

14º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2023

MetodoLogic: um jogo de metodologia científica a partir do desenvolvimento web

GABRIELA ANDRADE BUENO DE LIMA¹, ROSANA FERRARETO LOURENÇO RODRIGUES²,

¹Graduanda em Ciência da Computação, Bolsista PIBIFSP, IFSP, Câmpus São João da Boa Vista, lima.gabriela@aluno.ifsp.edu.br.

²Professora Doutora em Linguística e Língua Portuguesa, docente do IFSP, Câmpus São João da Boa Vista, rosanaferrareto@ifsp.edu.br

Área de conhecimento (Tabela CNPq): 7.08.04.00-1 Ensino-Aprendizagem

RESUMO: O presente trabalho consiste no desenvolvimento de um sistema web gamificado para o ensino da metodologia científica. Toda área do conhecimento requer uma explicação clara e coesa e a caracterização e descrição da pesquisa deve envolver abordagens, métodos e técnicas que tornem a interpretação e compreensão do texto acessíveis a qualquer leitor. Assim, a escrita científica e a seleção das metodologias que mais se adequem ao próprio estudo é uma tarefa complexa. Nesse contexto, a aplicação MetodoLogic vem como suporte teórico e prático a esse processo, de modo a fornecer, através de um jogo dinâmico e educativo, meios para o pesquisador estudar e praticar o uso da metodologia. A gamificação do ensino torna o aprendizado ágil e flexível, uma vez que a jornada é dividida em etapas claras e lógicas que se adequem ao nível de conhecimento do jogador. Foram realizados estudos de gamificações e de suas aplicabilidades, levantamento de requisitos do sistema, modelagem e diagramação da plataforma, bem como de seus casos de uso e, por fim, o desenvolvimento do protótipo utilizando uma metodologia ágil. É previsto, portanto, o desenvolvimento de um sistema para web operacional e eficiente que integre as tecnologias Angular e Spring Boot a fim de construir lógicas de games e aplicar recursos a ele de modo a tornar o aprendizado de metodologia científico mais lúdico e educativo.

PALAVRAS-CHAVE: Metodologia Científica; Gamificação; Aplicação Web; Angular.

MetodoLogic: a scientific methodology game based on web development

ABSTRACT: This project consists of the development of a gamified web system for teaching scientific methodology. Every area of study requires a clear and cohesive explanation and the characterization and description of the research must involve approaches, methods and techniques that make the interpretation and understanding of the text accessible to any reader. Thus, scientific writing and methodologies selection that best suit the study is a complex task. In this context, the MetodoLogic application comes as theoretical and practical support to that process, in order to provide, through a dynamic and educational game, means for the researcher to study and practice the use of the methodology. The gamification makes an agile and flexible learning, since the journey is divided into clear and logical steps that suit the player's level of knowledge. Studies of gamification and their applicability, survey of system requirements, modeling and layout of the platform, as well as its use cases and, finally, the development of the prototype using an agile methodology were needed. It is planned, therefore, the development of an operational and efficient web system that integrates Angular and Spring Boot technologies in order to build game logic and apply resources to it so that we can make the learning of scientific methodology more playful and educational.

KEYWORDS: Scientific methodology; Gamification; Web Application; Angular.

INTRODUÇÃO

Grande parte dos pesquisadores, ao iniciar o próprio estudo, deparam-se com dificuldades quanto à seleção dos métodos e abordagens a serem utilizados para descrever a pesquisa. A escolha da metodologia adequada requer não só tempo, como também estudo para identificar todas as opções e, então, poder definir qual abordagem mais se adequa à própria pesquisa, tornando a tarefa de escolher a metodologia um processo complexo.

Assim, o presente estudo trata de desenvolver uma aplicação de jogo online para o ensino de metodologia científica: o *Metodologic*. Esta busca tornar o ensino mais dinâmico e ágil, além de

acessível, uma vez que sistemas hospedados na internet permitem que todos entrem na plataforma desde que tenham uma conexão *Wi-Fi*.

Para o desenvolvimento do protótipo do jogo, são utilizados frameworks de desenvolvimento web, de uma lógica de jogo (pontuação, fases, feedbacks, recompensas), que requerem um complexo algoritmo.

Neste artigo, apresentam-se os conceitos relativos ao referencial teórico, tais como métodos, técnicas e abordagens de escrita científica e frameworks de desenvolvimento como Angular e Spring Boot. Será abordada ainda a metodologia descrevendo-se como se deram as pesquisas bibliográfica e a experimental utilizadas para o desenvolvimento da aplicação. Por fim serão apresentados os resultados parciais, a categorização e gamificação do conteúdo referente à metodologia científica e o desenvolvimento da aplicação, desde o levantamento de requisitos e diagramação, até o design da interface e a construção do banco de dados e dos códigos necessários.

MATERIAL E MÉTODOS

Ao considerar a construção de qualquer sistema, as etapas iniciais consistem em identificar os desafios, as soluções e as interações do software para com o problema proposto. O desenvolvimento deste projeto, então, estará dividido em três segmentos principais, sendo eles: projeto, estruturação e implementação, cada um incluindo também subetapas, que estão representadas nas cores azul, verde e roxo, respectivamente, na Figura 1.



FIGURA 1. Fluxograma etapas de desenvolvimento. Fonte: Elaboração da autora

Referente ao levantamento de requisitos, foi necessário identificar tanto os problemas que o sistema se propõe a resolver, quanto as exigências e as condições das soluções. São, então, detectados os atores que vão operar na aplicação, especificado os casos de suas interações, para, por fim, levantar os componentes necessários para o desenvolvimento.

Ao considerar os requisitos e as funcionalidades do sistema, foi elaborado o Documento de Casos de Uso que se compromete a responder às questões relativas às utilidades da aplicação a partir do diagrama desenhado. Nessa etapa, todas as interações com o sistema são detalhadas.

Já a etapa de Modelagem do Protótipo é responsável por construir a estrutura do projeto. Para isso, foi elaborado, utilizando a plataforma BrModelo, o Diagrama de Entidade Relacional que projeta as entidades, seus atributos e seus relacionamentos.

A partir da estrutura mapeada, é possível fazer o Desenho da Interface, a partir do qual considera se por quais telas o usuário vai acessar cada um dos casos de uso levantados. Projetar as telas envolve considerar as páginas principais, bem como a de erros e as de carregamento. Além disso, ao arquitetar a interface, é necessário também definir os caminhos de acesso de cada página. Foi, então, utilizado o Figma para a construção das telas e de seus componentes.

A etapa de Construção de Conteúdo considera o desenvolvimento do código do front-end e para isso foi utilizado o framework Angular que, com uma extensa biblioteca e o CLI, facilita a escrita do código. A partir das interfaces desenhadas, são elaborados componentes que abstraem o conteúdo e as

¹ <https://www.figma.com/>

rotas de acesso. Com toda a base do projeto desenhado, é iniciado o desenvolvimento a partir da Exploração das APIs que abrangem a construção do back-end com framework Spring Boot. Todos os componentes identificados no MER são criados em formato de código.

Dessa forma, é possível conectar o banco de dados construído com o código. Por fim, é necessário testar o protótipo desenvolvido.

RESULTADOS E DISCUSSÃO

Os resultados apresentados estão compostos seguindo as mesmas etapas apresentadas na metodologia. O levantamento de requisitos abrange discussões do escopo e das expectativas em relação à aplicação, a fim de designar quais funcionalidades o Metodologic propõe-se a resolver. Nesse contexto, foi desenhado o seguinte ambiente:

- Plataforma online;
- Jogo modo jornada individual com níveis de dificuldade crescendo conforme desenvolvimento do conhecimento sobre metodologias do usuário;
- Sistema de pontuação e conquistas que permitem compartilhamento;
- Três tipos de perfil aluno, professor e administrador de informações;
- Jogo modo avaliativo atribuído pelo professor com a dificuldade e questões selecionadas;
- Sistema de organização de turmas e notas para o perfil professor;
- Análise estatística sobre as questões respondidas, acertos e erros por temas e por níveis.

Uma vez determinados o contexto e as finalidades, foram especificadas as funções e aplicações do sistema que respondam às ações e interações do usuário com o aplicativo para o funcionamento esperado do jogo. O diagrama elaborado foi delineado de acordo com o padrão UML. Para essa aplicação, foram identificados os atores, a saber:

- Usuário: responsável por jogar, realizar a jornada e interagir com a plataforma, conquistando pontos e vitórias e as compartilhando;
- Alunos: responsável por realizar as atividades e tarefas atribuídas;
- Professor: responsável por criar e gerenciar turmas e monitorar e atribuir tarefas, propondo atividades aos alunos;
- Administrador da informação: responsável pela análise da influência da aplicação.

A partir do diagrama é elaborado o Modelo de Casos de Uso, um documento que descreve e especifica o diagrama (Figura 2). Nesse documento, são designados os papéis dos atores e os cenários para cada um dos casos de uso levantados.

Para cada um dos casos de uso do diagrama, é elaborado um quadro (QUADRO 1), descrevendo todas as informações necessárias para o desenvolvimento: função do caso de uso, ator envolvido nessa funcionalidade, atributos dos dados de entrada, o cenário principal esperado e os cenários alternativos.

QUADRO 1. Exemplo Cenário Caso de Uso - UC11: Editar Perfil

| Breve descrição do Caso de Uso: | Editar o perfil. | |
|---|---|------------------------------------|
| Ator (s) Envolvido(s) | Usuário, Professor e Aluno | |
| Atributos Manipulados e seus Domínios de Dados: | Atributo | Domínio |
| | | Nome Usuario |
| | Email | Varchar(100) (exemplo@exemplo.com) |
| | Senha | Varchar(20) |
| | Instituição de Ensino | Varchar(50) |
| | Data de nascimento | Date |
| | Curso | Varchar(200) |
| Cenário Principal: | Ao clicar no botão “Editar”, os campos mostrarão, já com as informações originais preenchidas, permitindo assim que o usuário faça a edição. No final da página terão dois botões, “Salvar” e “Cancelar”, onde ao clicar no botão de “Salvar” a publicação do evento será atualizada e a tela será redirecionada para a página de Perfil. | |
| Cenário Alternativos/Exceção: | <p>Cenário Alternativo/Exceção #01: Caso o usuário deixe um ou mais campos de cadastramento de visitante como vazios e tentar clicar no botão “Cadastrar Informações” deverá aparecer uma mensagem informando = “Por favor preencha todos os campos!”;</p> <p>Cenário Alternativo/Exceção #0N: Descrição de todos os possíveis erros ao editar perfil</p> | |

Fonte: Elaboração da autora.

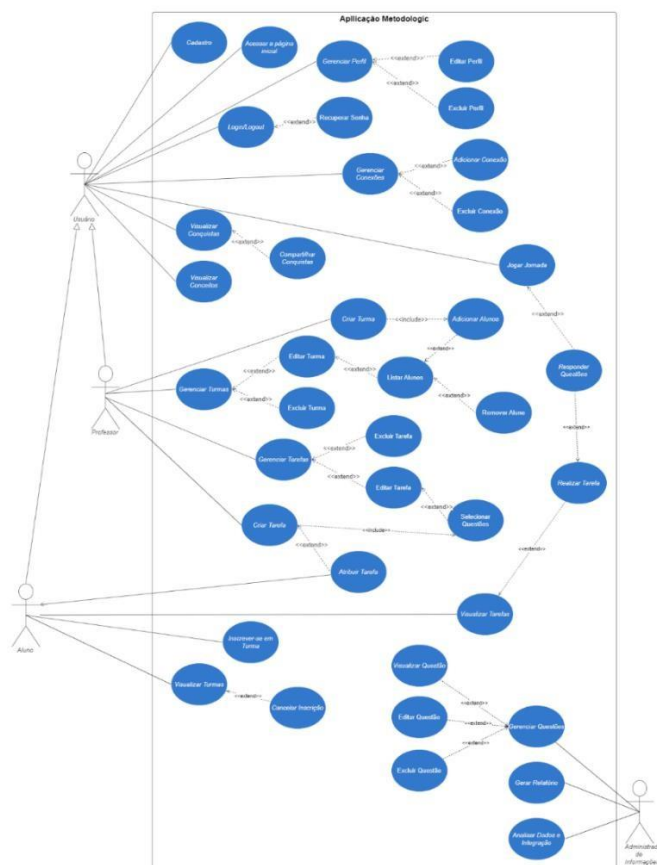


FIGURA 2. Diagrama de Casos de Uso do Metodologic. Fonte: Elaboração da autora.

Já para a estruturação foi desenvolvido o Diagrama de Entidade Relacionamento (Figura 3), que representa a organização dos dados, cada entidade e seus atributos, que são desenhados e como esses serão integrados.

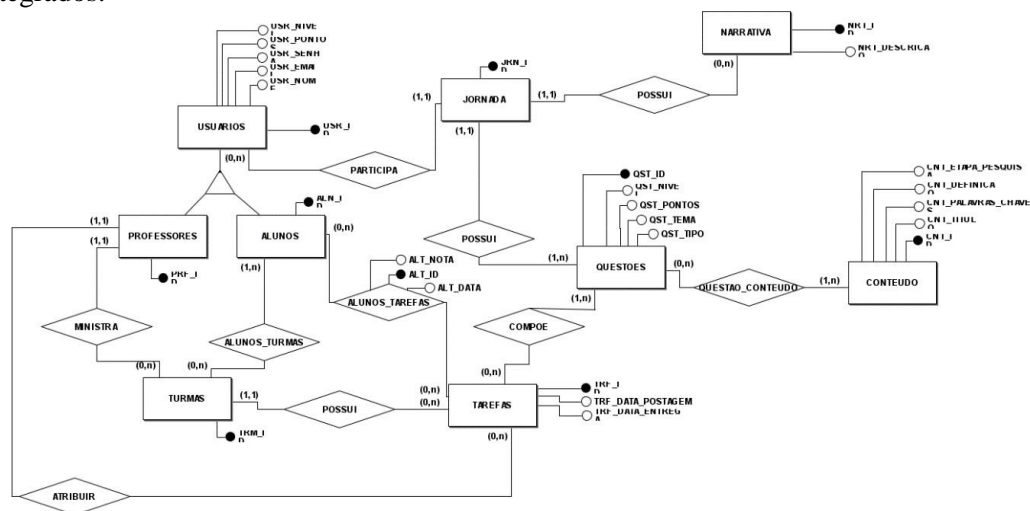


FIGURA 3. Diagrama Entidade Relacionamento da Aplicação. Fonte: Elaboração da autora.

Quanto à interface gráfica da aplicação, foram desenvolvidas telas que se adequem aos casos de usos e, conseqüentemente, às funcionalidades do sistema. As principais telas estão representadas nas Figuras 4 e 5.

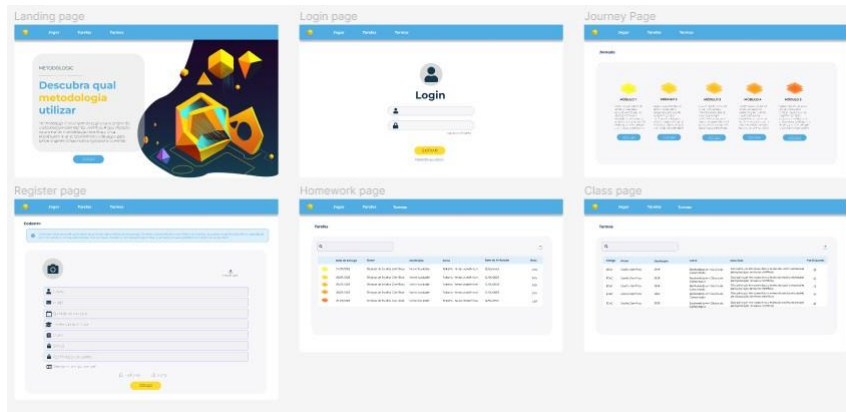


FIGURA 4. Design das páginas principais. Fonte: Elaboração da autora com a ferramenta X.

A construção do conteúdo do front-end se deu pelo framework Angular através da seguinte organização dos arquivos, com dois focos principais: as páginas na pasta home que são referentes às páginas de login, cadastro e a página de entrada, ou seja, os arquivos que não é necessário credenciais para acessar; já a pasta dashboard é composta pelos arquivos que representam as funcionalidades específicas do sistema, como a página de turmas, tarefas, jornada e perfil.

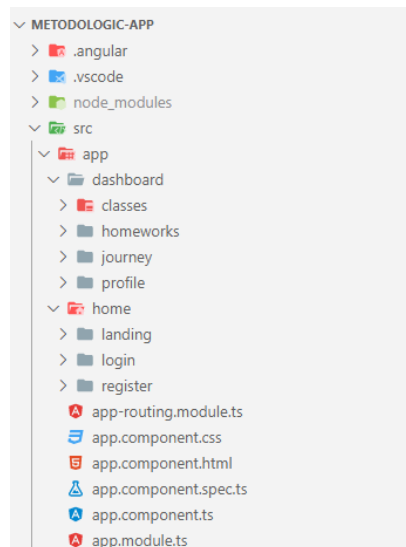


FIGURA 5. Estrutura dos arquivos do front-end. Fonte: Elaboração da autora com a ferramenta X.

Cada um dos componentes precisa dos seguintes arquivos para seu funcionamento: component.html que define a estrutura do conteúdo, component.css responsável pela estilização, component.spec.ts responsável para criar os testes, component.ts responsável pela lógica do componente e conexão com a api. Por fim, há os arquivos referentes ao módulo que permite a configuração de rotas (Figura 7).

```

<div class="form-login">
  <p class="variant4">Login</p>
  <mat-form-field appearance="outline" color="primary" style="width: calc(30% + 90px);">
    <mat-label style="flex-direction: column;">
      <svg width="16" height="15" viewBox="0 0 27 27" fill="none">
        <path d="M13.3739 15.2845C17.595" fill="#1A1A37"/>
      </svg>
      Email
    </mat-label>
    <input matInput />
  </mat-form-field>
  //Outros campos de entrada seguindo o mesmo padrão
</div>

```

FIGURA 7. Fragmento do código html da página de login. Fonte: Elaborado pela autora.

Quanto ao desenvolvimento do back-end, utilizou-se o framework Spring Boot e seguiu-se a seguinte organização: entidades, controladores e repositórios.

Para cada entidade que representa uma tabela do banco de dados, foram definidos seus atributos e seus respectivos tipos. Foi utilizada a biblioteca jakarta para a configuração da tabela a partir dos *decorators* (tags com prefixo de @ que descrevem funcionalidades). A Figura 8 representa a entidade Alunos e suas colunas a partir de seus atributos definido.

```
//Importações das bibliotecas
@Entity
@Table(name="ALUNOS")
@Data
@EqualsAndHashCode(onlyExplicitlyIncluded = true)
public class Aluno {

    @Id
    @Column( name="ALN_ID" )
    @GeneratedValue( strategy = GenerationType.IDENTITY )
    @EqualsAndHashCode.Include
    private Long id;

    @NotNull( message = "Nome é obrigatório" )
    @Column( name="USR_NOME" )
    @Length( max = 60 )
    private String nome;
    //E o padrão segue para todos atributos da tabela Aluno
}
```

FIGURA 8. Código java da entidade Aluno. Fonte: Elaborado pela autora.

CONCLUSÕES

O aplicativo *Metodologic* representa um projeto computacional para compor um jogo com a lógica de fases, feedback, pontuação por meio de elementos e componentes web simples. Além disso, também abrange a área educacional ao apresentar conteúdos relevantes para os estudos acadêmicos, a fim de facilitar o aprendizado por meio de elementos divertidos e envolventes.

CONTRIBUIÇÕES DOS AUTORES

R.F.L.R contribuiu com a análise e coleta dos dados referentes ao conteúdo teórico do jogo, procedeu com a designação do escopo, metodologia e com a orientação do projeto. G.A.B.L contribuiu com a análise dos dados, desenvolvimento e escrita do projeto. Todos os autores contribuíram com a revisão do trabalho e aprovaram a versão submetida.

AGRADECIMENTOS

À Pró-reitoria de Pesquisa (PRP) do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) - Fomento por meio do PIBIFSP - edital DRG-SBV 039/2022, de 20 de setembro de 2022, aos docentes da instituição, à Profa. Dra. Rosana Ferrareto Lourenço Rodrigues, aos discentes e equipe de bolsistas, às instituições parceiras do projeto e ao grupo de pesquisa aCOMTECe.

REFERÊNCIAS

ALMEIDA, F. D. et al. **O jogo do método: jogos de tabuleiro como suporte ao ensino da disciplina metodologia científica.** Research, Society and Development, v. 6, p. 23, 2017.

ANGULAR. **Developer guide.** 2023. Acesso em: 8 abr. 2023.

MATTAR, J. **Games em Educação: como os nativos digitais aprendem.** São Paulo: Pearson Universidades, 2009.

MATTAR, J.; SOUZA Álvaro L. M.; BEDUSCHI, J. de O. **Games para o ensino de metodologia científica: revisão de literatura e boas práticas.** Educação, Formação e Tecnologias, v. 10, p. 16, 2017.

VOLPATO, G. L. **Método lógico para redação científica.** Botucatu: Best Writing, 2017.