

Desenvolvimento de uma Plataforma para o projeto, Quimera de fisiologia animal.

Autores: Fernanda Bonfim Santos¹, Jean-Jacques De Groote²

Colaboradores: Maria Eduarda do Prado Carvalho³, Gelson Genaro⁴

^{1,2,3,4}Centro Universitário Barão de Mauá

¹bonfimfernanda12@gmail.com - Ciência da Computação, ²jean.georges@baraodemaua.br

Resumo

Nos últimos anos, tem havido um crescente interesse no desenvolvimento de métodos educacionais alternativos que promovam a aprendizagem eficaz. Este artigo apresenta a estrutura de *frontend* e *backend* de uma plataforma educacional destinada a estudantes que buscam compreender sua área de estudo. O objetivo principal desta plataforma é oferecer uma abordagem educacional envolvente e motivadora, através de um portal educacional que incorpora salas de estudo gamificadas. Ao utilizar essa abordagem, pretende-se não apenas promover meios para o aprendizado ativo, mas também estimular o interesse dos alunos pelos temas abordados, particularmente na fisiologia animal. A plataforma apresenta métodos interativos e eficazes que visam substituir a necessidade de utilização de animais vivos em contextos educacionais, refletindo preocupações éticas e promovendo práticas mais sustentáveis.

Introdução

A educação em fisiologia animal requer frequentemente o uso de animais vivos em experimentos práticos, o que pode levantar questões éticas e logísticas. O projeto Quimera surge como uma resposta a esses desafios, com a proposta de uma solução inovadora baseada em simulações computacionais e aprendizado interativo. Este artigo explora o desenvolvimento de uma plataforma em web para o projeto Quimera, detalhando sua arquitetura técnica, funcionalidades e aplicações educacionais (DE NORONHA, 2019).

A plataforma para o projeto Quimera foi desenvolvida seguindo uma abordagem estruturada e metodológica. Inicialmente, foram conduzidas revisões de literatura para identificar os requisitos e mecanismos necessários para a criação de uma plataforma educacional eficaz. Com base nessas análises, foram estabelecidos os métodos e tecnologias a serem empregados (HARDER, 2010).

Materiais e Métodos

Os cálculos e dados que levam aos resultados utilizados neste trabalho foram realizados em *backend*, economizando recursos de memória e processamento dos dispositivos utilizados na interação com a plataforma. O processo de desenvolvimento envolve fluxogramas, diagramas de UML e planilhas de dados, no caso o Excel, passando para o desenvolvimento da API REST da plataforma, para o qual foi utilizada a linguagem JavaScript em Node.js, com o framework Express e a biblioteca Mongoose para realizar a conexão com o banco de dados MongoDB Atlas. A API REST foi criada seguindo o padrão de arquitetura MVC (Model, View e Controller). Nesse contexto, os Schemas do usuário e do experimento foram definidos e gerenciados no modelo (model), enquanto as funções lógicas relacionadas à aplicação foram implementadas no controlador (Controller). Isso permitiu a criação dos endpoints da API. Todo o backend foi criado e desenvolvido em programação funcional.

Para o desenvolvimento da plataforma, foram realizadas a criação de um front-end, uma API REST e scripts em Python para operações numéricas da simulação. Para o backend da aplicação, utilizou-se JavaScript em Node.js e a utilização de um banco de dados MongoDB Atlas para registrar as informações de cada atividade na plataforma. Para a interação dos usuários, o sistema utiliza o framework React.js.

O experimento implementado como base para construção da plataforma envolve a correção da queda de água do corpo de um animal através da administração de diferentes tipos de controle do organismo.

Resultados e Discussão

Na construção da plataforma, foi utilizado um experimento base para desenvolvimento com resultados aplicáveis a atividades em sala de aula. A seguir são descritos os elementos do experimento e também os elementos desenvolvidos na plataforma.

Experimento

O experimento implementado envolve a correção da queda de água do corpo de um animal, disponibilizando 20 opções disponíveis para o aluno escolher. Destes, apenas duas, o hormônio ADH (Hormônio Antidiurético) e o controle pelo hipotálamo, são eficazes na correção da queda de água. Para a atividade foi escolhida para efeitos de simulação que o ADH tem uma eficácia de 20%, enquanto o hipotálamo de 80%. Essa escolha se baseia na maior contribuição associada ao hipotálamo. Os outros itens disponíveis não têm efeito na correção da queda de água neste experimento.

Os alunos são instruídos com base em um estudo anterior, que inclui uma introdução à matéria e o caso do animal com a queda de água no corpo. Eles devem usar esse conhecimento para escolher o item correto que irá afetar o animal e evitar a queda anormal da quantidade de água no organismo.

Os seletores na interface do programa estão divididos em duas categorias, com as opções contendo o hormônio ADH e o hipotálamo como as únicas escolhas eficazes. Cada opção tem seu peso associado que representa sua eficácia na correção da queda de água.

Para ajudar a interpretação dos resultados, um gráfico é utilizado para mostrar o que é esperado em uma linha do experimento com as duas opções corretas (ADH e hipotálamo) e o que o aluno escolheu.

Funcionalidades Principais

A Quimera oferece uma variedade de funcionalidades projetadas para enriquecer a experiência de aprendizado dos usuários. Destacam-se:

1. Registro e autenticação de professores e alunos.
2. Criação e gerenciamento de experimentos virtuais.
3. Simulações de quedas anormais de água corporal em animais.
4. Geração de gráficos e análises de dados experimentais.

Para representar as entidades do sistema, foram definidos três modelos de dados: Estudante, Professor e Sala de Experimento. Cada modelo possui atributos específicos que descrevem suas características e relacionamentos com outras entidades. Os modelos foram implementados utilizando o Mongoose Schema, que permite a definição de campos, tipos de dados e validações.

No banco de dados MongoDB, utilizado para desenvolver o projeto, existem três coleções principais: experimento, professor e aluno. Essas coleções armazenam os documentos em formato BSON, que é o formato binário usado pelo MongoDB para representar dados, (Redmond, E., & Wilson, J. R.).

- **Coleção Experimento:** Esta coleção armazena informações relacionadas aos experimentos realizados na plataforma Quimera. Cada documento nesta coleção pode conter detalhes sobre um experimento específico, como seus parâmetros, resultados, datas importantes, entre outros dados relevantes.
- **Coleção Professor:** Aqui são armazenadas as informações dos professores que utilizam a plataforma Quimera. Cada documento nesta coleção pode conter dados como nome, e-mail, instituição, disciplinas ministradas, entre outras informações pertinentes.
- **Coleção Aluno:** Nesta coleção são mantidos os registros dos alunos que interagem com a plataforma Quimera. Cada documento nesta coleção pode incluir informações como nome, e-mail, curso, período acadêmico, entre outros dados relacionados aos alunos.

Os controllers foram responsáveis por implementar as funcionalidades lógicas da API, manipulando os dados recebidos das requisições e interagindo com o banco de dados conforme necessário. Foram criados três controllers principais: um para cada modelo de dados (Estudante, Professor e Sala de Experimento). Cada controller possui diversas funções responsáveis por realizar operações como criação, consulta, atualização e exclusão de registros.

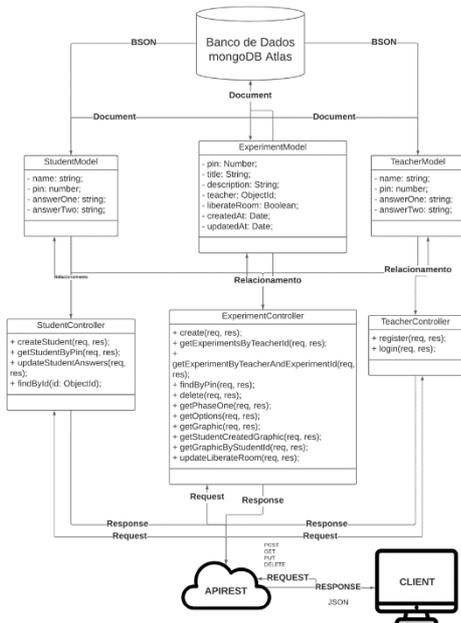
Este diagrama demonstra como a API REST foi desenvolvida com o modelo de arquitetura de software MVC, incluindo seus modelos, esquemas e controladores. Ele mostra como as solicitações e respostas em JSON são feitas para o cliente e como os documentos em formato de BSON (Binary Json) são armazenados nas coleções do banco de dados.

Arquitetura de Pastas & MVC

A arquitetura de pastas do projeto foi organizada de acordo com o padrão MVC (Model-View-Controller), garantindo uma estrutura clara e modular. As pastas foram divididas em modelos, controladores, rotas, serviços, contextos, componentes, páginas e utilitários, cada uma com sua responsabilidade específica. Isso facilitou a manutenção, escalabilidade e extensibilidade do código ao longo do desenvolvimento do projeto. Na

figura 1 é apresentado um diagrama que esboça como foi pensada essa arquitetura, (Gama, N., & Campos, M. L. M.).

Figura 1. MVC - Diagrama da Api Rest.



Fonte: Próprios autores.

A estrutura do backend foi dividida em três principais componentes:

Models: Esta pasta contém os schemas e atributos dos objetos utilizados na aplicação. Os modelos definem a estrutura dos dados e suas relações no banco de dados.

- **Controllers:** Aqui reside a parte lógica responsável pela manipulação dos modelos de dados. Foram criadas duas subpastas neste diretório:
- **Auth:** Contém os controladores responsáveis pela autenticação dos usuários do sistema, incluindo funções como registro, login e recuperação de dados.
- **Experiment:** Engloba os controladores relacionados aos experimentos, como criação, manipulação de dados e geração de gráficos.
- **Database:** Este diretório trata da conexão com o banco de dados. O arquivo `conn.js` é responsável por tentativas de conexão com o MongoDB a partir de uma string armazenada no arquivo `.env`.

Arquitetura de projeto

A arquitetura de pastas do projeto foi organizada de acordo com o padrão MVC (Model-View-Controller), garantindo uma estrutura clara e modular. As pastas foram divididas em modelos, controladores, rotas, serviços, contextos, componentes, páginas e utilitários, cada uma com sua responsabilidade específica. Isso facilitou a manutenção, escalabilidade e extensibilidade do código ao longo do desenvolvimento do projeto, (Richardson, L., & Amundsen, M.).

A estrutura do backend foi dividida em três principais componentes:

- **Models:** Esta pasta contém os schemas e atributos dos objetos utilizados na aplicação. Os modelos definem a estrutura dos dados e suas relações no banco de dados.
- **Controllers:** Aqui reside a parte lógica responsável pela manipulação dos modelos de dados. Foram criadas duas subpastas neste diretório:
- **Auth:** Contém os controladores responsáveis pela autenticação dos usuários do sistema, incluindo funções como registro, login e recuperação de dados.
- **Experiment:** Engloba os controladores relacionados aos experimentos, como criação, manipulação de dados e geração de gráficos.
- **Database:** Este diretório trata da conexão com o banco de dados. O arquivo `conn.js` é responsável por tentativas de conexão com o MongoDB a partir de uma string armazenada no arquivo `.env`.

Organização da Estrutura Front-end

A estrutura de pastas do front-end foi meticulosamente organizada para otimizar o desenvolvimento e a manutenção do projeto. O diretório `src` representa o arquivo raiz que abriga todo o conteúdo bruto do projeto front-end, enquanto os arquivos fora dele são os padrões, incluindo o `index.js`.

1. Componentes Reutilizáveis

Na pasta `components`, encontram-se todos os componentes reutilizáveis do código. Por exemplo, o `BaseAuth` é um componente que contém o estilo base da página de autenticação do usuário, como a configuração da tela de fundo e do card, além de uma área para inserção de conteúdo. Cada componente é desenvolvido com um arquivo `index.jsx`, que contém o código JavaScript do React, e seu próprio arquivo de estilo, `styles.css`.

2. Contextos e Providers

A pasta context abriga todos os contextos e provedores utilizados na aplicação. Dentre eles, destaca-se o contexto de Users, que engloba informações relevantes ao professor, facilitando sua reutilização em toda a plataforma. Além disso, há o contexto de Authorization, que gerencia o estado booleano do botão responsável por liberar o resultado do experimento aos alunos.

3. Rotas de Navegação

As rotas de navegação são organizadas na pasta routes, criadas a partir das páginas destinadas à navegação dos usuários. Para uma melhor organização, foram criadas duas novas subpastas: StudentPages e TeacherPages, que contêm os arquivos JSX e CSS de cada página, separando claramente aquelas acessadas pelo usuário comum e pelo professor.

4. Páginas do Usuário

A pasta pages engloba todos os arquivos criados para a construção das páginas de navegação do usuário. A fim de separar adequadamente os arquivos das páginas destinadas aos usuários comuns e aos professores, foram criadas as pastas StudentPages e TeacherPages. Dentro delas, estão os arquivos JSX e CSS de cada página individualmente, garantindo uma organização clara e intuitiva da estrutura.

Detalhes Adicionais

Pastas Result Tables: Esta pasta contém arquivos com arrays extraídos de planilhas Excel que representam gráficos de queda de água corporal. Os dados foram divididos em arquivos diferentes para melhor organização.

Arquivo .env: Local onde são armazenadas as configurações sensíveis da aplicação, como a string de conexão do banco de dados. Por questões de segurança, este arquivo não é incluído no repositório Git.

Arquivo .gitignore: Responsável por excluir do controle de versão arquivos que não devem ser compartilhados no repositório Git, como node_modules, yarn_lock e .env.

Imagens da Plataforma e operação

Nessa seção são apresentadas as etapas de operação da plataforma. A tela inicial da plataforma é apresentada na figura 2.

Figura 2. Tela Inicial da plataforma.



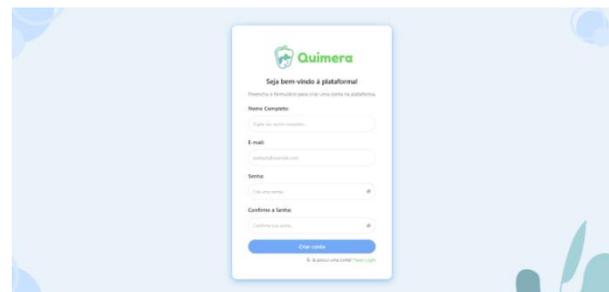
Fonte: Próprios autores.

A seguir a operação do sistema é dividida em onze etapas.

Etapa 1

O acesso do professor permite que este crie um cadastro com seus dados na plataforma (Fig.3).

Figura 3. Cadastro do professor.



Fonte: Próprios autores.

Etapa 2

O professor realiza login na plataforma com seus dados informados anteriormente (Fig.4).

Figura 4. Login do professor.



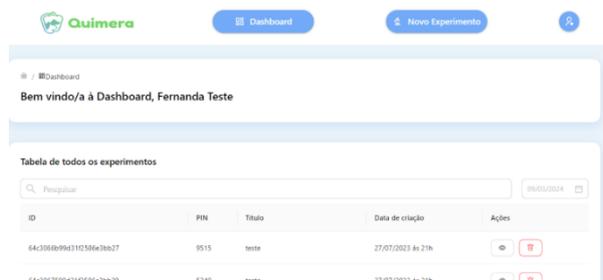
Fonte: Próprios autores.

Etapa 3

O professor é redirecionado após o login para a Dashboard (Página principal), onde ele pode

visualizar experimentos anteriores na tabela ou ir para a página de criar novos experimentos (Fig. 5).

Figura 5. Dashboard, onde o professor visualiza o histórico de experimentos.

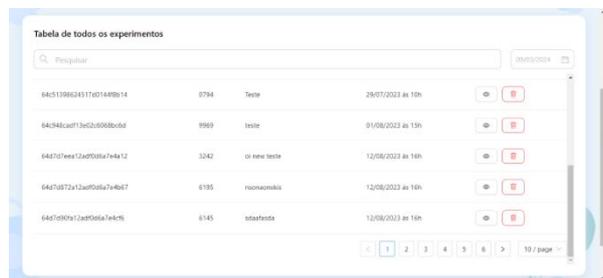


Fonte: Próprios autores.

Etapa 4

A tabela de visualização de todos os experimentos permite filtrar por data, nome, excluir experimento e visualizar experimento (Fig. 6).

Figura 6. Tabela dos experimentos com filtros de busca.

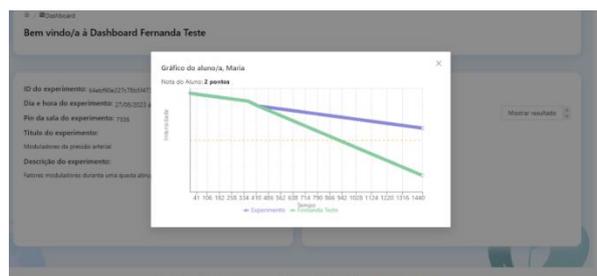


Fonte: Próprios autores.

Etapa 5

Professor consegue visualizar informações sobre a sala de experimento e uma lista com os nomes de todos os alunos que participaram e realizaram o experimento. Ao clicar em 'Mostrar resultado' é possível ver o resultado do aluno em gráfico e nota (Fig.7).

Figura 7. Resultado do aluno em gráfico e nota.

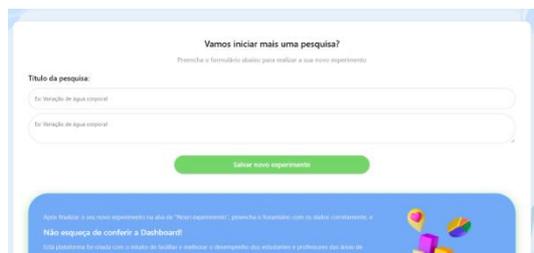


Fonte: Próprios autores.

Etapa 6

Em 'Novo experimento', o Professor é redirecionado para a página que cria novos experimentos. Bastando preencher apenas o Título e a descrição (Fig. 8).

Figura 8. Seção de criação de novos experimentos.



Fonte: Próprios autores.

Etapa 7

Após criar um novo experimento o professor é levado para a sala do mesmo, onde lhe é entregue um pin randômico para os outros alunos poderem acessar o experimento (Fig.9).

Figura 9. Geração de um código para o login dos estudantes em uma nova atividade.

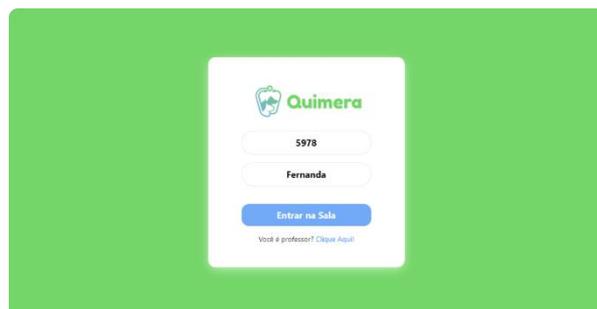


Fonte: Próprios autores.

Etapa 8

Aluno entra na sala com o pin fornecido pelo professor e seu nome (Fig. 10).

Figura 10. Acesso dos estudantes à atividade gerada pelo professor.



Fonte: Próprios autores.

Etapa 9

Aluno acessa a sala de introdução a matéria na qual ele realiza o experimento e escolhe entre estudar o caso clínico ou iniciar o experimento propriamente (Fig. 11).

Figura 11. Acesso dos estudantes à introdução da atividade.

• **PERDA:** Ocorre na forma de Urina + Sudorese + Respiração + Fezes + Leite.

Tanto o ganho como a perda de água do organismo animal são contínuos, com episódios mais significativos e outros menos, por exemplo: ao urinar o animal elimina quantidade significativa de água pontualmente, por outro lado a perda via respiração não é pontual, ao invés, ocorre continuamente, variando segundo vários fatores ambientais. O ganho ocorre da mesma forma, com ingestão de água in natura em quantidade significativa pontualmente, e a metabólica não tão pontual.

Quando a perda atinge níveis significativos mecanismos para a ingestão são desencadeados, justamente pela sensação de sede, porém outros mecanismos concomitantes também serão promovidos, por exemplo, com a concentração de urina sendo acentuada, portanto diminuindo o volume de água eliminado. Essa situação será contínua para qualquer animal, com o desafio da **DESIDRATAÇÃO** ameaçando a vida permanentemente.

Assim que a desidratação tem início o animal irá procurar água para beber, isso varia em função de vários aspectos, como: espécie, estado fisiológico desse indivíduo, idade, temperatura e umidade ambiente, etc.

Outro ponto importante: relacionado com a perda de água, é a alteração dos eletrólitos presentes no organismo animal, porém esse tema será motivo para outra atividade. Os mecanismos referentes ao controle dos eletrólitos possuem íntima relação com o controle de água.

DESEJA ESTUDAR O CASO CLÍNICO ANTES DE INICIAR OU PULAR PARA O EXPERIMENTO?

Fonte: Próprios autores.

Etapa 10

Aluno estuda o caso clínico do animal antes de iniciar o experimento (Fig. 12).

Figura 12. Caso clínico da atividade.

Bem vindo/a ao caso clínico do experimento: 5978, Fernanda

Animal: Quimera I

Explicação: Problemas renais são muito comuns em gatos. Nessa condição um sintoma muito importante que devemos monitorar será a desidratação. Quando da manutenção do animal em clínicas veterinárias, para procedimentos diversos, frequentemente observamos essa perigosa situação.

Sintomas

Os principais sintomas da desidratação nesse animal são:

- Pele enrugada
- Respiração ofegante
- Aumento da frequência cardíaca
- Olhos fundos
- Dificuldade em urinar

Causas

Causes de desidratação:



Fonte: Próprios autores.

Etapa 11

Após o estudo realizado pelo aluno ele é redirecionado para a página do experimento, onde ele deve escolher os dois hormônios corretos que irá solucionar o caso do animal. Caso a escolha não seja correta, o gráfico mostrará a consequência para a saúde do animal. O resultado final é apresentado apenas após o professor liberar o acesso, a partir de quando o aluno conseguirá visualizar o resultado, além de sua nota final do experimento (Fig. 13).

Figura 13. Indicação gráfica do resultado correto comparada à escolha do estudante.



Fonte: Próprios autores.

Conclusões e Perspectivas Futuras

A plataforma Quimera representa uma abordagem inovadora e ética para o ensino de fisiologia animal, oferecendo uma alternativa sustentável ao uso de animais vivos em ambientes educacionais. Ao envolver os estudantes de forma ativa e adaptar o conteúdo às suas necessidades individuais, espera-se alcançar resultados educacionais significativos.

Referências

NORONHA, D. X. de; SILVA, G. da; SOARES, V. C. EscapeLab: an escape room for teaching Chemistry. Research, Society and Development, [S. l.], v. 9, n. 11, p. e98691110511, 2020. DOI: 10.33448/rsd-v9i11.10511. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/10511>. Acesso em: maio 2024.

HARDER, B. Nicole. Use of Simulation in Teaching and Learning in Health Sciences: a systematic review. **J. Nurs. Educ.**, v. 49, n. 1, p. 23-28, 2010. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/19731886/>. Acesso em: maio de 2024.

GAMA, N.; CAMPOS, M. L. M. Understanding Model-View-Controller (MVC) Architecture in ASP.NET Core. In: Advanced Applications of MVC Framework Using ASP.NET Core. IGI Global, 2018. p. 1-16. Disponível em: <https://learn.microsoft.com/pt-br/aspnet/core/mvc/overview?view=aspnetcore-8.0>. Acesso em: março de 2023.

COOKSEY, B. RESTful Web APIs: Services for a Changing World. O'Reilly Media, Inc., 2013. Disponível em: <https://zapier.com/resources/guides/apis/design>. Acesso em: março de 2023.

Documentação do MongoDB. NoSQL, mongodb:
<https://www.mongodb.com/document-databases>.
Acesso em: março de 2023.