliano Schimiguel

Carlos Adriano Martins

DOI: 10.5281/zenodo.17172341

Resumo

Este trabalho propõe um site interativo para ensinar programação, visando superar as dificuldades de acesso e compreensão das informações nesta área em constante evolução. Com o avanço tecnológico, a demanda por conhecimentos em programação cresce, porém muitos enfrentam barreiras para aprender. O objetivo é oferecer um ambiente de aprendizado acessível, contribuindo para disseminar o conhecimento tecnológico. A metodologia empregada foi de revisão bibliográfica, análise de dados estatísticos, pesquisa de campo e aplicação de princípios de engenharia de software. A facilidade de acesso ao conteúdo pode motivar mais pessoas a explorar a programação, impulsionando seu crescimento pessoal e profissional e ajudando-as a adquirir habilidades relevantes.

Palavras-Chave: aprendizado interativo; acessibilidade de conteúdo; motivação para programação

Abstract

This paper proposes an interactive website to teach programming, aiming to overcome the difficulties of accessing and understanding information in this constantly evolving area. With technological advances, the demand for programming knowledge grows, but many face barriers to learning. The goal is to offer an accessible learning environment, contributing to the dissemination of technological knowledge. The methodology used was a bibliographic review, statistical data analysis, field research and application of software engineering principles. Easy access to content can motivate more people to explore programming, boosting their personal and professional growth and helping them acquire relevant skills.



Keywords: interactive learning; content accessibility; motivation for programming

INTRODUÇÃO

Na era digital em que vivemos, a demanda por profissionais qualificados em programação atingiu um ponto crucial em diversos setores da sociedade. Este cenário é evidenciado por pesquisas recentes, como o levantamento realizado pela Associação das Empresas de Tecnologia da Informação e Comunicação e de Tecnologias Digitais, que revelou um crescimento expressivo na contratação de profissionais de TI, atingindo mais de 43.624 mil contratações desde janeiro de 2021, representando um aumento de 183,2% em relação a 2020. Entretanto, mesmo com esse esforço, a projeção aponta a necessidade de cerca de 159 mil profissionais a cada novo ano até 2025 para suprir a crescente demanda.

Esta lacuna entre a demanda e a oferta de profissionais qualificados em programação destaca a urgência em buscar soluções inovadoras para capacitar e instruir aqueles interessados em aprender programação. Este artigo propõe a criação de um site dedicado ao ensino de programação, visando preencher essa lacuna educacional. A proposta é oferecer uma abordagem abrangente, acessível e interativa, abordando desde conceitos fundamentais até tópicos avançados em diversas linguagens de programação.

No atual contexto, a rápida evolução tecnológica e a demanda crescente por soluções digitais têm gerado uma escassez significativa de profissionais qualificados em programação. O estudo da Brasscom destaca a complexidade desse desafio, indicando a necessidade de estratégias inovadoras para suprir o mercado com profissionais capacitados.

A importância de abordar essa questão reside na relevância social e econômica de investir em iniciativas que promovam a educação em programação. Um corpo qualificado de profissionais em programação não apenas atende às demandas do mercado, mas também impulsiona a inovação, contribuindo para o desenvolvimento tecnológico e a competitividade global.

O objetivo principal deste trabalho é propor o desenvolvimento de um site educacional dedicado ao ensino de programação, oferecendo uma abordagem abrangente e acessível para capacitar indivíduos interessados em adquirir habilidades nessa área.

Este trabalho fundamenta-se em pesquisas acadêmicas recentes, como os artigos "Mapeamento Sistemático sobre Metodologias Ferramentas de apoio para o Ensino de



Programação " (Blatt, Lucas, 2017) e " Percepções do ensino de programação em cursos de tecnologia da informação: o que pensam discentes e docentes " (Neto, Oscar, 2021) e "Princípios de Análise e Projeto de Sistemas com UML. 2." (Bezerra, 2006).

O artigo está estruturado da seguinte forma: na seção 2, Desafios Educacionais e Impactos; na seção 3, Cyberspace: Expandindo Horizontes, Conectando Conhecimento; na seção 4, Conceito de engenharia de software, na seção 5, Metodologia; na seção 6, Estudo de caso; 7. Uso De Engenharia De Software Para Criação De Um Site Voltado Ao Ensino De Programação e por fim na seção 8, Considerações finais.

DESAFIOS EDUCACIONAIS E IMPACTOS

A falta de profissionais qualificados em programação é um desafio significativo enfrentado pela indústria atualmente. A rápida evolução das tecnologias e a demanda crescente por soluções digitais têm criado uma lacuna entre a oferta e a procura de profissionais com habilidades especializadas. Essa escassez de talentos pode afetar negativamente o desenvolvimento e o desempenho das organizações, dificultando a implementação de projetos de TI e limitando a capacidade de inovação.

A falta de didática adequada por parte dos educadores de Tecnologia da Informação (TI) pode ter um impacto significativo no desempenho e na motivação dos novos estudantes, levando a uma desistência prematura.

Quando os educadores não conseguem transmitir de forma clara e envolvente os conceitos complexos os alunos podem se sentir desestimulados e desencorajados a continuar no campo.

Dentre as tecnologias digitais, grande parte das ferramentas mais recentes são construídas com uso de linguagens de programação, como Java, C, Python, Ruby, entre várias outras, cada uma com sua peculiaridade e objetivo. Sendo ministradas e aprendidas em cursos do segmento, que apesar da fundamental importância dessas competências ainda é encontrado dificuldades e barreiras na metodologia do ensino e aprendizagem de conteúdos indispensáveis a assimilação do conteúdo, caso das disciplinas de algoritmo e lógica de programação que são disciplinas em que os discentes são apresentados aos conceitos iniciais



do segmento, em que são encontrados conteúdos precursores da programação, como a lógica de programação, ou seja, é inserido a metodologia de projeção e construção da resolução da problemática e é com base nesta metodologia que será desenvolvida a solução, fazendo uso de uma das linguagens de programação possíveis, ou adequadas para a demanda a ser tratada.

Apesar da necessidade constante do mercado necessitar de cada vez mais de profissionais com competências na área do desenvolvimento, devido à complexidade evidenciada na docência e ao aprendizado de disciplinas vinculadas a programação e a lógica de programação, que as taxas de desistências e reprovações de discentes nestas matérias são elevadas (Souza; et al., 2016) De forma que a metodologia de ensino se configura como um desafio, pois, além do fator da complexidade requerida da própria disciplina, ainda deve-se considerar a heterogeneidade da turma, que pode apresentar níveis de conhecimento prévio, dificuldades e ritmo de aprendizado variado, ou seja, a questão não é apenas como passar um conteúdo dinâmico (pois depende da sequência lógica de cada pessoa para se chegar a uma mesma solução), mas, como transmitir este conteúdo da forma mais eficaz possível para perfis cada vez mais variados.

Portanto, sabe-se que a técnica tradicional de ensino não se aplica a disciplina de lógica de programação, gerando contínua reprovação, além do estímulo negativo a ingresso, pois, desta forma percorre a falsa ideia que é impossível aprender programação, desincentivando os alunos a iniciar na área (França; Amaral, 2013).

CYBERSPACE: EXPANDINDO HORIZONTES, CONECTANDO CONHECIMENTO

A Plataforma Cyberspace nasce com o objetivo de servir como uma ponte entre a inovação tecnológica e os desafios enfrentados pela crescente e exigente comunidade interessada em programação. A plataforma oferece cursos, eventos com o objetivo de expandir o conhecimento e habilidades dos participantes, preparando-os para oportunidades de emprego e oferecendo um ambiente colaborativo para troca de ideias e experiências.

A Cyberspace conta com especialistas voluntários de diversas linguagens de programação que compartilham seus conhecimentos e experiências, contribuindo para o desenvolvimento de uma comunidade profissional altamente qualificada e atualizada. Através



desse ambiente colaborativo, a Cyberspace ajuda a expandir a rede de contatos dos participantes, destacando a importância do networking no mundo profissional. O impacto da Cyberspace na indústria de tecnologia será significativo, pois tem o potencial de influenciar o desenvolvimento de novas tecnologias e soluções. Em resumo, a Cyberspace é uma plataforma online que oferece um ambiente colaborativo para que os participantes possam expandir seus conhecimentos e rede profissional, com o potencial de impactar o desenvolvimento da indústria de tecnológica.

CONCEITOS DE ENGENHARIA DE SOFTWARE

O termo software refere-se a programas desenvolvidos para determinado fim. É um mecanismo para automatizar negócios, indústria e governo. O software é central para quase todo aspecto do negócio. Para Pressman (2002, p. 828), *“software é uma dentre poucas tecnologias importantes que terão um impacto significativo em praticamente todo o aspecto da sociedade moderna.*

O software passa a ter uma importância para toda a organização, pois torna-se *“um método de captar conhecimento, um modo de diferenciar os produtos de uma empresa de seus competidores e uma janela para o conhecimento coletivo de uma corporação.”* (Pressman, 2002, p.828). A engenharia de software, para Sommerville (2003, p. 5), é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação.

Analisando este conceito, Sommerville (2003, p. 5-6), destaca nele dois momentos: “disciplina de engenharia”, que diz respeito a aplicação de teorias, métodos e ferramentas apropriadas em momentos apropriados; e “todos os aspectos da produção de software”, o que elucidada que a engenharia de software não se encarrega apenas dos processos de desenvolvimento de software, mas também do gerenciamento de projetos de software e do desenvolvimento de mecanismos de apoio a produção de software. Pressman (2002, p.22) define a engenharia de software como: “o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais”.

Dos conceitos apresentados, a produção de software é abordada como um produto resultado de um processo de engenharia. Um processo de desenvolvimento de software pode



ser visto como um conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software. Um processo eficaz deve, claramente, considerar as relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido. Para ser eficaz, um processo deve ser adequado ao domínio da aplicação e ao projeto específico. Deste modo, processos devem ser definidos caso a caso, considerando-se as especificações da aplicação, a tecnologia a ser adotada na sua construção, a organização onde o produto será desenvolvido e o grupo de desenvolvimento.

Segundo Souza (2003, p. 99), “*a engenharia de software é um domínio altamente orientado ao conhecimento, no qual os fatores de sucesso estão relacionados com a experiência das pessoas envolvidas nas seguintes fases: projeto, construção, teste e implantação*”.

O uso da engenharia de software na criação de um site voltado ao ensino de programação é essencial para garantir o desenvolvimento eficiente, escalável e de alta qualidade. O processo envolve várias etapas, desde o levantamento de requisitos até a manutenção contínua do site.

ENGENHARIA DE REQUISITOS

A criação e sucesso de um software dependem e muito de uma engenharia de requisitos bem elaborada e definida. Uma das definições mais aceitas para a engenharia de requisitos é a de Zave (1997), ele diz que a engenharia de requisitos está relacionada com a identificação das metas a serem atingidas pelo sistema, o qual será desenvolvido, assim como a operacionalização dessas metas em serviços e restrições. Outros autores definem a engenharia de requisitos de forma mais técnica.

CASO DE USO E CENÁRIOS

- Casos de Uso

A técnica de modelagem através de casos de uso está sendo utilizada cada vez mais pelos engenheiros de requisitos. Idealizada, na década de 70, pelo engenheiro de software sueco, Ivar Jacobson, tem sido de grande utilidade para a documentação dos requisitos



funcionais de um sistema (Bezerra, 2006). A modelagem baseada em casos de uso possui uma notação gráfica simples e uma descrição em linguagem natural o que facilita a comunicação entre o desenvolvedor e o usuário. Foi incorporada pela Linguagem de Modelagem Unificada (UML) que é representada pelo diagrama de casos de uso (Bezerra, 2006). Uma das características mais importantes dos casos de uso é que independentemente de sua forma, um caso de uso descreve o software ou o sistema do ponto de vista do usuário (Pressman, 2006). Os modelos de casos de uso são compostos por atores, casos de uso e relacionamentos entre eles. Um caso de uso é a representação de uma sequência de interações entre um sistema e os atores. Os atores são agentes externos que utilizam esse sistema (Bezerra, 2006).

- Cenários

Um dos meios para se fazer um levantamento de informações visando desenvolver um novo sistema é através da construção de cenários. Essa técnica corresponde a uma descrição parcial do comportamento da aplicação. Ela pode ocorrer em um dado momento e em uma situação específica. Utiliza-se de uma coleção de narrativas de situações no domínio para levantar informações, identificar problemas e, com isso, resolver problemas antecipadamente.

Os cenários devem ser elaborados durante as reuniões entre todos os interessados em determinado projeto (*stakeholders*). Segundo Sommerville (2007), de maneira geral, o cenário deve possuir: uma descrição do que os usuários esperam do sistema no início do cenário; uma descrição do fluxo normal de eventos no cenário; uma descrição do que pode ocasionar em erro e como isso será tratado; relatar atividades que podem ocorrer simultaneamente e; uma descrição do estado de sistema no fim do cenário.

DIAGRAMA DE SEQUÊNCIA

Um diagrama de sequência é um diagrama Unified Modeling Language (UML) que ilustra a sequência das mensagens entre objetos em uma interação. Um diagrama de sequência consiste em um grupo de objetos representados por linhas de vida e as mensagens que eles trocam durante a interação.

Um diagrama de sequência mostra a sequência de mensagens transmitidas entre



objetos. Diagramas de sequência também mostram as estruturas de controle entre objetos.

PROTOTIPAÇÃO

Para Sommerville (2011), dentre muitos modelos de processos de software existentes, a prototipação chama atenção por facilitar o entendimento dos requisitos, além de deixar mais claro os conceitos e funcionalidades de uma aplicação. Para isso, a interface deve apresentar mecanismos sociais em comunicação e colaboração que forneçam suporte para que pessoas exerçam atividades juntas, sejam de trabalho, aprendizagem ou lazer (Preece; Rogers; Sharp, 2005).

Os autores acima mencionados destacam que ainda a prototipação possibilita que os stakeholders interajam com um produto imaginado visando a adquirir alguma experiência de como utilizá-la em um ambiente real e explorar os usos por ele imaginado. As autoras também relatam que os protótipos são muito úteis quando se estão discutindo ideias com stakeholders, sendo dispositivos que facilitam a comunicação entre os membros das equipes e que consistem em uma maneira eficaz de testar as ideias para você mesmo.

Sommerville (2011) relata que a prototipação também é uma parte essencial do processo de projeto da interface de usuário em função da natureza dinâmica das interfaces, descrições textuais e diagramas que não são bons o suficiente para expressar seus requisitos. Por esse motivo, os objetivos devem ser explicitados desde o início do processo, caso contrário, a gerência ou os usuários finais podem não entender a função do protótipo ou não obter os benefícios que esperavam. Corroborando com esse dizer, Preece; Rogers e Sharp (2005), destacam que é importante projetar sistemas interativos que deem suporte a diversas formas de socialização, verificando as tecnologias existentes e examinando maneiras de aprimorá-las para melhor atender às necessidades dos usuários.

- Prototipação De Baixa Fidelidade

Preece, Rogers e Sharp (2005) definem Prototipação de baixa fidelidade como aquela que não se assemelha ao produto final e utiliza materiais diferentes da versão final pretendida. O desenvolvedor faz uso de papel e cartolina em vez de telas eletrônicas e metal, pois são úteis porque tendem a ser simples, baratos e de rápida produção, isso também significa que podem ser rapidamente modificados, oferecendo suporte à exploração e ideias alternativas.



Tais vantagens são importantes nos primeiros estágios de desenvolvimento, uma vez que os protótipos utilizados para explorar ideias devem ser flexíveis e encorajar a exploração e a modificação. Porém, dizem os referidos autores que nunca são projetados para ser mantidos e integrados ao produto final, servem apenas para a exploração. A prototipação de baixa fidelidade quase sempre se baseia na realização de esboços. Partindo dessa premissa, podemos dizer que, esse tipo de protótipo possui baixo grau de detalhamento, uma vez que só apresenta visualmente sua funcionalidade, não possui recursos de interação, e nem exhibe, necessariamente, seu aspecto visual definitivo.

- Prototipação De Alta Fidelidade

Segundo Preece, Rogers e Sharp (2005), esse protótipo utiliza matérias que você espera que estejam no produto final e realiza um protótipo que se parece muito mais com algo acabado. É útil para vender ideias a pessoas e para testar questões técnicas. Pode ser desenvolvido em algum programa gerador de interface ou em alguma linguagem de programação que permita acelerar o processo e é útil para realizar avaliação com usuário, assim como para servir de especificação para o desenvolvimento, ou seja, uma ferramenta de marketing usada para incentivar investidores.

Os autores ainda acrescentam que os problemas de projetos realizados com prototipação de alta fidelidade ocorrem porque levam muito tempo para ser construídos, o que faz com que revisores e aplicadores dos testes estejam tendenciosos em comentar aspectos superficiais, ao invés do conteúdo; os desenvolvedores relutam em mudar algo no qual trabalharam artesanalmente por horas, pois um protótipo em software pode elevar demais expectativas; é necessário apenas um bug para interromper o teste.

METODOLOGIA

Segundo os trabalhos de Marconi e Lakatos (2008), na abordagem qualitativa, o pesquisador tem por finalidade mostrar as opiniões e sugestões de determinado grupo de indivíduos em relação a um fenômeno específico, visando proporcionar na sua descrição e explorá-lo minuciosamente para melhor compreendê-lo. Agora em relação à pesquisa quantitativa, Roesch (2006) explica que esta procura obter informações sobre determinada população, como quantos membros da organização possuem a mesma opinião sobre



determinado assunto. Quanto à pesquisa descritiva, segundo Gil (2006) esse tipo de pesquisa tem como objetivo primordial a descrição das características de determinada população ou fenômeno, ou, então, o estabelecimento de relações entre variáveis.

- Revisão Bibliográfica

A revisão bibliográfica foi conduzida para compreender as tendências atuais no ensino de programação, identificar desafios enfrentados por educadores e alunos, e explorar iniciativas inovadoras que abordem a escassez de profissionais qualificados em TI. Artigos acadêmicos, livros e recursos online foram consultados, com foco em estudos relevantes sobre métodos de ensino de programação, demanda por profissionais de TI e estratégias para preencher essa lacuna.

- Análise de Dados Estatísticos

A análise de dados estatísticos baseou-se em fontes confiáveis, como pesquisas da Brasscom, para quantificar a demanda por profissionais de TI no mercado brasileiro. Essa análise proporcionou insights sobre a magnitude do desafio enfrentado pelas empresas na busca por profissionais qualificados em programação.

- Pesquisa de Campo

A pesquisa de campo consistiu em entrevistas com profissionais de TI, visando obter perspectivas práticas sobre os desafios no ensino de programação e as necessidades do mercado de trabalho. Essa abordagem qualitativa permitiu uma compreensão mais profunda das questões enfrentadas no cenário atual.

- Aplicação de Princípios de Engenharia de Software

A engenharia de software desempenha um papel crucial na criação de soluções eficazes para os desafios identificados. A aplicação de princípios de engenharia de software envolveu a definição de requisitos funcionais e não funcionais para o site educacional, o desenvolvimento de casos de uso, a criação de diagramas de sequência e a elaboração de cenários de uso. Essa abordagem sistemática assegurou uma estrutura sólida para a implementação da plataforma de ensino.

- Validação da Proposta

A validação da proposta foi realizada por meio de feedbacks de especialistas em educação, profissionais de TI e potenciais usuários da plataforma de ensino. Esse processo de validação proporcionou insights adicionais para aprimorar a proposta, garantindo sua



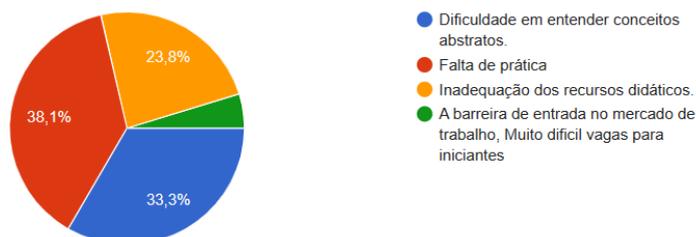
viabilidade e eficácia na prática.

ESTUDO DE CASO

O presente trabalho foi desenvolvido com o intuito de entender o comportamento do público-alvo e validar a necessidade do desenvolvimento do site. A pesquisa foi respondida por estudantes do curso de ciência da computação da universidade cruzeiro do sul de semestre variáveis.

Nesse gráfico podemos ver que 38,1% falta de prática 33,3 dificuldades em entender conceitos abstratos. 23,8% inadequação de recursos didáticos. O Gráfico indica que o nosso público-alvo, prefere desafios práticos, devemos implementar no site.

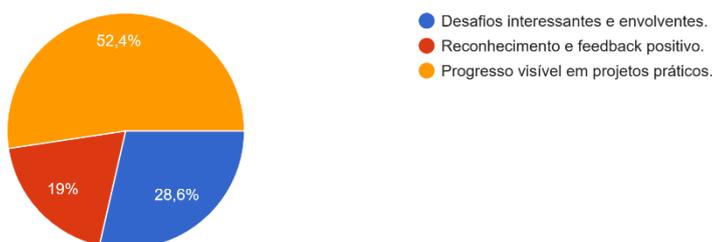
1- Qual destes desafios você considera mais impactante ao aprender programação? [Copiar](#)
21 respostas



Fonte: os autores

Nesse gráfico podemos ver que 52% dos entrevistados preferem um progresso visível em projetos práticos, 19% de reconhecimento e feedback positivo e 28% preferem desafios interessantes e envolventes. Através disso sabemos o que mais impacta o nosso público positivamente.

2- O que mais impacta positivamente sua motivação para estudar programação?
21 respostas

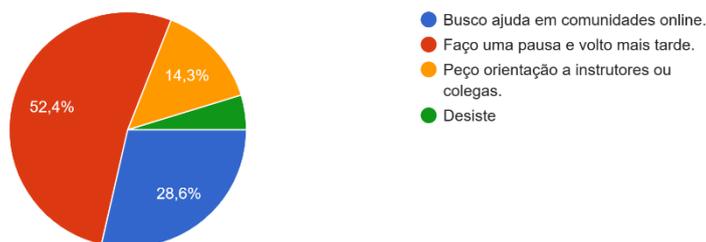


Fonte: os autores



Nesse gráfico podemos ver que 52,4% fazem uma pausa e volta mais tarde 28,6% busca comunidades online 14,3%. Através desse gráfico podemos observar que a maioria faz uma pausa, podemos explorar essa preferência para inclusão futura de recursos voltado a isso no site.

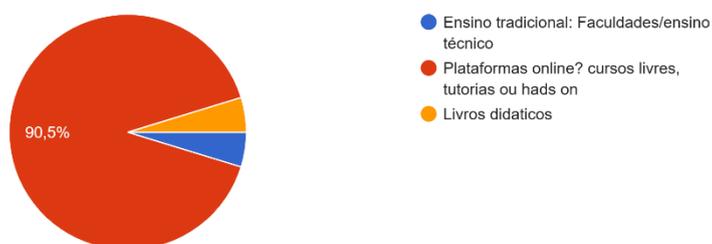
3 - Como você lida com momentos de desânimo ou dificuldade ao aprender programação?
21 respostas



Fonte: os autores

Nesse gráfico podemos ver uma esmagadora maioria totalizando 90,5% preferem usar plataformas online.

4- Qual a sua maior fonte de aprendizagem de programação?
21 respostas



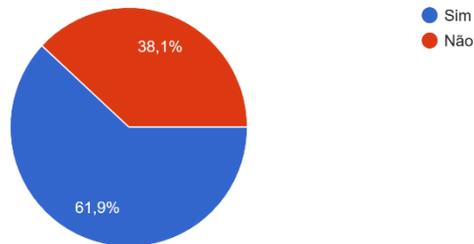
Fonte: os autores

Nesse gráfico podemos ver que 61,9% a totalizando maioria já teve vontade de desistir da área de programação, a expectativa é que nosso site mude esse cenário.



5- Em algum momento teve vontade de desistir da área devido algum tipo de dificuldade no aprendizado?

21 respostas

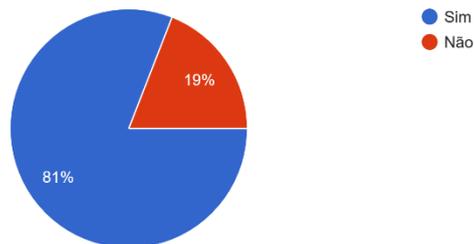


Fontes: os autores

Nesse gráfico podemos ver que 81% Assinariam a nossa plataforma e 19% não.

6- Se existisse uma plataforma voltado ao ensino de programação, com recursos didáticos e interativo. você assinaria?

21 respostas

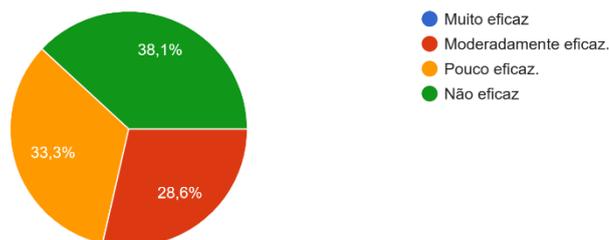


Fonte: os autores

Nesse gráfico podemos ver que 38,1% acham a abordagem de ensino tradicional não eficaz. 33,3% acham pouco eficaz 28,6% moderadamente eficaz. Demonstra que a maioria não está feliz com a metodologia tradicional.

7- Como você avalia a eficácia da abordagem tradicional de ensino de programação?

21 respostas



Fonte: os autores



USO DE ENGENHARIA DE SOFTWARE PARA CRIAÇÃO DE UM SITE VOLTADO AO ENSINO DE PROGRAMAÇÃO

- Cyberspace Engenharia de requisitos Requisitos Funcionais

1) Cadastro de Usuário:

- a) Os usuários devem poder se cadastrar no site.
- b) O sistema deve validar e armazenar as informações do usuário.

2) Perfis de Usuário:

- a) Diferentes níveis de acesso para alunos, instrutores e administradores.
- b) Funcionalidades específicas para cada perfil (ex.: administração de cursos para instrutores, acompanhamento do progresso para alunos).

3) Cursos e Módulos:

- a) Os cursos devem ser categorizados por linguagem de programação, nível de dificuldade, etc.
- b) Cada curso deve ser dividido em módulos ou lições.

4) Conteúdo de Aprendizagem:

- a) Disponibilização de videoaulas, tutoriais escritos, quizzes e exercícios práticos.
- b) Fóruns ou chats para discussão sobre o conteúdo.

5) Acompanhamento de Progresso:

- a) Rastreamento do progresso do aluno em cada curso.
- b) Geração de relatórios e estatísticas de desempenho.

6) Busca e Navegação:

- a) Capacidade de pesquisar cursos, módulos e lições.
- b) Navegação intuitiva pelo site.

Requisitos Não Funcionais:

7) Desempenho:

- a) O site deve carregar rapidamente para garantir uma boa experiência do usuário.
- b) Suporte a um grande número de usuários simultâneos.

8) Segurança:

- a) Proteção dos dados do usuário.
- b) Implementação de autenticação segura.



9) Compatibilidade:

- a) Compatibilidade com diferentes navegadores (Chrome, Firefox, Safari, etc.).
- b) Responsividade para dispositivos móveis.

10) Escalabilidade:

- a) Capacidade de expandir para incluir mais cursos e usuários.

11) Disponibilidade:

- a) Mínimo de tempo de inatividade.
- b) Estratégias de backup e recuperação de dados.

12) Usabilidade:

- a) Interface de usuário intuitiva e amigável.
- b) Acessibilidade para usuários com necessidades especiais.

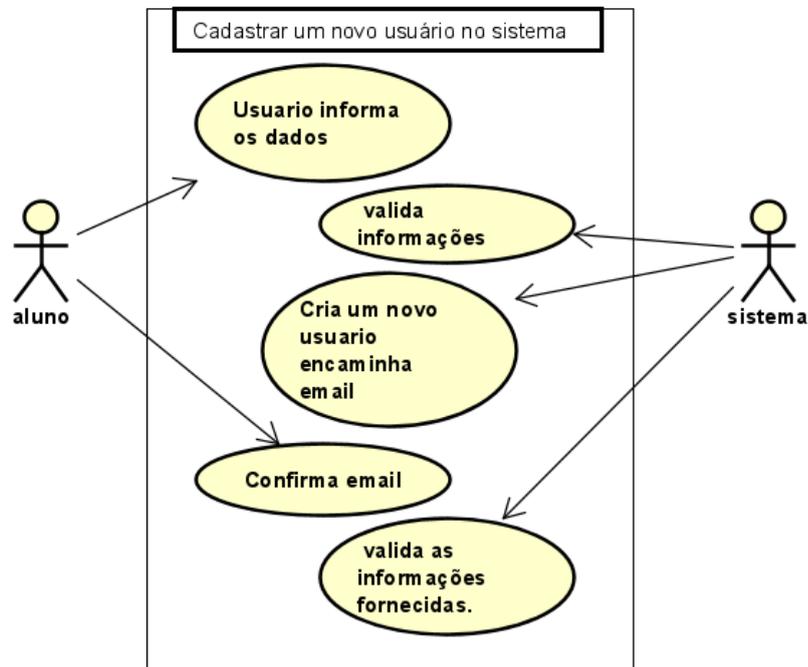
13) Manutenibilidade:

- a) Facilidade de atualização do conteúdo do curso.
- b) Documentação clara para futuras manutenções.

14) Conformidade:

- a) Conformidade com regulamentações de privacidade e segurança.

7.2 Caso de uso e cenários



Caso de uso 1: Cadastrar usuário

Cenário de Caso de uso 1: Cadastrar usuário

Ator: Usuário

Objetivo: Cadastrar um novo usuário no sistema.

Pré-condições: Nenhuma.

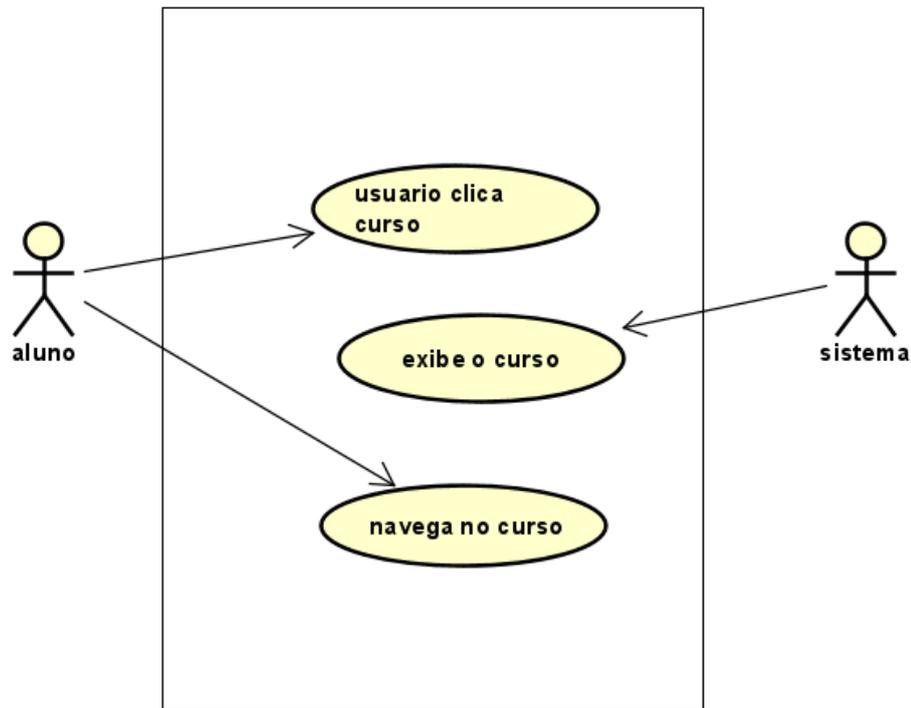
Fluxo principal:

1. O usuário fornece seu nome, e-mail, senha e outras informações necessárias.
2. O sistema valida as informações fornecidas.
3. O sistema cria um novo usuário e envia um e-mail de confirmação para o usuário.
4. O usuário confirma o endereço de e-mail clicando no link no e-mail de confirmação.
5. O sistema ativa a conta do usuário.

Pós-condições: O usuário está cadastrado no sistema.

Fluxos alternativos:

- Se as informações fornecidas forem inválidas, o sistema exibirá uma mensagem de erro.
- Se o usuário já estiver cadastrado, o sistema exibirá uma mensagem de erro.



Caso de uso 2: Visualizar curso

Cenário de Caso de uso 2: Visualizar curso

Ator: Usuário

Objetivo: Visualizar informações sobre um curso, incluindo título, descrição, módulos, lições, requisitos, etc.

Pré-condições: O usuário deve estar autenticado no sistema.

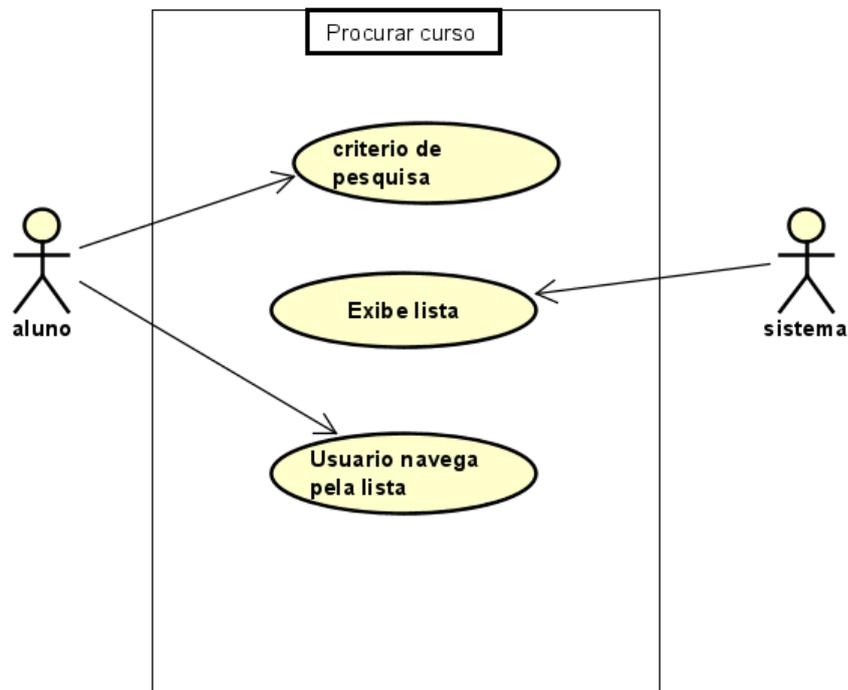
Fluxo principal:

1. O usuário clica no link do curso que deseja visualizar.
2. O sistema exibe uma página com informações sobre o curso.
3. O usuário pode navegar pelas informações do curso.

Pós-condições: O usuário tem acesso às informações sobre o curso.

Fluxos alternativos:

- Se o usuário não estiver autenticado, o sistema o redirecionará para a página de login.
- Se o curso não existir, o sistema exibirá uma mensagem de erro.



Caso de uso 3: Procurar curso

Cenário de Caso de uso 3: Procurar curso

Ator: Usuário

Objetivo: Procurar um curso por título, descrição, linguagem de programação, nível de dificuldade ou área de conhecimento.

Pré-condições: O usuário deve estar autenticado no sistema.

Fluxo principal:

1. O usuário fornece os critérios de pesquisa.
2. O sistema exibe uma lista de cursos que correspondem aos critérios de pesquisa.
3. O usuário pode navegar pela lista de cursos.

Pós-condições: O usuário tem acesso a uma lista de cursos que correspondem aos critérios de pesquisa.

Fluxos alternativos:

- Se os critérios de pesquisa não forem fornecidos, o sistema exibirá uma lista de todos os cursos.

- Diagrama de sequência

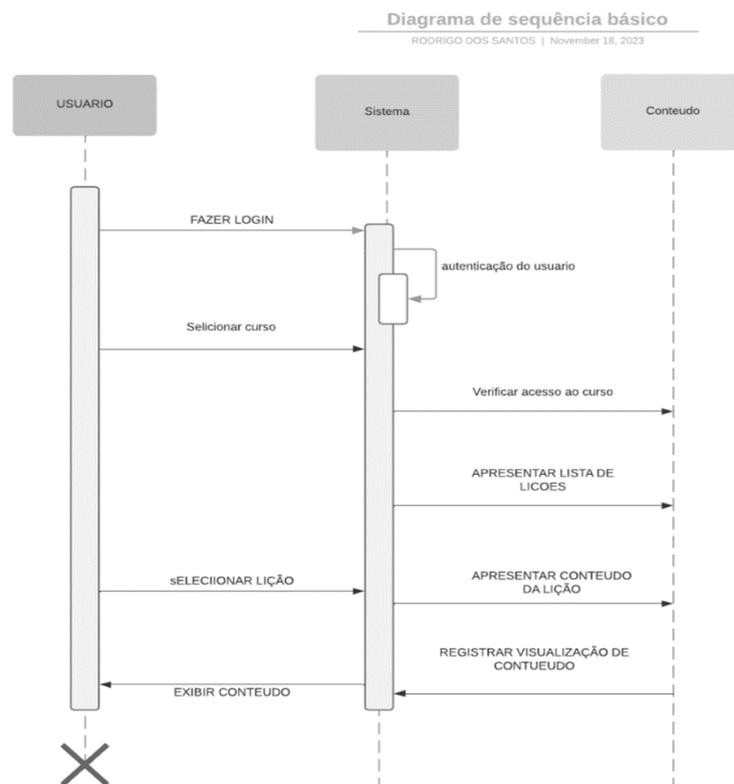


Um diagrama de sequência ilustra a interação entre objetos ao longo do tempo. No contexto do caso de uso "Assistir conteúdo" o sistema durante esse processo.

Usuário (Aluno): Representa o aluno que inicia o caso de uso.

Sistema: Representa as interações com o sistema, incluindo autenticação do usuário.

Conteúdo: Representa a interação com o conteúdo



1ª Imagem: Diagrama de Sequência

- Prototipação de alta fidelidade Cyberspace

- Página inicial

1. Header:

- Nome do site à esquerda.
- Menu de navegação à direita com opções como, "Cursos", "Sobre", "login", "cadastre".

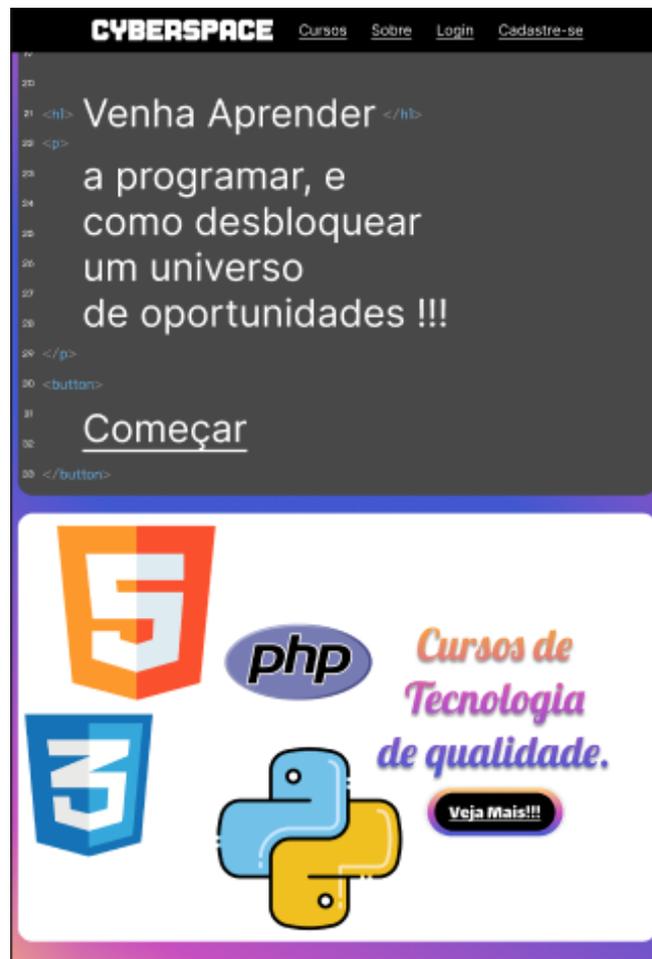
2. Banner Principal:

- Mensagem destacada centralizada: "Venha aprender a programar, e como desbloquear



um universo de oportunidade!!!"

- Mensagem em destaque: "Cursos de tecnologia de qualidade"
- Botão "veja mais"



1ªImagem: Página Inicial

- Área de login

1. Header:

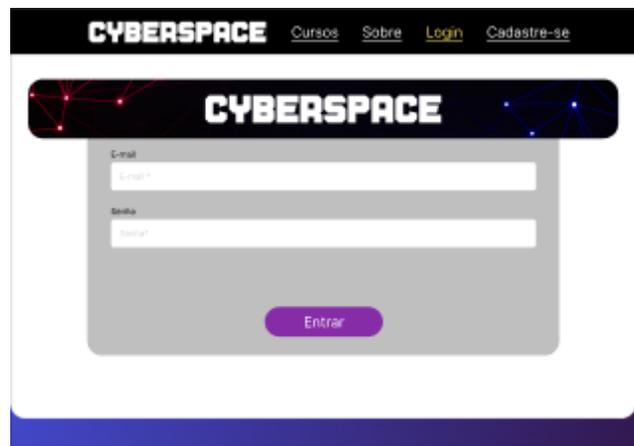
- Nome do site à esquerda.
- Menu de navegação à direita com opções como, "Cursos", "Sobre", "login" "cadastre".

2. Formulário de Login:

- Título centralizado: "Cyberspace"
- Campos para preenchimento:
- E-mail



- Senha
- Botão "Entrar" para submeter o formulário.



2ª Imagem: Area de Login

- Prototipação da Tela de Cadastro:

1. Header:

- Logo do site à esquerda.
- Link "Início" à direita para retorno à página inicial.

2. Formulário de Cadastro:

- Título centralizado: "Comece sua Jornada de Aprendizado!"
- Campos para preenchimento:
 - Nome
 - sobrenome
 - E-mail
 - Confirmar Email
 - Senha
 - Confirmação de Senha
- Botão "registrar" para submeter o formulário.



The image shows a registration form for the CYBERSPACE website. At the top, there is a navigation menu with links for 'Cursos', 'Sobre', 'Login', and 'Cadastre-se'. The form itself is titled 'CYBERSPACE' and contains the following fields: 'Nome' (Name), 'Sobrenome' (Surname), 'E-mail', 'Confirme seu e-mail' (Confirm email), 'Senha' (Password), and 'Confirme Senha' (Confirm Password). A purple 'Registrar' button is located at the bottom of the form.

3ºImagem: Area de Cadastro

- Prototipação da Página de Cursos:

1. Header:

- Logo do site à esquerda.
- Menu de navegação à direita com opções como, "Cursos", "Sobre", "login" "cadastre".
- Barra de pesquisa

2. Detalhes do Curso:

- Imagem representativa do curso.
- Título do curso em destaque.
- Botão veja mais.



4ªImagem: Página de Cursos

- Prototipação pesquisa

1. Header:

- Logo do site à esquerda.
- Menu de navegação à direita com opções como, "Cursos", "Sobre", "login" "cadastre".
- Barra de pesquisa

2. Barra de Pesquisa:

- Localizada no topo da página.
- Campo de texto para inserir palavras-chave.
- Botão "Pesquisar" para iniciar a busca.



5ªImagem: Procura de Cursos.

CONSIDERAÇÕES FINAIS

A escassez de profissionais qualificados em programação é um desafio significativo enfrentado pela indústria atualmente. A rápida evolução tecnológica e a demanda crescente por soluções digitais têm criado uma lacuna entre a oferta e a procura de profissionais com habilidades especializadas. Essa escassez de talentos pode afetar negativamente o desenvolvimento e o desempenho das organizações, dificultando a implementação de projetos de TI e limitando a capacidade de inovação. O uso da engenharia de software no desenvolvimento de um site voltado ao ensino de programação pode ajudar a garantir que o site seja de alta qualidade e atenda às necessidades dos usuários. Isso é importante, pois pode ajudar a aumentar a eficácia do ensino de programação e a atender à crescente demanda por profissionais qualificados nessa área. A engenharia de software foi uma ferramenta valiosa para o desenvolvimento do site a utilização de métodos e técnicas de engenharia de software pode nos ajudar a garantir que o site seja de alta qualidade, eficaz e atenda às necessidades dos usuários.

O desenvolvimento de sites educacionais voltados ao ensino de programação é uma área promissora com potencial para impactar positivamente a educação e a sociedade. O site



elaborado se mostra dinâmico e poderá crescer não só em conteúdo como em funcionalidades. Em uma perspectiva futura, pretendemos tornar o site ainda mais sofisticado, que utilizem tecnologias emergentes, como a realidade virtual e a inteligência artificial.

REFERÊNCIAS

- BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. 2. ed. Rio de Janeiro: Campus, 2006.
- BLATT, Lucas. **Mapeamento Sistemático sobre Metodologias Ferramentas de apoio para o Ensino de Programação de 2017**. Disponível em <https://periodicos.unesc.net/ojs/index.php/sulcomp/article/view/794/745> Acesso em: 26 nov. 2023.
- BRASCOM. **Demanda de Talentos em TIC e Estratégia Σ TCEM - Brasscom**. Disponível em: <https://brasscom.org.br/inteligencia/demanda-de-talentos-em-tic-e-estrategia-%cf%83-tcem> Acesso em: 12 jul. 2023.
- DYER, S., Martin, J. and Zulauf, J. (1995) “Motion Capture White Paper”, http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html, December. Acesso em: 12 jul. 2023.
- FRANÇA, R. S.; AMARAL, H. J. C. Mineração de dados na identificação de grupos de estudantes com dificuldades de aprendizagem no ensino de programação. **RENOTE-Revista Novas Tecnologias na Educação**, v. 11, n. 1, 2013. Knuth, D. E. (1984), *The TeXbook*, Addison Wesley, 15^a edition.
- ESTÊVÃO, GUILHERME. **Qualificação profissional em TI: como escolher?** IMASTERS 4 de novembro de 2013. Disponível em <https://imasters.com.br/carreira-dev/qualificacao-profissional-em-ti-como-escolher>. Acesso em 05 de junho de 2023.
- HOLTON, M. and Alexander, S. Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials. **Computer Graphics: Developments in Virtual Environments**, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd., 1995, p. 449-460.
- IEEE Std. 830, IEEE. **Guide to Software Requirements Specification**, The Institute of Electrical and Electronics Engineers, New York, EUA, (1984).
- KOURI, Márcia G. **Definição de requisitos para um sistema de monitoramento de veículos no transporte rodoviário de cargas**. São Paulo, 2007. 165f... Dissertação (Mestrado em Engenharia Elétrica) – Escola Politécnica, Universidade de São Paulo.
- NETO, Oscar, 2021 **Percepções do ensino de programação em cursos de tecnologia da informação: o que pensam discentes e docentes?** Novembro de 2013. Disponível em <http://conjecturas.org/index.php/edicoes/article/view/107/81> Acesso em: 26 nov. 2023.



PREECE, Jennifer. ROGERS, Yvonne. SHARP, Helen. **Design de Interação: além da interação homem-computador**. Porto Alegre: Bookman, 2005.

PRESSMAN, R.S. **Engenharia de Software**. 5ª ed. Rio de Janeiro: McGraw-Hill, 2002, 843p.

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução: Selma Shin Shimizu Melnikoff, Reginaldo Arakaki, Edilson de Andrade Barbosa. 8. ed. São Paulo: Person Addison-Wesley, 2007.

SOMMERVILLE, Ian. **Engenharia de Software**. 6ª ed. São Paulo: Addison Wesley, 2003, 592p.

SOMMERVILLE, Ian. **Engenharia de Software**. 9 ed. São Paulo: Pearson Prentice Hall, 2011.

SOUZA, D. M.; *et al.* **Problemas e dificuldades no ensino de programação: Um mapeamento sistemático**. Revista Brasileira de Informática na Educação, v. 24, n. 1, p. 39, 2016.

THAYER, Richard; DORFMAN, Merlin. **System and Software Requirements Engineering - Second Edition**. Los Alamitos: IEEE Computer Society Press Tutorial, 2000. 528p .

ZAVE, Pamela. **Classification of Research Efforts in Requirements Engineering**. ACM Computing Surveys. v. 29, n. 4. Dev. 1997. p. 315-321.

AUTORES

Ismael M. A. de Souza

Minicurrículo: Bacharel em Ciências da Computação – Universidade Cruzeiro do Sul – UNICSUL.

Rodrigo dos Santos

Minicurrículo: Bacharel em Ciências da Computação – Universidade Cruzeiro do Sul – UNICSUL.

Thomas N. Henrique



Minicurriculo: Bacharel em Ciências da Computação – Universidade Cruzeiro do Sul – UNICSUL.

Juliano Schimiguel

link do Orcid: <https://orcid.org/0000-0001-8552-7984>

Possui Doutorado em Ciência da Computação pela Universidade Estadual de Campinas - UNICAMP (2006), Mestrado em Ciência da Computação pela Universidade Estadual de Campinas - UNICAMP (2002) e Graduação de Bacharelado em Informática pela Universidade Estadual de Ponta Grossa (1999). Atualmente é Coordenador do Mestrado Profissional (MP) em Ensino de Ciências e Matemática, Professor Permanente do Programa de Doutorado / Mestrado em Ensino de Ciências e Matemática, Coordenador de Estágio Supervisionado e de TGI - Trabalho de Graduação Interdisciplinar, do Curso de Bach. em Ciência da Computação, da Universidade Cruzeiro do Sul (São Paulo, SP); Professor do Centro Universitário Anchieta - UNIANCHIETA (Jundiaí, SP); Associado Profissional no Pecege da USP/Esalq (Piracicaba/SP), e Professor Formador na Pós-Graduação em Inovação na Educação Mediada Por Tecnologias na UFABC/MEC/UAB. Tem experiência na área de Ciência da Computação, com ênfase em Interação Humano-Computador (IHC) e Engenharia de Software, atuando principalmente nos seguintes temas: desenvolvimento web, design e avaliação de interfaces, sistemas de informação geográfica, geoprocessamento, análise de sistemas, UML, UP, ensino-aprendizagem de tecnologias e matemática, conteúdos digitais interativos, objetos de aprendizagem, ambientes virtuais e colaborativos, jogos para o ensino, etc.

Carlos Adriano Martins

link do Orcid: <https://orcid.org/0000-0003-0688-3441>

Pós-Doutorando em Educação e Professor Associado da Unilogos International. Pós-Doutorado em Ensino de Ciências pela Universidade Cruzeiro do Sul. Doutor em Ensino. Mestre em Educação. Especialista em diversas áreas do conhecimento. Graduado em Pedagogia, Educação Especial e Ciências Biológicas. Qualificação e atuação nas áreas de formação docente, TDICs, educação ambiental, coordenação de cursos (graduação, extensão e especialização), consultoria educacional. Educação a Distância: gestão, elaboração de cursos, produção MDM, tutoria, AVAs.



PAIDÉI@
ISSN - 1982-6109

REVISTA CIENTÍFICA DE EDUCAÇÃO A DISTÂNCIA



Artigo Recebido em: 18/05/2024

Aceito para Publicação em: 20/06/2025