

Avaliação Diagnóstica

Sistema de avaliação para desenvolvimento de sistemas para internet

Sumário

1. Introdução e contextualização.
2. Justificativa
3. Capacidades desenvolvidas.
4. Instrumento avaliativo sugerido.
5. Protótipo.

Introdução e contextualização

O instrumento desenvolvido trata-se de uma avaliação diagnóstica, cuja finalidade é identificar o conhecimento prévio da turma em tópicos considerados fundamentais para o desenvolvimento das competências previstas no Curso Técnico em Desenvolvimento de Sistemas (CT-DESI). A aplicação ocorre em um ambiente web com interface intuitiva, semelhante ao *Google Forms/Classroom*, e com design responsivo, adaptado tanto para dispositivos móveis quanto para desktop. Ao acessar o sistema, o estudante se depara com um formulário de entrada contendo campos obrigatórios para nome completo, turma e seleção de experiência prévia em desenvolvimento de sistemas, o que possibilita uma contextualização inicial do perfil da turma. Na sequência, o aluno inicia a avaliação diagnóstica, composta por diferentes tipos de questões. São abordados tanto conteúdos teóricos — com o objetivo de aferir a compreensão de conceitos essenciais — quanto questões práticas com editor de código (pseudocódigo), como exemplificado na "Questão A2 - Algoritmo", que permite avaliar diretamente as habilidades de lógica e programação dos estudantes. Para os docentes, a ferramenta oferece ainda um *Dashboard* do Professor, que disponibiliza um relatório detalhado com estatísticas da turma, gráficos de barras organizados por competência e recomendações pedagógicas automáticas. Esses recursos apoiam o planejamento didático e auxiliam na identificação de áreas que demandam reforço no processo de ensino-aprendizagem.

Justificativa

A ferramenta foi escolhida e desenvolvida com características que a tornam ideal para a avaliação diagnóstica no contexto do curso técnico em desenvolvimento de sistemas (CT-DESI). Ela permite a coleta de dados do aluno através de campos obrigatórios e um *dropdown* de experiência prévia. Sua capacidade de incluir questões práticas com editor de código (pseudocódigo) e funcionalidade de teste de código, assegura uma avaliação das habilidades lógicas e de programação. Para o professor, o "Dashboard do Professor" é uma justificativa chave, oferecendo um relatório completo com estatísticas da turma, gráficos de barras por competência e recomendações pedagógicas automáticas, facilitando o planejamento didático e o reforço em áreas específicas. A experiência do estudante é priorizada com uma interface "limpa similar ao Google Forms", salvamento automático das respostas, validação em tempo real e design responsivo, garantindo uma avaliação fluida e acessível.

Capacidades desenvolvidas

O instrumento avaliativo foi projetado para identificar o conhecimento prévio dos alunos, considerado um requisito fundamental para o sucesso no desenvolvimento das competências no Curso Técnico em Desenvolvimento de Sistemas (CT-DESI).

Conforme o critério de avaliação da equipe, o instrumento foca no levantamento das seguintes oito capacidades principais, que são cruciais para o itinerário formativo do curso:

- ☐ C1 - Utilizar aplicações e sistemas operacionais no desenvolvimento de documentação de sistemas web.
- ☐ C2 - Compreender fundamentos de eletroeletrônica aplicada no desenvolvimento de sistemas.
- ☐ C3 - Aplicar lógica de programação na resolução de problemas computacionais. Esta capacidade é avaliada diretamente por meio de questões práticas com editor de código, como a "Questão A2 - Algoritmo", que exige o desenvolvimento de um algoritmo para resolver um problema específico e testar o código.
- ☐ C4 - Utilizar técnicas de modelagem e linguagem na manipulação de banco de dados. Esta capacidade está alinhada com as recomendações pedagógicas de "revisão intensiva de SQL básico e consultas complexas".

- ☐ C5 - Selecionar recursos e linguagem de programação para integração IoT.
- ☐ C6 - Estruturar aplicativos e sistemas por meio de técnicas de modelagem. Esta capacidade se relaciona com as avaliações sobre Arquitetura de Software e Metodologias Ágeis.
- ☐ C7 - Desenvolver aplicações e sistemas por meio de linguagem de programação. Esta capacidade abrange o Web Frontend e o desenvolvimento de exercícios práticos de JavaScript e manipulação do DOM, conforme indicado nas recomendações.
- ☐ C8 - Selecionar procedimentos de teste que assegurem a aderência aos requisitos.


Além dessas capacidades específicas, o sistema de avaliação também coleta informações sobre a "Experiência Prévia em Desenvolvimento Web" dos alunos, variando de "Nenhuma experiência" a "Experiência profissional", o que é um dado relevante para contextualizar o perfil do estudante e suas necessidades de aprendizado. As "Recomendações Pedagógicas" automáticas geradas pelo Dashboard do Professor detalham ainda mais as habilidades específicas que necessitam de reforço, como conceitos fundamentais de CSS e layout responsivo e introdução a versionamento e Git.

Instrumento avaliativo sugerido

Abaixo, apresentamos o instrumento avaliativo sugerido, desenhado para diagnosticar o conhecimento prévio e as habilidades fundamentais dos estudantes, componentes cruciais para o sucesso no itinerário formativo do Curso DESI. A Seção A, focada em Lógica de Programação e Algoritmos, tem como objetivo levantar as capacidades dos estudantes nesta área, considerada um requisito essencial para a construção das competências necessárias nos módulos subsequentes. As questões formuladas, como o desafio de desenvolver um algoritmo para calcular médias e classificar resultados, visam avaliar a compreensão e a aplicação prática dos conceitos de pensamento computacional e resolução de problemas.

Seção A: Lógica de programação e algoritmos

1. Desenvolva um algoritmo que calcule a média aritmética de três números e determine se o resultado é maior, menor ou igual a 7.0. O algoritmo deve exibir a média calculada e uma mensagem indicando a classificação. Exemplo de Entrada: 8.0, 6.5, 7.5. Exemplo de Saída: Média: 7.33 - Resultado: Maior que 7.0.



2. Qual das seguintes estruturas de controle de fluxo é a mais adequada para repetir um bloco de código um número conhecido de vezes?

- a) WHILE
- b) IF-ELSE
- c) FOR
- d) DO-WHILE

3. Se um algoritmo precisa tomar uma decisão baseada em uma condição, qual tipo de estrutura de controle de fluxo deve ser utilizada?

- a) Estrutura de Repetição
- b) Estrutura Condicional
- c) Estrutura de Iteração
- d) Estrutura de Laço

Seção B: Web Frontend (HTML, CSS, JavaScript, DOM)

4. O que é "Design Responsivo" no desenvolvimento web?

- a) É a criação de um site que só funciona em dispositivos móveis.
- b) É a técnica de usar JavaScript para animar elementos da página.
- c) É a abordagem que garante que um site se adapte e tenha boa aparência em diferentes tamanhos de tela e dispositivos.
- d) É o uso exclusivo de tabelas para organizar o layout de uma página.

5. Para que serve a propriedade CSS display: flex;?

- a) Para adicionar uma sombra aos elementos.
- b) Para alinhar texto dentro de um parágrafo.
- c) Para criar um contexto de layout flexível que permite alinhar e distribuir itens em um contêiner.

d) Para mudar a cor de fundo de um elemento.

6. Qual trecho de código JavaScript seleciona um elemento HTML com o id="titulo" e altera seu conteúdo de texto para "Novo Título"?

a) `document.getElementById("titulo").text = "Novo Título";`

b) `document.query("titulo").content = "Novo Título";`

c) `document.getElementById("titulo").textContent = "Novo Título";`

d) `document.selectId("titulo").innerText = "Novo Título";`

7. Qual a principal diferença e a recomendação atual sobre o uso de `let` e `var` para declaração de variáveis em JavaScript?

a) `let` é para constantes, `var` para variáveis mutáveis; `var` é o mais recomendado.

b) `var` tem escopo de bloco, `let` tem escopo de função; `var` é mais seguro.

c) `let` permite redeclaração, `var` não; `let` é o mais recomendado.

d) `let` tem escopo de bloco, `var` tem escopo de função; `let` é geralmente mais recomendado para evitar problemas de escopo.

8. O que representa o DOM (Document Object Model) no contexto do desenvolvimento web com JavaScript?

a) Um protocolo de comunicação entre o cliente e o servidor.


b) Uma interface de programação para documentos HTML e XML, permitindo que scripts acessem e manipulem o conteúdo da página.

c) Uma linguagem de estilo para estilizar elementos HTML.

d) Um método de compressão de imagens para otimização de páginas.

Seção C: Banco de dados (SQL)

9. Dada uma tabela `Produtos` com colunas `ID`, `NomeProduto` e `Preco`, qual consulta SQL selecionaria o `NomeProduto` e o `Preco` de todos os produtos com `Preco` superior a 50.00?

- 
- a) `SELECT * FROM Produtos WHERE Preço > 50.00;`
 - b) `SELECT NomeProduto, Preço FROM Produtos HAVING Preço > 50.00;`
 - c) `SELECT NomeProduto, Preço FROM Produtos ORDER BY Preço > 50.00;`
 - d) `SELECT NomeProduto, Preço FROM Produtos WHERE Preço > 50.00;`

10. Qual a principal característica de um LEFT JOIN em SQL?

- a) Retorna apenas as linhas que possuem correspondência em ambas as tabelas.
- b) Retorna todas as linhas da tabela da direita e as correspondências da tabela da esquerda.
- c) Retorna todas as linhas de ambas as tabelas, combinando-as quando possível.
- d) Retorna todas as linhas da tabela da esquerda e as linhas correspondentes da tabela da direita. Se não houver correspondência, retorna NULL para as colunas da tabela da direita.


11. Qual a finalidade da cláusula GROUP BY em uma consulta SQL?

- a) Ordenar os resultados da consulta de forma ascendente ou descendente.
- b) Filtrar linhas com base em uma condição específica antes de aplicar agregações.
- c) Agrupar linhas que têm os mesmos valores em colunas especificadas em um conjunto de linhas de resumo, frequentemente utilizada com funções de agregação.
- d) Inserir novos registros em uma tabela do banco de dados.

Seção D: Metodologias ágeis e arquitetura de software

12. Em metodologias ágeis como o Scrum, o que representa uma "Sprint"?

- a) Uma reunião diária de alinhamento da equipe.
- b) Um período de tempo fixo e curto (geralmente de 1 a 4 semanas) durante o qual uma equipe de desenvolvimento trabalha para criar um incremento de produto funcional.
- c) O documento que lista todos os requisitos do projeto.
- d) A fase final de testes antes do lançamento do produto.



13. Qual dos seguintes artefatos Scrum é uma lista ordenada de tudo o que pode ser necessário no produto e é mantido pelo Product Owner?

- a) Sprint Backlog
- b) Incremento
- c) Product Backlog
- d) Burndown Chart

14. Qual a principal característica de uma arquitetura de microsserviços em comparação com uma arquitetura monolítica?


- a) É um único aplicativo com todos os seus componentes interligados em uma única base de código.
- b) Consiste em uma coleção de pequenos serviços independentes, cada um rodando em seu próprio processo e se comunicando através de interfaces bem definidas.
- c) Permite que todos os desenvolvedores trabalhem na mesma parte do código simultaneamente.
- d) Requer um único banco de dados centralizado para todos os módulos.

15. No processo de design e modelagem de software, qual o principal objetivo de um Diagrama de Classes UML?

- a) Descrever o fluxo de trabalho de um usuário no sistema.
- b) Representar a estrutura estática do sistema, mostrando classes, seus atributos, operações e os relacionamentos entre elas.
- c) Ilustrar a sequência de mensagens entre objetos em um cenário específico.
- d) Detalhar os componentes de hardware necessários para o sistema.

Seção E: Ferramentas, testes e outras capacidades

16. Qual par de ferramentas de linha de comando (terminal) é mais comumente usado por desenvolvedores para gerenciar pacotes e controlar versões de código?

- 
- a) ping e ipconfig
 - b) cd e ls
 - c) npm (ou yarn) e git
 - d) mkdir e rm

17. Qual a principal diferença entre os comandos git pull e git fetch?


- a) git pull apenas baixa as alterações, git fetch baixa e mescla imediatamente.
- b) git fetch baixa as alterações do repositório remoto para o repositório local sem mesclá-las no branch de trabalho, enquanto git pull faz o fetch e, em seguida, um merge automático.
- c) git pull envia alterações para o repositório remoto, git fetch as recebe.
- d) Ambos os comandos são idênticos e fazem a mesma coisa.

18. Qual a principal vantagem de criar e trabalhar com "branches" (ramificações) em um repositório Git?

- a) Permite que vários desenvolvedores trabalhem na mesma funcionalidade sem conflitos.
- b) Permite que desenvolvedores trabalhem em diferentes funcionalidades ou correções de bug de forma isolada, sem afetar a linha principal de desenvolvimento (main/master) até que estejam prontas para serem integradas
- c) Ajuda a otimizar o tamanho do repositório, removendo arquivos antigos.
- d) É um backup automático do código.

19. O que são "Testes de Unidade" (Unit Tests) no desenvolvimento de software?

- a) Testes que verificam se o sistema inteiro funciona corretamente do ponto de vista do usuário.
- b) Testes realizados por um grupo seleto de usuários antes do lançamento público.
- c) Testes que validam componentes individuais ou as menores partes testáveis de uma aplicação de forma isolada, como funções ou métodos.
- d) Testes que garantem a segurança do sistema contra ataques externos.



20. Qual dos seguintes itens é um componente essencial na maioria dos sistemas de Internet das Coisas (IoT) que coleta dados do ambiente?

- a) Atuador
- b) Sensor
- c) Servidor web
- d) Banco de dados relacional

Protótipo

O protótipo da aplicação apresenta, de forma visual, as principais interfaces e funcionalidades da ferramenta de avaliação diagnóstica. A Interface Principal (ou Tela Inicial) exibe campos para preenchimento do nome completo, turma (exemplo: DESI-2024.1) e um menu suspenso (dropdown) para seleção da experiência prévia em Desenvolvimento Web. Essa tela também fornece informações importantes, como o tempo estimado para realização da avaliação (90 minutos), o número de seções (5), a pontuação total (100 pontos), além de destacar o recurso de salvamento automático e apresentar instruções relevantes para os estudantes. A Figura 1 ilustra essa interface.

Figura 1 – Tela inicial da aplicação

A captura de tela mostra a interface inicial da aplicação 'Avaliação Diagnóstica DESI'. No topo, há um cabeçalho com o título e o subtítulo. Abaixo, há uma barra de navegação com três opções: 'Interface Principal' (destacada), 'Questão Prática' e 'Dashboard Professor'. O corpo da tela contém um formulário para o usuário se identificar, com campos para 'Nome Completo' e 'Turma'. Abaixo disso, há um menu suspenso para 'Experiência Prévia em Desenvolvimento Web'. Uma barra de informações exibe detalhes como 'Tempo estimado: 90 minutos', 'Seções: 5 (A, B, C, D, E)', 'Total de pontos: 100' e 'Salvamento: Automático'. Abaixo, há uma seção de 'Instruções Importantes' com uma lista de pontos. No rodapé, há dois botões: 'Iniciar Avaliação' e 'Salvar Rascunho'.

Avaliação Diagnóstica DESI
Sistema de Avaliação para Desenvolvimento de Sistemas para Internet

Interface Principal Questão Prática Dashboard Professor

Avaliação Diagnóstica DESI - Módulo Web

Nome Completo * Turma *

Digite seu nome completo Ex: DESI-2024.1

Experiência Prévia em Desenvolvimento Web *

Selecione sua experiência

⌚ Tempo estimado: 90 minutos 📄 Seções: 5 (A, B, C, D, E) 📊 Total de pontos: 100 💾 Salvamento: Automático

📌 Instruções Importantes

- Leia atentamente cada questão antes de responder
- Você pode salvar rascunhos e retomar depois
- Questões práticas incluem editor de código
- Teste seu código antes de submeter
- Certifique-se de completar todas as seções

Iniciar Avaliação Salvar Rascunho

Fonte: Autores, 2025.

A Interface da Questão Prática com Editor de Código oferece um editor de código interativo, juntamente com os botões “Testar Código” e “Salvar Resposta”. Após a execução do teste, o sistema retorna uma mensagem automatizada, como “Teste executado com sucesso”, exibindo a saída esperada. A Figura 2 apresenta essa tela.

Figura 2 – Tela de questão prática com editor de código

The screenshot displays the 'Avaliação Diagnóstica DESI' (Diagnostic Assessment DESI) interface. At the top, a green header bar contains the title and subtitle: 'Sistema de Avaliação para Desenvolvimento de Sistemas para Internet'. Below the header, a navigation bar includes three buttons: 'Interface Principal', 'Questão Prática' (highlighted in blue), and 'Dashboard Professor'. The main content area is titled 'Questão A2 - Algoritmo' with a sub-header '10 pontos'. It contains a problem statement in Portuguese, an example input/output, and a code editor. The code editor is titled 'Editor de Código (Pseudocódigo/Algoritmo):' and shows a pseudocode for calculating the average of three numbers. At the bottom of the code editor are two buttons: 'Testar Código' and 'Salvar Resposta'. Navigation buttons at the very bottom are '← Questão Anterior' and 'Próxima Questão →'.

Avaliação Diagnóstica DESI
Sistema de Avaliação para Desenvolvimento de Sistemas para Internet

Interface Principal **Questão Prática** Dashboard Professor

Questão A2 - Algoritmo
10 pontos

Enunciado: Desenvolva um algoritmo que calcule a média aritmética de três números e determine se o resultado é maior, menor ou igual a 7.0. O algoritmo deve exibir a média calculada e uma mensagem indicando a classificação.

Exemplo:
Entrada: 8.0, 6.5, 7.5
Saída: Média: 7.33 - Resultado: Maior que 7.0

Editor de Código (Pseudocódigo/Algoritmo):

```
algoritmo.txt - Pseudocódigo

INÍCIO
// Declare as variáveis
DECLARE num1, num2, num3, media: REAL

// Entrada dos dados
ESCREVA "Digite o primeiro número: "
LEIA num1
ESCREVA "Digite o segundo número: "
LEIA num2
```

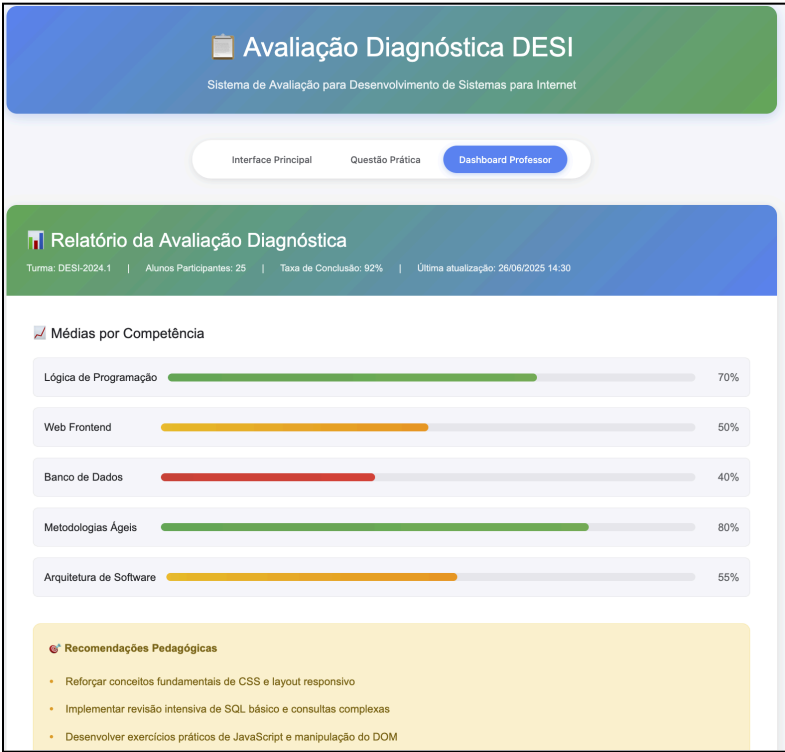
Testar Código Salvar Resposta

← Questão Anterior Próxima Questão →

Fonte: Autores, 2025.

O Dashboard do Professor exibe um relatório consolidado da avaliação diagnóstica, com informações detalhadas da turma, gráficos de barras por competência (ex.: Lógica de Programação, *Web Frontend*), recomendações pedagógicas geradas automaticamente, estatísticas gerais (como média e desvio padrão) e indicadores de tempo de execução, por aluno e por questão. Visualmente, a aplicação adota um design responsivo, com animações suaves, paleta de cores inspirada nos produtos Google, interface intuitiva (semelhante ao Google Forms/Classroom), tipografia clara (Google Sans), ícones em estilo emoji, barras de progresso animadas e botões com efeitos visuais interativos. A Figura 3 apresenta o dashboard.

Figura 3 – Dashboard do professor



Fonte: Autores, 2025.