**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA**
**CAMPUS DE CAJAZEIRAS**
**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DIUARI PESSOA BEZERRA**

# Enigmarity: UMA API REST PARA O GERENCIAMENTO DE UM JOGO EDUCATIVO PARA O AUXÍLIO NO ENSINO DE SEGURANÇA DE DADOS PARA ALUNOS DE ENSINO MÉDIO

**CAJAZEIRAS-PB**
**2025**

**Diuari Pessoa Bezerra**

# Enigmarity: UMA API REST PARA O GERENCIAMENTO DE UM JOGO EDUCATIVO PARA O NO AUXÍLIO NO ENSINO DE SEGURANÇA DE DADOS PARA ALUNOS DE ENSINO MÉDIO

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Campus Cajazeiras, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

**Orientador:** Prof. Dr. Fabio Gomes de Andrade

**Cajazeiras-PB**
**2025**

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

B574e Bezerra, Diuari Pessoa.

Enigmarity : uma API Rest para o gerenciamento de um jogo educativo para o auxílio no ensino de segurança de dados para alunos de ensino médio / Diuari Pessoa Bezerra. – Cajazeiras, 2025.
28f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise de Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador: Prof. Dr. Fabio Gomes de Andrade.

1. Desenvolvimento de software. 2. Gamificação. 3. Educação. 4. Segurança de dados. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ CDU: 004.42:371.382(043.2)

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

DIUARI PESSOA BEZERRA

**Enigmarity: UMA API REST PARA O GERENCIAMENTO DE UM JOGO EDUCATIVO PARA O AUXÍLIO NO ENSINO DE SEGURANÇA DE DADOS PARA ALUNOS DE ENSINO MÉDIO**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Dr. Fabio Gomes de Andrade

Aprovada em: **09 de setembro de 2025.**

Prof. Dr. Fabio Gomes de Andrade - Orientador

Prof. Me. Diogo Dantas Moreira - Avaliador

IFPB - Campus Cajazeiras

Prof. Me. Afonso Serafim Jacinto - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto**, **COORDENADOR(A) DE CURSOS - FUC1 - CADS-CZ**, em 10/09/2025 16:01:48.
- **Fabio Gomes de Andrade**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/09/2025 16:57:28.
- **Afonso Serafim Jacinto**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 11/09/2025 09:52:53.
- **Diogo Dantas Moreira**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 11/09/2025 10:26:39.

Este documento foi emitido pelo SUAP em 10/09/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código 764382
Verificador: 75980ca3a2
Código de Autenticação:

## RESUMO

A segurança da informação é um tema cada vez mais relevante no contexto digital, especialmente para jovens que utilizam a internet sem o devido conhecimento sobre práticas seguras. Este trabalho propõe o desenvolvimento de um jogo educativo baseado em enigmas, voltado para o ensino de segurança de dados a alunos do ensino médio. Através de desafios interativos, a solução busca promover o aprendizado sobre ameaças cibernéticas, proteção de dados e prevenção de golpes online. O projeto tem como foco a implementação do backend do jogo, garantindo a lógica e a estrutura necessárias para o seu funcionamento.



**Palavras-chave:** Segurança da informação, jogo educativo, ensino de segurança digital, proteção de dados.

## ABSTRACT

Information security is an increasingly relevant topic in the digital age, especially for young people who use the internet without proper knowledge of safe practices. This work proposes the development of an educational puzzle-based game aimed at teaching data security to high school students. Through interactive challenges, the solution seeks to promote learning about cyber threats, data protection, and online fraud prevention. The project focuses on implementing the game's backend, ensuring the necessary logic and structure for its functionality.



**Keywords:** Information security, educational game, digital security education, data protection.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| API | Application ProgrammingI interface |
| BCrypt | Blowfish Crypt |
| CRUD | Create, Read, Update and Delete |
| CSR | Controller–Service–Repository |
| GridFS | Grid File System |
| HTTP | HyperText Transfer Protocol |
| JWT | JSON Web Token |
| LMS | Learning Management System |
| REST | Representational State Transfer |
| SQL | Structured Query Language |
| TCC | Trabalho de Conclusão de Curso |

**SUMÁRIO**

## 1. INTRODUÇÃO

A segurança da informação tem se tornado cada vez mais relevante na sociedade atual, onde a conectividade digital é uma parte fundamental do dia a dia das pessoas. O aumento do uso de redes sociais, plataformas online e jogos digitais trouxe não apenas benefícios, mas também riscos significativos relacionados à privacidade e à proteção de dados. De acordo com Bastos (2014), conversas aparentemente inofensivas em aplicativos de conversas podem representar perigos. Por isso, é fundamental que os usuários sejam orientados sobre como navegar na internet com segurança e que aprendam a lidar com situações de risco. Entre os usuários vulneráveis estão os jovens, que frequentemente interagem com o ambiente digital sem ter conhecimento suficiente para identificar ameaças ou evitar práticas inseguras.

Uma das práticas mais usadas por criminosos é a engenharia social, que é uma técnica de manipulação para obter informações confidenciais, conforme explicado por Conceição (2017).

> A Engenharia Social usa a persuasão e o mérito de ser um bom comunicador, para enganar as pessoas. Há quem já tenha ouvido falar do termo no ambiente fora da rede, porém, além dos ataques por intermédio do discurso e exposição oral, o engenheiro social pode utilizar de suas artimanhas para induzir o indivíduo a disponibilizar senhas e logins, dentro de uma empresa, por exemplo, e repassar a um hacker, ou pode ser que o próprio atue nas duas funções. (CONCEIÇÃO, 2017, p. 2).

No Brasil, a situação é ainda mais alarmante. Uma pesquisa realizada pela Sophos[1] em 2021 destaca o aumento de ataques cibernéticos, como roubo de credenciais e tentativas de *phishing* (em inglês corresponde a “pescaria”), que tem o objetivo de “pescar” informações e dados pessoais importantes através de mensagens falsas, especialmente em períodos de maior dependência da internet, como ocorreu durante a pandemia da COVID-19. Esses ataques exploram a falta de conscientização e educação em segurança digital, evidenciando a necessidade de soluções que combinem acessibilidade e eficácia. Como explicam Neves e Borges (2020, p. 2), o “Comportamento infocomunicacional refere-se às formas como as

[1]rodapé https://securityleaders.com.br/ataques-de-phishing-aumentaram-em-70-das-organizacoes-durante-a-pandemia/

pessoas se informam e se comunicam, ou seja, aos modos como consomem informação, mas também a produzem, comunicam e se relacionam".

Neste contexto, o presente trabalho busca abordar o ensino de segurança de dados de maneira prática e inovadora, propondo estratégias para promover a conscientização sobre segurança da informação, com foco em jovens do ensino médio. Esse público, que é amplamente ativo no ambiente digital, representa tanto uma população vulnerável quanto uma oportunidade para a disseminação de boas práticas de proteção *online*.

## 1.1 Motivação

Muitos jovens utilizam a internet diariamente para acessar redes sociais, jogos e outras plataformas. Entretanto, muitos deles não possuem qualquer instrução sobre boas práticas de segurança da informação. Essa falta de conhecimento os torna vulneráveis a riscos, como o preenchimento de formulários inseguros, a aceitação irrefletida de termos e condições, e o compartilhamento imprudente de dados pessoais.

Os crimes cibernéticos passaram a ser cometidos com mais intensidade nos últimos anos, como explica MARICHAL.

> (...) refere-se ao aumento de casos de crimes cibernéticos decorrente da captura de informações pessoais e institucionais. Delitos estes, motivados principalmente no período da pandemia do novo coronavírus (2020 2022), o que levou inclusive à criação da Lei nº 14.155/2021 que aumenta a gravidade de crimes como o estelionato ocorrido em meios digitais. Dessa forma, percebe-se que estes tipos de crimes vêm recebendo cada vez mais atenção por parte do Poder Legislativo. (MARICHAL,2022)

Embora atualmente existam recursos disponíveis sobre bons hábitos de segurança da informação, como blogs, cursos e plataformas informativas, a falta de acesso ou de interesse por esses conteúdos contribui para que muitos jovens permaneçam desinformados sobre a importância da proteção *online*. Esse cenário resulta em uma maior exposição desse público a ataques cibernéticos e violações de privacidade.

Diante disso, este Trabalho de Conclusão de Curso (TCC) oferece uma solução prática e acessível para a educação em segurança da informação. Por meio de um jogo de enigmas, ele proporciona uma experiência interativa e envolvente, na tentativa de aumentar o interesse e o aprendizado sobre o tema. Focado em alunos do ensino médio, o trabalho visa não apenas conscientizar os jovens sobre os riscos do ambiente digital, mas também promover práticas seguras e responsáveis de navegação.

## 1.2 Objetivos

Esta seção descreve os objetivos deste trabalho.

### 1.2.1 Objetivo Geral

O objetivo geral deste TCC consiste em desenvolver a API de um jogo educativo com potencial facilitador no ensino de segurança da informação, tendo como público alvo os alunos do ensino médio.

### 1.2.2 Objetivos Específicos

O trabalho tem também os seguintes objetivos específicos:

- Pesquisar e revisar a literatura sobre segurança da informação e sua relevância para o público jovem, com foco em práticas seguras no uso da internet;
- Tornar os alunos do ensino médio aptos a utilizarem a internet de forma mais segura e consciente com relação à segurança de dados;
- Tornar o aprendizado de segurança de dados mais interessante para os alunos do ensino médio, por meio de uma alternativa prática e lúdica;
- Avaliar o impacto do jogo na compreensão dos alunos sobre práticas seguras na internet, abordando temas como identificação de ameaças cibernéticas, proteção de dados pessoais e prevenção de golpes online.

### 1.3 Organização do Documento

O restante deste documento está organizado da seguinte forma: o Capítulo 2 apresenta a metodologia a ser utilizada no desenvolvimento deste TCC, detalhando as atividades que foram desenvolvidas; o Capítulo 3 descreve a solução proposta, abordando as suas funcionalidades e alguns aspectos do seu projeto. Finalmente, o Capítulo 4 consiste nas considerações finais.

## 2. METODOLOGIA

Para atingir os objetivos deste trabalho, a metodologia escolhida para o seu desenvolvimento foi composta por uma série de etapas que foram desenvolvidas de forma sequencial e interativa. As atividades desenvolvidas foram:

- **Revisão da literatura (A1):** esta atividade incluiu uma revisão da literatura relacionada à temática sobre segurança digital , com foco em práticas seguras no uso da internet;
- **Levantamento de requisitos (A2):** esta atividade consistiu no levantamento dos requisitos funcionais necessários para implementar o jogo proposto por este TCC;
- **Definição da arquitetura (A3):** esta atividade consistiu na definição da arquitetura do jogo, identificando os seus principais componentes e suas respectivas funções;
- **Implementação (A4):** nesta atividade o *back-end* do jogo proposto foi implementado, usando-se as tecnologias e ferramentas definidas em seu projeto arquitetural;.
- **Elaboração do documento do TCC (A5):** nesta atividade foi feita a elaboração do documento final do TCC.

# 3. SOLUÇÃO PROPOSTA

Este capítulo descreve o jogo *Enigmarity,* que é  o jogo interativo de enigmas sobre segurança da informação proposto neste TCC. A solução desenvolvida neste TCC é responsável por gerenciar os dados relacionados ao conteúdo educativo do jogo, como  as perguntas e respostas de cada enigma, o controle do progresso dos usuários e o acompanhamento dos dados sobre o tempo gasto em cada enigma, proporcionando a infraestrutura necessária para a integração com o f*rontend*.

## 3.1 Requisitos Funcionais

Esta seção descreve os requisitos funcionais que devem ser oferecidos pelo jogo  proposto neste TCC. Esses requisitos são apresentados no Quadro 2

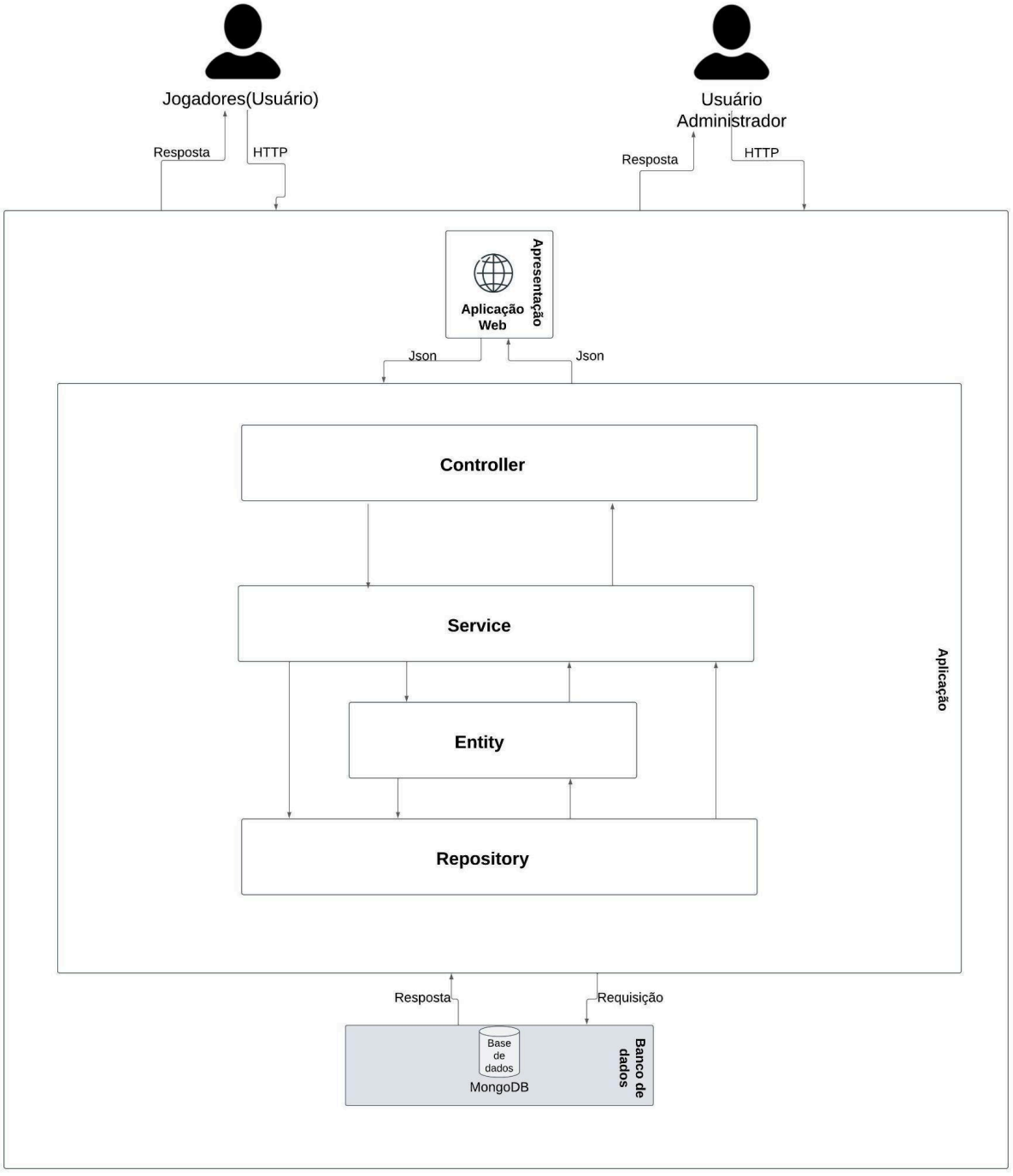
***Quadro 2 - Requisitos funcionais***

| Requisito | Descrição |
|---|---|
| RF01 | O jogo deve permitir o gerenciamento de enigmas por parte do usuário administrador, possibilitando a criação, recuperação, atualização e exclusão de enigmas |
| RF02 | O jogo deve possibilitar o gerenciamento de usuários, incluindo a criação, atualização e exclusão de contas |
| RF03 | O jogo deve realizar a autenticação dos seus usuários |
| RF04 | O jogo deve registrar e exibir a pontuação do usuário com base nos enigmas que ele já resolveu |
| RF05 | O jogo deve controlar o progresso do usuário ao longo do jogo, exibindo o tema do desafio atual e o tópico da área de segurança da informação ao qual ele está relacionado. O jogo também deve permitir que o usuário salve o seu progresso para continuar o jogo em outro momento |
| RF06 | O jogo deve permitir ao usuário solicitar uma dica para ajudá-lo a solucionar o enigma da fase atual |
| RF07 | O jogo deve permitir que o usuário comece um novo jogo |
| RF08 | O jogo deve permitir ao usuário administrador definir e registrar a ordem de apresentação dos enigmas, utilizando uma lógica pré-estabelecida, garantindo uma experiência consistente para o usuário |
| RF09 | O jogo deve verificar se um enigma foi resolvido corretamente, registrando a resposta do usuário e comparando-a com os critérios de solução definidos previamente |
| RF10 | O jogo deve permitir que o administrador cadastre, atualize e remova os temas relacionados à segurança da informação que serão abordados durante o jogo, associando-os aos respectivos enigmas |

## 3.2 Arquitetura

Esta seção apresenta a arquitetura do jogo proposto, que foi organizada em camadas para facilitar o seu desenvolvimento, manutenção e evolução. A Figura 1 ilustra essa arquitetura. Nela, percebe-se que a arquitetura proposta é dividida em três camadas principais: apresentação, aplicação e dados.

**Figura 1 – Arquitetura do jogo**




Fonte : elaborado pelo autor

O jogo *Enigmarity* foi desenvolvido como uma aplicação *API REST*, estruturada em múltiplas camadas e implementada usando a tecnologia *Spring Boot*. A arquitetura usada para o seu desenvolvimento é baseada no padrão *Controller–Service–Repository* (CSR), o que garante a separação de responsabilidades, o baixo acoplamento e maior facilidade de manutenção. Esse modelo organiza a aplicação em camadas bem definidas: interface de API (Controller), lógica de negócio (Service) e persistência de dados (Repository).

A camada de apresentação disponibiliza o acesso às funcionalidades do jogo por meio de *endpoints RESTful*, que seguem convenções adequadas de verbos HTTP (GET, POST, PUT, DELETE). A comunicação é realizada usando o formato *JSON*, assegurando padronização e interoperabilidade. Essa camada também é integrada a ferramentas de documentação automática, como *Swagger*, e contempla o versionamento para evolução contínua da API.

A segurança é implementada através de autenticação baseada na tecnologia *JSON Web Token (JWT)*, autorização por papéis (usuário comum ou administrador) e validação rigorosa dos dados de entrada. Além disso, os controladores retornam respostas semânticas adequadas por meio de códigos de status HTTP, como 200 (sucesso), 201 (criação), 401 (não autorizado) e 404 (não encontrado).

A camada de aplicação concentra a lógica de negócio, sendo responsável por coordenar as operações entre as diferentes partes do sistema. Essa camada é composta por controladores, serviços, entidades e repositórios. Os controladores recebem as requisições, aplicam as validações iniciais e invocam os serviços requisitados.

Os serviços dessa camada implementam as regras de negócio, garantindo a integridade das operações. Entre as suas funções, estão incluídos o controle sequencial dos enigmas, o cálculo de pontuação de cada jogador (considerando tempo e tentativas), a prevenção de exclusão de temas com enigmas vinculados e a aplicação de regras de acesso com base em perfis.

As entidades representam os modelos de dados utilizados pelo sistema, refletindo a estrutura dos documentos armazenados no banco de dados. Cada entidade corresponde a uma coleção, como enigma, usuário e tema. Para cada

coleção foram definidos atributos e relacionamentos que asseguram a consistência entre a lógica de negócio e a camada de persistência.

Os repositórios realizam a comunicação direta com o banco de dados, constituindo a camada de persistência responsável pelo armazenamento, recuperação e manipulação dos dados.

A camada de persistência representa o banco de dados no qual os dados do jogo são armazenados. Ela foi implementada usando o *MongoDB*, um banco de dados NoSQL que oferece flexibilidade de esquema e escalabilidade horizontal. O sistema utiliza o *Spring Data MongoDB* para abstrair o acesso aos dados, permitindo a criação de consultas derivadas, consultas personalizadas e a paginação com ordenação eficiente.

A arquitetura adotada proporciona vantagens em termos de manutenibilidade, pela separação de responsabilidades; escalabilidade, pelo suporte a crescimento de usuários e conteúdo; segurança, com autenticação robusta e criptografia de senhas via BCrypt; e performance, através de indexação e otimização de consultas. Além disso, o sistema mantém a integridade referencial entre os documentos, assegurando consistência e confiabilidade nos dados armazenados.

### 3.3 Modelagem do banco de dados

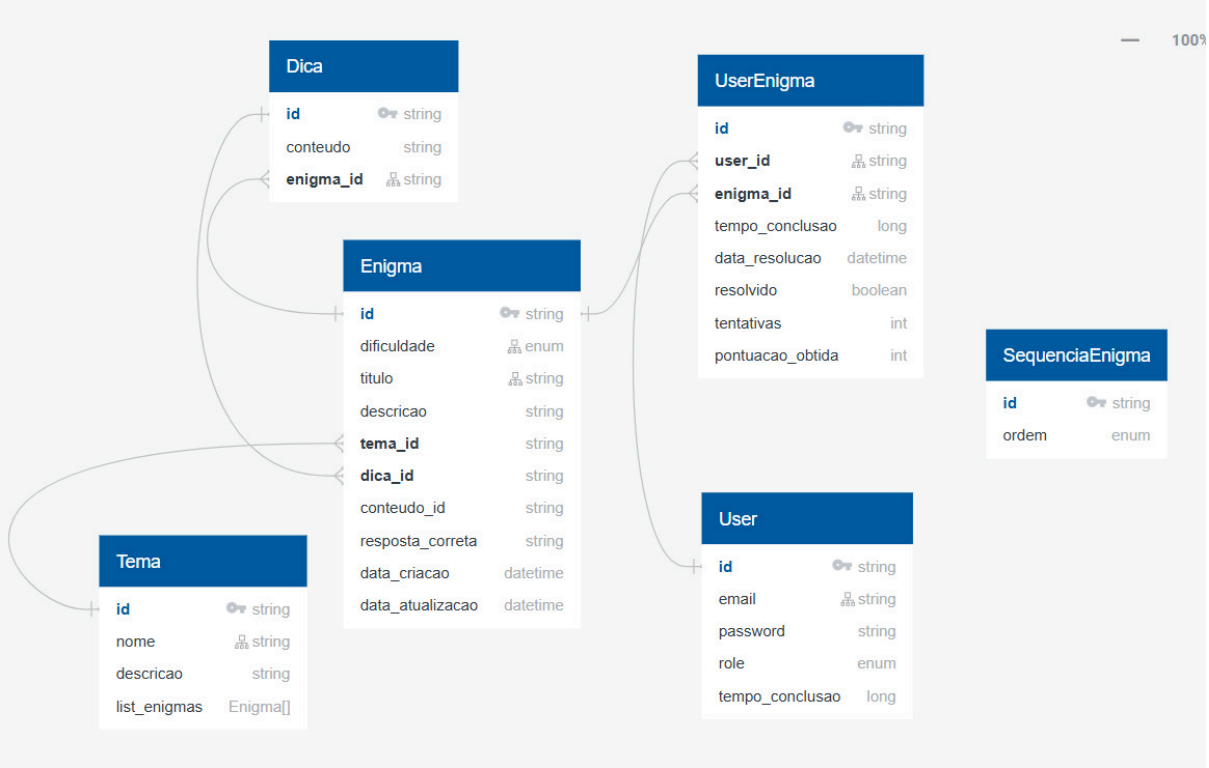
A modelagem dos dados é uma etapa essencial no desenvolvimento do software, pois organiza de maneira estruturada todas as informações que o sistema irá processar. No caso do jogo proposto neste TCC, optou-se pela utilização do banco de dados *noSQL* orientado a documentos *MongoDB,* pois a sua linguagem de consulta baseada em JSON proporciona um alto grau de conforto e harmonia no uso desse banco de dados (BANKER et al., 2016, tradução nossa). A escolha do MongoDB se justifica por diversos motivos:

- **Flexibilidade:** ele permite armazenar dados em formato JSON/BSON, possibilitando a manipulação de documentos sem a necessidade de um esquema rígido, o que é ideal para aplicações que lidam com informações dinâmicas e heterogêneas;

- **Escalabilidade**: ele oferece suporte à alta performance e escalabilidade horizontal, facilitando o crescimento do sistema conforme o aumento de usuários e dados;
- **Integração**: ele facilita a integração com APIs modernas e frameworks de desenvolvimento web, permitindo uma comunicação ágil entre o backend e outros componentes do sistema.

O esquema lógico do banco de dados, que é mostrado na Figura 2, é composto pelas coleções *Desafio*, *Dica*, *Tema*, *Dificuldade*, *Usuario* e *Usuario_Desafio*, além dos relacionamentos existentes entre elas.

**Figura 2 – Modelagem de Dados**




A coleção *Enigma* constitui o núcleo central do esquema lógico, sendo responsável pelo armazenamento dos desafios propostos aos jogadores. Cada documento dessa coleção contém atributos como o *id*, que corresponde ao identificador único do enigma (chave primária do MongoDB); o *título*, que representa

o nome do enigma e é indexado para acelerar a resolução de consultas; a *descrição*, que contém o texto detalhado do desafio; a *dificuldade*, definida por uma enumeração que classifica o nível de complexidade em fácil, médio ou difícil; o *temaId*, que relaciona o enigma a um tema; o *dicaId*, que indica a dica vinculada ao enigma; e o *conteudoId*, que identifica arquivos multimídia, como imagens e áudios, armazenados via GridFS. Além disso, são registrados a resposta correta, a data de criação e a data da última atualização.

A coleção *Dica* é responsável pelo armazenamento de mensagens auxiliares que podem ser utilizadas pelos jogadores durante a resolução dos enigmas. Os seus atributos principais são o *id*, que identifica de forma única a dica; o *conteúdo*, que corresponde ao texto de orientação; e o *enigmaId*, que referencia o enigma ao qual a dica se refere. Cada dica mantém uma relação de cardinalidade um para um com um enigma, permitindo que os jogadores solicitem auxílio de forma direcionada.

A coleção *Tema* organiza os enigmas em categorias temáticas. Essa estrutura contempla atributos como o *id*, que corresponde ao identificador único do tema; o *nome*, que representa a denominação da categoria e é indexado para otimização de consultas; a *descrição*, que detalha a área temática; e o campo *listEnigmas*, que armazena os objetos do tipo *Enigma* relacionados por meio de *embedding*. Essa modelagem permite agrupar enigmas por áreas de interesse, facilitando a navegação e a escolha dos usuários.

A coleção *User* gerencia as informações dos jogadores e viabiliza o processo de autenticação no sistema. Entre os atributos definidos estão o *id*, que identifica o usuário de forma única; o *email*, utilizado como credencial de login; o *password*, que armazena a senha criptografada com BCrypt; o *role*, que define o perfil de acesso ; e o *tempoConclusao*, que corresponde ao tempo total gasto pelo jogador para concluir todos os engimas. A implementação dessa coleção foi integrada ao *Spring Security*, por meio da interface *UserDetails*, possibilitando a autenticação e a autorização baseadas em perfis de acesso.

A coleção *UserEnigma* tem como finalidade registrar o histórico de interação entre os usuários e os enigmas. Os atributos que a compõem incluem o *id* do registro, o *userId,* que referencia o usuário, o *enigmaId,* que referencia o enigma, o

*tempoConclusao*, a data de resolução, o campo *resolvido*, que indica se o desafio foi ou não solucionado, o número de tentativas realizadas e a pontuação final obtida. Essa coleção viabiliza o monitoramento detalhado do desempenho e do progresso individual de cada jogador.

A coleção *SequenciaEnigma* define a ordem de apresentação dos enigmas de forma global para todos os usuários. Para isso, ela contém o *id*, que é fixo e é denominado *sequencia_global*, e o campo ordem, responsável por indicar o tipo de ordenação aplicada, definido por meio de uma enumeração denominada *TipoOrdenacao*. Essa configuração garante consistência na experiência de jogo.

O sistema também implementa funcionalidades de gestão de conteúdo multimídia, como o *upload* de arquivos (imagens e áudios) o *download* seguro, com controle de acesso mediante autenticação, e a visualização online, que possibilita a exibição direta de arquivos no navegador.

Para garantir o bom desempenho do banco de dados, foram criados índices para campos críticos usados em consultas frequentes, incluindo campos de busca **(**título, nome e dificuldade**)**, chaves de relacionamento **(***temaId*, *dicaId*, *userId* e *enigmaId***)** e campos de ordenação temporal **(**data de criação e data de atualização**)**. Essa abordagem garante flexibilidade, escalabilidade e integridade referencial, e proporciona um desempenho adequado para o contexto de um sistema de jogos educativos.

## 3.4 Implementação

A implementação do jogo *Enigmarity* seguiu a arquitetura em camadas descrita na Figura 1, contemplando as camadas de apresentação, aplicação (controladores, serviços, entidades e repositórios) e dados.

### 3.4.1 Controladores

Os controladores têm como função receber as requisições HTTP, validar os dados de entrada e invocar os serviços responsáveis pela lógica de negócio. Eles foram implementados usando a tecnologia *Spring Boot*. Cada controlador é

mapeado para um conjunto de endpoints REST. Durante a implementação do jogo foram desenvolvidos os seguintes controladores:

- **AuthenticationController**: este controlador é responsável pela autenticação e registro de usuários, com validação de credenciais e geração de tokens JWT;
- **EnigmaController**: este controlador implementa as operações de gerenciamento de enigmas, além do upload e download de arquivos multimídia armazenados no GridFS;
- **UserController**: este controlador gerencia os perfis de usuários, estatísticas individuais e alteração de senha;
- **UserEnigmaController**: este controlador processa as tentativas de resolução de enigmas, registra o histórico de interações e calcula a pontuação obtida pelo usuário;
- **TemaController:** este controlador implementa o gerenciamento completo de temas e categorias de enigmas, oferecendo operações CRUD;
- **DicaController**: este controlador gerencia a criação e vinculação de dicas aos enigmas, controlando sua liberação progressiva;
- **RankingController:** este controlador implementa a lógica de classificação dos usuários, calculando posições e rankings baseados em pontuação e desempenho na resolução de enigmas;
- **SequenciaEnigmaController**: este controlador define a ordem global de apresentação dos desafios, gerenciado pré-requisitos e desbloqueio progressivo de enigmas conforme o progresso individual dos usuários;

### 3.4.2 Serviços

A camada de serviços concentra a lógica de negócio da aplicação. Nela, foram implementadas as regras que garantem a integridade e o funcionamento correto do sistema. Entre as principais implementações da lógica de negócio destacam-se:

- O cálculo da pontuação**,** com base no tempo gasto e no número de tentativas do usuário;

- A gestão do progressão sequencial, garantindo que os enigmas sejam apresentados na ordem estabelecida durante  a configuração do jogo;
- a validação de unicidade, como a impossibilidade de se cadastrar dois usuários com o mesmo e-mail;
- o controle de integridade, como a prevenção da exclusão de temas que ainda possuam enigmas associados.

Os serviços também centralizam o acesso ao repositório de dados, garantindo que toda a lógica esteja isolada dos controladores, favorecendo a manutenção e os testes do sistema.

### 3.5.3 Entidades

As entidades são responsáveis por representar os objetos de domínio do jogo, servindo como modelo para o mapeamento dos dados armazenados no banco. Cada entidade corresponde a uma coleção do MongoDB, refletindo a estrutura de informações utilizadas na aplicação. Cada entidade possui atributos que representam os dados essenciais do jogo, bem como as anotações necessárias para o mapeamento com o Spring Data MongoDB. Além disso, as entidades garantem a coerência entre os dados persistentes e a lógica de negócio definida nos serviços.

Durante a implementação do jogo, foram desenvolvidas as seguintes entidades:

- **User**: representa os usuários cadastrados no jogo, contendo informações como nome, e-mail, senha, estatísticas e papéis de acesso (roles);
- **Enigma**: descreve os enigmas do jogo, armazenando o  título, a descrição, o nível de dificuldade, a mídia associada (armazenada no GridFS).
- **Tema**: define a categorização dos enigmas, organizando-os em áreas temáticas;
- **Dica**: representa as dicas que podem ser solicitadas pelos usuários para a resolução dos enigmas. Cada dica é vinculada a um único enigma;

- **UserEnigma**: armazena o relacionamento entre usuários e enigmas, registrando o número de tentativas, os acertos, o tempo de resolução e a pontuação final obtida;
- **Ranking**: reflete a classificação geral dos jogadores, com base nas estatísticas calculadas pelo jogo;.
- **SequenciaEnigma**: define a ordem global de apresentação dos enigmas no jogo, garantindo, assim, uma progressão sequencial.

As entidades são essenciais para manter a integridade estrutural da aplicação, já que todas as operações de negócio e persistência partem de modelos consistentes.

### 3.5.4 Repositórios

Os repositórios realizam a comunicação com o banco de dados, servindo como intermediários entre as entidades e a lógica de negócio implementada nos serviços. Eles foram implementados utilizando o *Spring Data MongoDB*, que fornece uma abstração sobre o acesso ao banco de dados. Essa camada possibilita a definição de consultas de duas formas:

- **Consultas derivadas**: representam as consultas geradas automaticamente a partir da nomenclatura dos métodos, como *findByEmail* (para a recuperação de usuários) e *findByDificuldade* (para a recuperação de enigmas por nível de dificuldade);

- **Queries personalizadas**: representam as consultas definidas por meio da anotação *@Query***,** que são utilizadas em cenários que exigem maior flexibilidade.

Além disso, os repositórios oferecem o suporte necessário para a paginação **e** ordenação dos resultados das consultas, o que é fundamental para otimizar consultas em grandes volumes de dados. Por fim, os repositórios foram implementados seguindo o *Repository Pattern*, o que promove o baixo acoplamento e facilita futuras alterações na persistência.

# 4. CONSIDERAÇÕES FINAIS

A vulnerabilidade dos jovens usuários diante dos riscos do ambiente digital evidencia um problema crítico: a falta de instrução adequada em segurança da informação expõe esses jovens a ataques cibernéticos, fraudes e outros perigos online. Essa lacuna de conhecimento não compromete apenas a integridade dos seus dados pessoais, mas também dificulta o desenvolvimento de uma cultura digital segura e consciente.

Para enfrentar esse desafio, este TCC propôs o *Enigmarity*, um jogo educativo, baseado em enigmas interativos, para o ensino de segurança da informação. O seu objetivo consiste em abordar o ensino de segurança de dados de forma lúdica e atrativa, engajando os alunos por meio de enigmas que, ao serem solucionados, reforçam conceitos essenciais para a proteção de informações e práticas seguras na internet.

O desenvolvimento do jogo seguiu uma abordagem orientada a objetos, utilizando entidades bem definidas para representar os elementos centrais do sistema e garantir a integridade dos dados. A arquitetura implementada, baseada na integração entre controladores, serviços e repositórios, permitiu o desenvolvimento completo de todas as funcionalidades planejadas: cadastro e autenticação de usuários, gerenciamento dinâmico de enigmas, sistema de pontuação baseado em desempenho e controle de progresso sequencial.

A modelagem clara das entidades facilitou significativamente a integração com o banco de dados MongoDB, assegurando operações eficientes de armazenamento e recuperação de dados. Como resultado, todos os objetivos estabelecidos para o projeto foram plenamente alcançados.

## 4.1 Trabalhos Futuros

Alguns trabalhos futuros podem ser desenvolvidos para a evolução do trabalho proposto neste TCC. Exemplos de evoluções importantes incluem:

- **Integração com sistemas de gamificação externos**: a implementação de recursos que permitam sincronizar conquistas e progressos dos usuários com

plataformas educacionais ou sistemas de gestão de aprendizagem (LMS) pode ampliar o alcance do jogo no contexto acadêmico;

- **Aplicação de análise de dados e relatórios**: o desenvolvimento de módulos que coletem e processem métricas de desempenho dos usuários permitiria a geração de relatórios sobre a evolução individual e coletiva. Esses relatórios auxiliariam professores e instituições de ensino a identificar lacunas de conhecimento e personalizar estratégias educativas;
- **Modo multiplayer**: a introdução de funcionalidades que permitam a interação entre múltiplos jogadores em tempo real ou em competições assíncronas estimularia a colaboração, o aprendizado em grupo e a motivação dos usuários por meio de desafios cooperativos ou competitivos.

# REFERÊNCIAS


Ataques de phishing aumentaram em 70% durante a pandemia. Disponível em: https://securityleaders.com.br/ataques-de-phishing-aumentaram-em-70-das-organizacoes-durante-a-pandemia/. Acesso em: 10 mar. 2025.

**AUTH0. JWT Handbook: Zero to Mastery in JSON Web Tokens.** [S. l.]: Auth0, 2020. E-book. Disponível em: https://assets.ctfassets.net/2ntc334xpx65/o5J4X472PQUI4ai6cAcqg/13a2611de03b2c8edbd09c3ca14ae86b/jwt-handbook-v0_14_1.pdf. Acesso em: 03 set. 2025.

BANKER, K. et al. **MongoDB in Action**: Covers MongoDB version 3.0. Londres, England: Simon and Schuster, 2016.

BASTOS, António José. Os perigos da internet - alguns cuidados a ter com o seu uso. 2014. Disponível em: < http://www.prof2000.pt/users/lbastos/osperigosdainternet.htm >. Acesso em: 14 jan. 2025.

CONCEIÇÃO, J. P. da. A arte da fraude no campo da informação: engenharia social, big data e a manipulação do usuário na rede. Bibliotecas Universitárias: pesquisas, experiências e perspectivas, Belo Horizonte, v. 4, n. 1, p. 36-45, jan./jun. 2017. Disponível em: https://periodicos.ufmg.br/index.php/revistarbu/article/view/3110. Acesso em: 6 fev. 2025.

MARICHAL. Pedro Luiz de. **Phishing na era da informação: relevância da proteção de dados pessoais**. Porto Alegre, 2022 Disponível em: <https://lume.ufrgs.br/bitstream/handle/10183/258868/001170484.pdf;jsessionid=0BAD60D5C64CA81F16C4DA44256E4675?sequence=1>. Acesso em: 19 fev. 2025.

**MONGODB**. Manual do banco de dados – Referências de banco de dados. 2025. Disponível em: https://www.mongodb.com/pt-br/docs/manual/reference/database-references/. Acesso em: 29 ago. 2025.

NEVES, B. C.; BORGES, J. **Por que as Fake News têm espaço nas mídias sociais?:** uma discussão a luz do comportamento infocomunicacional. Informação & Sociedade: Estudos, João Pessoa, v. 30, n. 2, p. 1-22, abr./jun. 2020. Disponível em: https://periodicos.ufpb.br/ojs2/index.php/ies/article/view/50410. Acesso em: 19 fev. 2025.

ROSENSTOCK, Josh; **PANELAT, José. Designing APIs with Swagger and OpenAPI.** 1. ed. Shelter Island: Manning Publications, 2023. E-book. Disponível em: https://dl.ebooksworld.ir/books/Designing.APIs.with.Swagger.and.OpenAPI.Ponelat.Rosenstock.Manning.9781617296284.EBooksWorld.ir.pdf. Acesso em: 03 set 2025.

SILVA, Glauco**. MongoDB: Construa Aplicações Com o Banco de Dados NoSQL Mais Popular do Mundo.** 1. ed. São Paulo: Casa do Código, 2021. E-book. ISBN 978-65-86110-92-5. Disponível em: https://books.google.com.br/books?id=kzkzEAAAQBAJ. Acesso em: 03 set.2025.

**SMARTBEAR SOFTWARE. OpenAPI Specification**: Disponível em: https://swagger.io/specification/. Acesso em: 29 ago. 2025.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA**
Campus Cajazeiras - Código INEP: 25008978
Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

# Documento Digitalizado Ostensivo (Público)

## Tcc arquivo

| | |
|---|---|
| **Assunto:** | Tcc arquivo |
| **Assinado por:** | Diuari Bezerra |
| **Tipo do Documento:** | Anexo |
| **Situação:** | Finalizado |
| **Nível de Acesso:** | Ostensivo (Público) |
| **Tipo do Conferência:** | Cópia Simples |

Documento assinado eletronicamente por:

- **Diuari Pessoa Bezerra**, **DISCENTE (202112010011) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 12/09/2025 18:58:24.

Este documento foi armazenado no SUAP em 12/09/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1607801
Código de Autenticação: 3cdcd7f330