



FACULDADE PROJEÇÃO

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Lucas da Silva Barbosa

SISTEMA ATHENA: SISTEMA DE QUIZ DE PERGUNTAS DE TI

Brasília, novembro de 2022



FACULDADE PROJEÇÃO

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

SISTEMA ATHENA: SISTEMA DE QUIZ DE PERGUNTAS DE TI

Projeto do Curso de Análise e Desenvolvimento de Sistemas da Faculdade Projeção, como requisito complementar na disciplina de projeto integrado e obrigatório à obtenção de aprovação na disciplina.

Brasília, 25 de novembro de 2022.

Profa. M.a Ivonete Ferreira de Sousa
Orientadora

AGRADECIMENTOS

A Deus por ter me dado resiliência para manter a fé e perseverança para me reinventar diante das adversidades.

A esta instituição, seu corpo docente, direção e administração que viabilizaram a minha formação enquanto Analista de Sistemas e Desenvolvedor.

À minha professora-orientadora Ivonete Ferreira de Sousa, pelo suporte durante toda a disciplina que envolveu esse projeto.

Aos meus pais, pelo amor e apoio incondicional.

E a todos que fizeram parte da minha formação, seja com contribuições diretas ou indiretas.

SUMÁRIO

RESUMO	5
CAPÍTULO I	6
VISÃO INICIAL.....	6
1. INTRODUÇÃO	6
2. CONTEXTUALIZAÇÃO	6
3. PROBLEMÁTICA	7
4. SOLUÇÃO PROPOSTA	8
5. O QUE SE ESPERAR.....	10
6. OBJETIVO GERAL.....	10
6.1 OBJETIVOS ESPECÍFICOS	11
7. ESTRUTURA DO PROJETO	11
CAPÍTULO II	12
MODELAGEM DE DADOS	12
2.1. Diagrama de Entidade Relacional	12
2.2. Modelo de Entidade-Relacional.....	14
2.3 Dicionário de Dados.....	16
2.3.1 Tabela <i>bancas</i>	17
2.3.3 Tabela <i>log_acessos</i>	19
2.3.4 Tabela <i>migrations</i>	20
2.3.5 Tabela <i>opcoes</i>	21
2.3.6 Tabela <i>perguntas</i>	22
2.3.7 Tabela <i>provas</i>	24
2.3.8 Tabela <i>tentativas</i>	25
2.3.9 Tabela <i>usuarios</i>	27
CAPÍTULO III	29
3.1. Conjunto de Tabelas	30
3.2. Diagrama de Caso de Uso	35
3.3. Conjunto de Descrições de Caso de Uso	36
3.3.1 Realizar Cadastro	36
3.3.2 Realizar Login	38
3.3.3 Responder Pergunta	40
3.3.4 Manter Pergunta	42
CONCLUSÃO.....	44
BIBLIOGRAFIA.....	46

RESUMO

Muitas instituições de ensino observaram como a gamificação poderia ser uma aliada no que se trata do aprendizado de seus estudantes especializados em diversas áreas do conhecimento, o que inclui a área de Tecnologia da Informação (TI). A gamificação pode ser utilizada pelas instituições de ensino por meios digitais, incrementando à iniciativa do ensino à distância. Este documento visa relatar todo o processo de prototipação de um sistema Web que oferecerá as funcionalidades de um quiz de perguntas. A preocupação do sistema está no aprendizado que as pessoas, principalmente os estudantes de informática, podem adquirir ao utilizar um sistema que torne a experiência de responder perguntas de tecnologia da informação satisfatória e gamificada, tomando o formato de um quiz.

Palavras-Chave: Quiz, Perguntas de Informática, Gamificação, Aprendizado de Tecnologia

CAPÍTULO I

VISÃO INICIAL

O projeto “AthenaWeb”, a ser desenvolvido por mim, Lucas da Silva Barbosa, visa, além de exercitar os conhecimentos adquiridos ao longo da minha atividade como estudante na Faculdade Projeção como aluno de Análise e Desenvolvimento de Sistemas, prototipar funcionalidades de uma aplicação web que não se sustentará somente no conhecimento adquirido durante as disciplinas ministradas. Um dos elementos que constituem um passo além é, por exemplo, a administração de serviços web do lado do back-end por meio de requisições HTTP, utilizando o estilo de arquitetura REST.

1. INTRODUÇÃO

Hoje em dia, profissionais da educação andam preocupados em como podem exercer melhor seu ofício para que os alunos possam cada vez mais serem engajados com o estudo, tendo como objetivo a aquisição de mais conhecimento, e também tendo como demonstrar esse conhecimento na prática. O mercado de tecnologia da informação cada vez mais demanda profissionais da área capazes de solucionar os problemas atuais com a abordagem mais apropriada, se adaptando às mudanças e aos riscos.

Uma vez expondo ambos os contextos, tanto dos profissionais da educação quanto do mercado de tecnologia, percebe-se uma relação entre o mercado de tecnologia que precisa capacitar pessoas para atender demandas de forma mais eficiente. Portanto, a formação e o estudo dessas pessoas devem adaptar-se ao cenário atual com a finalidade de que elas possam prover soluções tecnológicas apropriadas para determinadas circunstâncias.

2. CONTEXTUALIZAÇÃO

Já existem algumas soluções e ferramentas, como o sistema web Socrative, que auxiliam no processo educacional e formativo, contribuindo para a absorção de conhecimento, tanto prático quanto teórico. Ou seja, é notado que, em oposição ao ensino tradicional, que dá pouca ênfase no desenvolvimento de competências, o uso de softwares de aprendizado, como o Socrative, associado a metodologias ativas e

inovadoras de aprendizagem, pode atingir resultados que ultrapassam aquilo que o ensino tradicional proporciona aos estudantes. (OLIVEIRA; FREITAS; LIMA; ARRUDA; JAIME; MOURA, 2019).

Há evidências de que a metodologia ativa aliada à gamificação proporciona resultados satisfatórios se comparados ao ensino tradicional. Dentre vários fatores, a falta de motivação com a finalidade de aprender os conteúdos ministrados no ambiente escolar mostra-se o mais impactante quando se trata de deficiências no ensino. O ensino tradicional possui como modelo, a estrutura passivo/expositivo de aprendizagem, ou seja, o professor expõe e explica um conteúdo com mínima ou nenhuma participação do aluno. (SILVA; GILVANDENYS; JUSCILEIDE, 2019).

A gamificação, que foi supracitada ao lado da metodologia ativa, não precisa ser restrita aos ambientes de aprendizagem. É uma abordagem que agrega muito também em ambientes de trabalho com o elemento de competitividade, tendo a finalidade de estimular motivação e comprometimento. Com esses estímulos, é possível visualizar um melhor desempenho daqueles que estão envolvidos diretamente no processo de produção de uma instituição. (MIRI; SANTOS; MATTE; CHAIS; OLEA, 2021).

3. PROBLEMÁTICA

Conforme discorrido acima sobre os temas de metodologia ativa, gamificação e a demanda de profissionais de Tecnologia da Informação (TI) no mercado de trabalho, a problemática reside no fato de que, enquanto o mercado demanda profissionais de TI capacitados para produzirem com eficiência e entregarem produtos disruptivos e inovadores, esses mesmos profissionais não estão sendo formados no ritmo que o mercado desejaria que fossem. Ou seja, há uma ausência de um modelo de ensino estabelecido que siga o mesmo ritmo das demandas do mercado de trabalho de tecnologia da informação, pois, caso esse modelo existisse, essa falta de profissionais provavelmente não existiria.

As instituições de ensino têm a proatividade de procurar iniciativas para abordar essa questão, como a metodologia ativa, a gamificação, a digitalização do ensino com as plataformas EAD (ensino à distância), dentre várias abordagens. Entretanto, falamos de um problema que possui uma complexidade considerável, o que cria uma

difficuldade para que somente as instituições de ensino consigam endereçar essa problemática de forma imediata e pontual, apesar de seu alto empenho. Considerando o contexto, este projeto tem o intuito de responder a seguinte problemática: Qual seria a abordagem apropriada que aproximaria a capacitação profissional e o estudo de tecnologia ao ritmo do mercado, possuindo foco em todas as pessoas envolvidas no processo tanto de formação profissional quanto de aquisição de profissionais na área de tecnologia?

4. SOLUÇÃO PROPOSTA

A solução para abordar a condição atual supracitada constitui na construção de um sistema web. Este sistema web será um quiz, o qual o usuário poderá entrar com sua conta e responder perguntas da área de tecnologia da informação. Esse sistema fará com que o usuário exercite seus conhecimentos em TI de forma prática, o que é um ganho em usabilidade.

Esse tipo de solução é útil e vai na mesma direção onde vão as instituições de ensino, o qual desempenham um alto esforço para endereçar a atual situação da capacitação dos profissionais de TI. Evidentemente, por se tratar de um sistema web, ela acaba conversando com a modalidade de ensino EAD, já que não há qualquer abordagem presencial proposta, e também há o componente de gamificação, já que se trata de um quiz onde as respostas são certas ou erradas.

As ferramentas que serão utilizadas no desenvolvimento do projeto incluem linguagens de programação, banco de dados, ambientes de desenvolvimento integrado, entre outros.

Quanto ao ambiente de desenvolvimento integrado utilizado, haverá predominância do Visual Studio Code. Caso seja necessário a mudança de IDE, a mudança será feita para o NetBeans. Tanto o Visual Studio Code quanto o NetBeans são detentores de diversos plugins e ferramentas que auxiliam o desenvolvedor a escrever melhor o código e diferenciar partes dele, como diferenciar um atributo de um método no caso de uma linguagem orientada à objetos, ou uma variável de uma função no caso de uma linguagem funcional e/ou estruturada.

Quanto às linguagens utilizadas para o desenvolvimento de sistemas, estas serão HyperText Mark-up Language, Cascading Styling Sheets, JavaScript, PHP:

HyperText Preprocessor e Structured Query Language. São linguagens padrão para o desenvolvimento de aplicações web, inclusive utilizadas no mercado.

Quanto aos frameworks utilizados, estes serão Bootstrap, Angular e Laravel. Essas abstrações que possuem pacotes de código são relevantes e muito utilizadas no mercado, cada uma adicionando aos projetos de sua própria forma, quando utilizados.

Quanto às ferramentas para diagramação UML e documentação de software, será utilizado predominantemente o Astah, porém caso seja necessário a mudança de ferramenta de documentação e diagramação, será feita a mudança para o Draw.io, ou LucidChart e/ou Microsoft Office. O Astah, que é uma ferramenta respeitada no mercado, produz diagramas UML com praticidade. O Draw.io e o LucidChart também produzem diagramas, porém já não tem a mesma posição do Astah enquanto software profissional. O Microsoft Office, principalmente o Word, é também uma ferramenta bem conhecida para criação e edição de documentos e, quando um software deve ser documentado na intenção de gerar documentos que não são diagramas, um editor de texto como o Word é bem-vindo.

Quanto ao banco de dados que será utilizado, trata-se do MySQL. O banco de dados MySQL é bem popular, ultrapassando 10 milhões de instalações pelo mundo e é mantido pela Oracle Corporation.

Quanto à ferramenta para modelar o banco de dados, será utilizado o BRModeloWeb para modelagem conceitual e o MySQL Workbench para modelagem física. O BRModeloWeb é uma solução web que traz muita praticidade ao usuário, uma vez que é hospedado na web, e cumpre uma boa parte das funcionalidades do BRModelo. Já o MySQL Workbench, vindo junto da instalação do MySQL, também compartilha dos mesmos status do banco de dados MySQL, porém referentes à modelagem física de um banco de dados.

Para apoiar o desenvolvimento do sistema, serão utilizados Postman e Git. O Postman é um software que oferece um ambiente para desenvolvedores projetarem, construírem e testarem suas APIs. O Git é um sistema gerenciador de versão para auxiliar o desenvolvedor a controlar as mudanças do software em relação tanto a requisitos funcionais quanto não funcionais.

Para edição de imagem, será utilizado o GIMP. O GIMP é um software de código aberto voltado principalmente para a edição de imagens, opção interessante e robusta para aqueles que não desejam investir em um software de edição de imagem como o Adobe Photoshop, por exemplo.

Para uso de imagens de uso livre, essas serão consumidas dos sites Unsplash e Pixabay. Nem toda a imagem pode ser utilizada em um sistema web. As imagens utilizadas devem possuir licença de uso, e é em sites como os previamente citados que podemos ter essa segurança no uso de imagens.

5. O QUE SE ESPERAR

Espera-se alcançar até o final deste trabalho, da disciplina de Projeto Integrado para Desenvolvimento Web, a demonstração dos conhecimentos obtidos durante a minha jornada enquanto estudante no UniProjeção. Essa demonstração será exibida como um protótipo de um sistema web, o qual utilizará as ferramentas expostas no tópico de Solução Proposta.

A expectativa é desenvolver e ganhar experiência no uso de tecnologias e arquiteturas que vão além do escopo abordado no período a faculdade. Como exemplo, informo que estou atuando atualmente como pesquisador no Laboratório LATITUDE no projeto NSGC (Novo Sistema de Gestão Corporativa), ao qual são utilizadas tecnologias ainda não abordadas na faculdade, porém que já são utilizadas no mercado de tecnologia.

Portanto, o projeto atinge sua meta a partir do momento que um protótipo funcional é apresentado contendo as tecnologias e ferramentas citadas, pois requer uma grande carga de estudo para se compreender plenamente sobre API's, requisições HTTP envolvendo Angular e Laravel e criação dinâmica de formulários no framework Angular.

6. OBJETIVO GERAL

Este sistema trata-se de um quiz de perguntas de TI, no qual o usuário deve se cadastrar para acessar o sistema. Uma vez cadastrado e autenticado, pode responder às perguntas.

6.1 OBJETIVOS ESPECÍFICOS

Tendo como perspectiva e contexto o objetivo geral, têm-se os seguintes objetivos específicos, que se traduzem como requisitos funcionais:

- Manter Cadastro
- Manter Login
- Responder Pergunta
- Manter Pergunta.

7. ESTRUTURA DO PROJETO

Este projeto foi dividido em três capítulos, onde será apresentado, de forma objetiva, o desenvolvimento de todo o projeto integrado de desenvolvimento web.

- O Capítulo I trata de expor uma visão inicial do contexto da aplicação, englobando seu contexto, a problemática que quer abordar, como se traduz em uma solução, e como, de fato, essa solução se concretizará no sistema por meio dos objetivos que deseja atingir;

- O Capítulo II especificará a modelagem, tanto física quanto conceitual, do banco de dados MySQL, que é um banco de dados relacional. A modelagem levará em conta o MER (Modelo Entidade Relacionamento), o DER (Diagrama Entidade-Relacional) e o Dicionário de Dados.

- O Capítulo III detalhará o Documento de Visão. O documento de visão consistirá na documentação do sistema, que será feita baseada nos conhecimentos de Engenharia de Software. Poderá incluir layout de telas, casos de uso, entre tantas coisas;

- Ao final haverá inclusão da conclusão e das referências bibliográficas. No Capítulo 1, houve necessidade de parafrasear alguns autores de publicações científicas, portanto as referências bibliográficas já começaram a ser desenvolvidas no Capítulo 1.

CAPÍTULO II

MODELAGEM DE DADOS

A modelagem de dados, no contexto de Banco de Dados, visa descrever formalmente a estrutura de um banco de dados, com finalidade de promover uma conexão entre aspectos altamente técnicos e computacionais com o fenômeno da realidade que visa ser modelado e abstraído. A modelagem de dados compreenderá o Diagrama de Entidade Relacional, o Modelo de Entidade Relacional e o Dicionário de Dados.

O Diagrama de Entidade Relacional é utilizado para descrever as entidades envolvidas em um domínio de negócios, com seus atributos e como elas se relacionam entre si. A forma como se dá o modelo é abstrata, sendo que esse modelo visará sua implementação em um banco de dados relacional. Uma de suas grandes vantagens é a independência de SGBD.

Já o Modelo de Entidade-Relacional possuirá as tabelas de banco de dados, seus respectivos campos a serem preenchidos quando um registro for inserido nas mesmas, suas respectivas ligações entre si mesmas, suas chaves primárias e estrangeiras com suas respectivas “constraints”, etc. Percebe-se que não se fala mais diretamente em entidades e atributos, mas sim em tabelas, chaves primárias e estrangeiras e campos, pois já estamos nos tratando de um modelo que está mais perto da solução computacional do que da solução de negócio.

2.1. Diagrama de Entidade Relacional

Na página seguinte, consta o Diagrama de Entidade Relacional, produzido a partir da ferramenta BR Modelo Web. Neste diagrama, percebe a exposição das entidades que definem o problema que se pretende ser resolvido com maior atenção ao domínio do que da implementação técnica, já que é um diagrama que independe de SGBD.

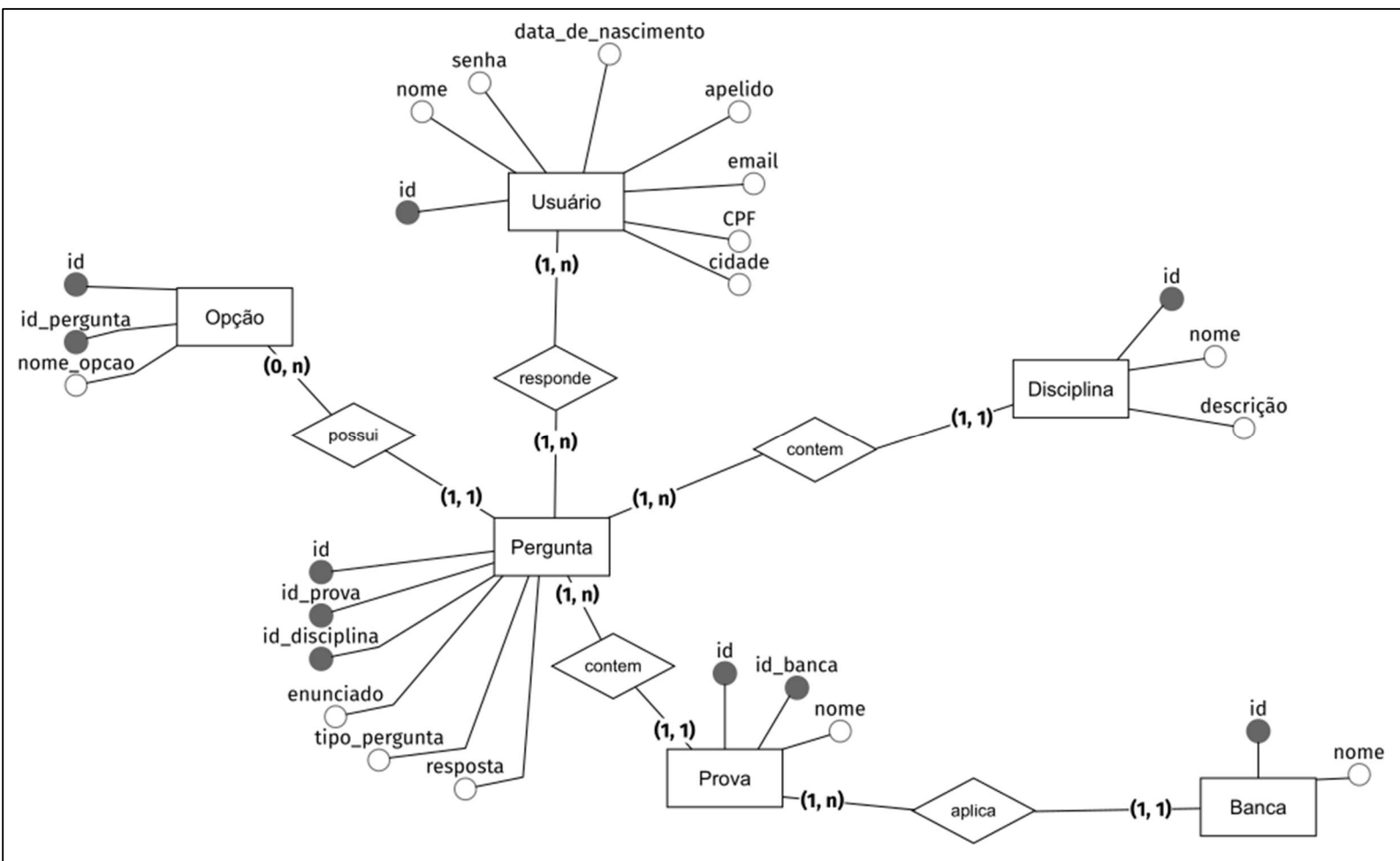


Figura 01 – Diagrama de Entidade Relacional
Fonte: BARBOSA, 2022.

2.2. Modelo de Entidade-Relacional

Na próxima página, é apresentado o Modelo de Entidade-Relacional, produzido a partir da ferramenta MySQL Workbench. Neste modelo, percebe-se a adição de três tabelas, uma referente aos acessos feitos às rotas do sistema (`log_acessos`), uma referente à execução do recurso de Migrations do framework Laravel (`migrations`) e uma referente à implementação do relacionamento entre Usuários e Perguntas (tentativas). Estas tabelas não foram declaradas como entidades no diagrama anterior, pois são pertinentes uma vez se tratando do banco de dados com a perspectiva da solução computacional manifestada na implementação de códigos. As outras tabelas foram representadas no diagrama passado como entidades pois possuíam valor de domínio para a explicação de quais seriam as entidades, seus atributos e seus relacionamentos.

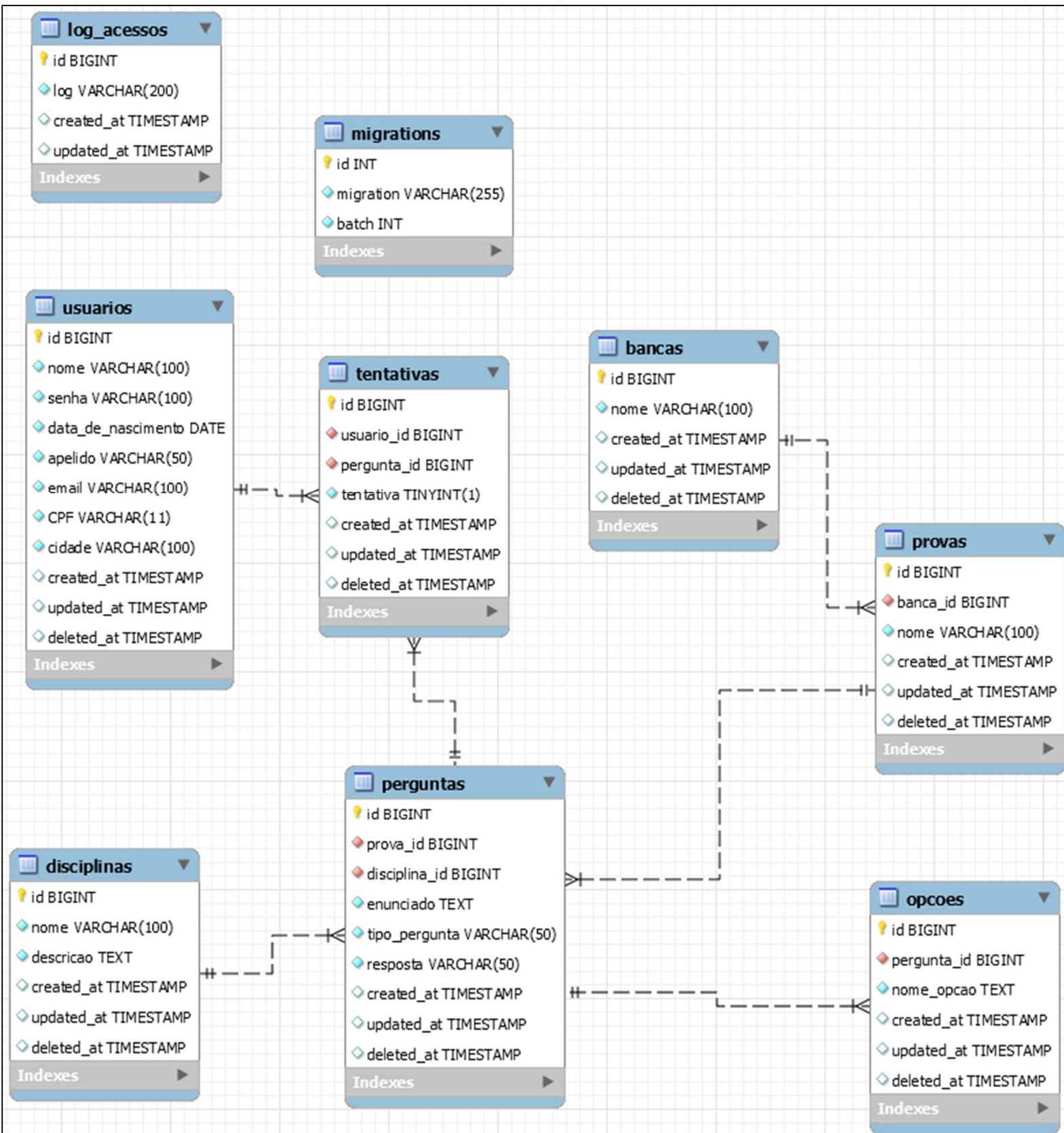


Figura 02– Diagrama de Banco de Dados: MER
Fonte: BARBOSA, 2022.

2.3 Dicionário de Dados

É importante que todo o banco de dados possua seu respectivo dicionário de dados. Este mesmo dicionário de dados consiste numa coleção de metadados, que são dados que descrevem os dados abordados em um contexto de banco de dados (dados sobre dados). Esses mesmos metadados contém definições do que são esses dados e pode conter representações sobre os elementos dos dados.

Quanto aos metadados que irão descrever os dados do Sistema Athena Web, é importante lembrar o uso do framework Laravel, que se utiliza da linguagem de programação PHP. As tabelas não foram criadas via script SQL, e sim via execução de código PHP (Migrations e seeders). O propósito disto é a utilização do recurso Eloquent ORM, um mapeador relacional-objeto que, além de automatizar muitos comandos de bancos de dados (DDL, DML, entre tantos tipos), se relaciona bem com a arquitetura MVC, padrão rigorosamente seguido pelo framework.

Um dos impactos da criação das tabelas com Eloquent ORM via execução de Migrations impacta que, ao final de todas as tabelas possui a letra “s”, inferindo a ideia de plural, uma relação de registros. Outro é a inserção das colunas “created_at”, “updated_at” e “deleted_at” (traduzindo para o português: criado em, atualizado em, deletado em).

Uma grande adição com o uso do framework Laravel é a permissão de exclusão lógica ao invés de exclusão física. Ao invés de um registro ser apagado de fato do banco de dados (exclusão física), quando um registro for deletado, a coluna é preenchida com a data da exclusão, mantendo o registro no banco de dados e fazendo com que o registro se comporte como um registro não existente. Isso torna mais fácil a reversão de exclusões indesejadas.

2.3.1 Tabela *bancas*

Esta tabela visa armazenar dados sobre as bancas examinadoras (CESPE, FGV, etc.) que são responsáveis pela aplicação de provas contendo conteúdos de Tecnologia da Informação.

Tabela 1 – Tabela “bancas”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da banca (chave primaria).
nome	VARCHAR(100)	S	Este é o campo que guardará o nome da banca
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa banca. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa banca. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa banca. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
provas	A tabela “bancas” possui relacionamento um para muitos com a tabela “provas”, ou seja, uma banca pode produzir e aplicar várias provas, porém uma prova só pode ser aplicada por uma banca. O relacionamento é efetivado pela presença da chave estrangeira “banca_id” na tabela “provas”.		

Fonte: BARBOSA, 2022.

2.3.2 Tabela *disciplinas*

Esta tabela visa armazenar dados sobre as disciplinas que contemplam as perguntas de Tecnologia da Informação. Afinal, a Tecnologia da Informação abrange muitas disciplinas como Engenharia de Software, Rede de Computadores, Arquitetura de Computadores, etc.

Tabela 2 – Tabela “disciplinas”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da disciplina (chave primaria).
nome	VARCHAR(100)	S	Este é o campo que guardará o nome da disciplina
descricao	TEXT	S	Este é o campo que guardará uma breve descrição sobre a disciplina
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa disciplina. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa disciplina. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa disciplina. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
perguntas	A tabela “disciplinas” possui relacionamento um para muitos com a tabela “perguntas”, ou seja, uma disciplina pode conter várias perguntas, porém uma pergunta, no contexto do sistema, só pode estar contida em uma disciplina. O relacionamento é efetivado pela presença da chave estrangeira “disciplina_id” na tabela “perguntas”.		

Fonte: BARBOSA, 2022.

2.3.3 Tabela *log_acessos*

Esta tabela visa armazenar log de acessos das rotas da API, portanto, não possuiria utilidade a presença de um recurso de exclusão lógica, já que o recurso de log tem utilidade para verificar o histórico de acesso de uma aplicação. A utilidade da coluna “updated_at” é a de sempre sua data ser igual à da coluna “created_at”. Caso haja diferença, isto poderá implicar em inconsistência nos registros da tabela.

Tabela 3 – Tabela “log_acessos”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único do log de acesso (chave primaria).
log	VARCHAR(100)	S	Este é o campo que guardará o log de acesso. Conterá a rota acessada, o IP, e o método que foi acessado naquela rota.
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente ao log. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente ao log. Coluna inserida pelo framework Laravel.

Fonte: BARBOSA, 2022.

2.3.4 Tabela *migrations*

Esta tabela visa armazenar dados sobre o andamento da execução das “migrations”, que são os scripts do framework Laravel dedicados a manipular o banco de dados, seja em sua definição (DDL), quanto manipulação (DML). Esta é uma tabela puramente dedicada ao desenvolvedor de sistemas, portanto não é pertinente que ela inclua as colunas “created_at”, “updated_at” e “deleted_at”, além de ser uma tabela automaticamente criada pelo framework Laravel ao se executar a primeira migration e mantida na execução de migrations futuras.

Tabela 4 – Tabela “migrations”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da migration (chave primaria).
migration	VARCHAR/255	S	Este é o campo que guardará o nome da migration executada pelo framework.
batch	INT	S	Este é o campo que guardará o pacote(batch) em que a migration foi executada. Se a migration foi executada na primeira vez que se deu seu respectivo comando, seu valor será 1. Caso não seja a primeira vez, será um valor maior do que 1.

Fonte: BARBOSA, 2022.

2.3.5 Tabela *opcoes*

Esta tabela visa armazenar dados sobre as opções que as provas detêm. Caso as perguntas sejam do tipo “Verdadeiro ou Falso”, não é criado registro nessa tabela para associar à pergunta, já que suas opções são somente “Verdadeiro” ou “Falso”, e essa resposta se traduzirá somente no preenchimento do campo “resposta” da tabela “perguntas”.

Toda a vez que um registro da tabela “opcoes” se relacionar com um registro da tabela “perguntas” por meio da chave estrangeira “pergunta_id” contida na tabela “opcoes”, o campo “tipo_pergunta” na tabela “perguntas” deve conter o valor “Múltipla Escolha”.

Tabela 5 – Tabela “opcoes”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da opção (chave primaria).
pergunta_id	BIGINT	S	Este é o campo que guardará o identificador único da pergunta (chave estrangeira).
nome_opcao	TEXT	S	Este é o campo que guardará o nome da opção (item) de uma pergunta.
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa opção. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa opção. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa opção. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
perguntas	A tabela “opcoes” possui relacionamento muitos para um com a tabela “perguntas”, ou seja, uma opção pertence à uma pergunta, porém uma pergunta, se for de múltipla escolha, pode ter 5 opções. O relacionamento é efetivado pela presença da chave estrangeira “pergunta_id” na tabela “opcoes”.		

Fonte: BARBOSA, 2022.

2.3.6 Tabela *perguntas*

Esta tabela visa armazenar dados sobre as perguntas do sistema de quiz. O sistema foi pensado para possuir dois tipos de questão: “Múltipla Escolha” (Uma pergunta em que, em resposta ao enunciado, possui cinco opções, sendo somente um deles verdadeiro) e “Verdadeiro ou Falso” (Uma pergunta em que o enunciado possui valor verdadeiro ou valor falso). Se for múltipla escolha, possuirá uma sequência de cinco números no campo resposta, dentre eles quatro números zeros para as opções falsas e um número um para a opção verdadeira, além de suas respectivas opções na tabela *opcoes*. Se for “Verdadeiro ou Falso”, somente possuirá, no campo resposta, o número 0 ou 1, sendo 0 para falso e 1 para verdadeiro.

Um exemplo de preenchimento do campo resposta, no caso do tipo da pergunta ser de múltipla escolha, seria a cadeia de caracteres “00100”. O significado seria que somente o terceiro item está correto. Já um exemplo de preenchimento do campo resposta, no caso do tipo da pergunta ser de verdadeiro ou falso, seria somente um caracter “0”, para falso, e “1”, para verdadeiro.

Tabela 6 – Tabela “perguntas”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da pergunta (chave primaria).
prova_id	BIGINT	S	Este é o campo que guardará o identificador único da prova (chave estrangeira).
disciplina_id	BIGINT	S	Este é o campo que guardará o identificador único da disciplina (chave estrangeira).
enunciado	TEXT	S	Este é o campo que guardará o enunciado da pergunta.
tipo_pergunta	VARCHAR(50)	S	Este é o campo que guardará o tipo da pergunta (“Verdadeiro ou Falso” ou “Múltipla Escolha”).
resposta	VARCHAR(50)	S	Este é o campo que guardará a resposta da pergunta.
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa pergunta. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa pergunta. Coluna inserida pelo framework Laravel.

deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa pergunta. Coluna inserida pelo framework Laravel via biblioteca "SoftDeletes".
RELACIONAMENTOS			
Tabela	Descrição		
disciplinas	A tabela "perguntas" possui relacionamento um para muitos com a tabela "disciplinas", ou seja, uma pergunta pode pertencer somente a uma disciplina, porém uma disciplina pode conter várias perguntas. O relacionamento é efetivado pela presença da chave estrangeira "disciplina_id" na tabela "perguntas".		
opcoes	A tabela "perguntas" possui relacionamento um para muitos com a tabela "opcoes", ou seja, uma pergunta pode conter várias opções (que seriam cinco), porém uma opção só está associada a uma pergunta. O relacionamento é efetivado pela presença da chave estrangeira "pergunta_id" na tabela "opcoes".		
provas	A tabela "perguntas" possui relacionamento um para muitos com a tabela "provas", ou seja, uma pergunta só existe em uma prova, porém uma prova pode conter várias perguntas. O relacionamento é efetivado pela presença da chave estrangeira "prova_id" na tabela "perguntas".		
tentativas	A tabela "perguntas" possui relacionamento muitos para muitos com a tabela "usuarios", que é implementada via tabela "tentativas" com um relacionamento um para muitos. Ou seja, uma pergunta pode possuir várias tentativas, porém, uma tentativa possui, por vez, uma única pergunta. O relacionamento é efetivado pela presença da chave estrangeira "pergunta_id" na tabela "tentativas".		

Fonte: BARBOSA, 2022.

2.3.7 Tabela *provas*

Esta tabela visa armazenar dados sobre as provas. As provas contêm perguntas e são aplicadas pelas bancas.

Tabela 7 – Tabela “provas”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da prova (chave primária).
banca_id	BIGINT	S	Este é o campo que guardará o identificador único da banca (chave estrangeira).
nome	VARCHAR(100)	S	Este é o campo que guardará o nome da prova.
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa prova. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa prova. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa prova. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
bancas	A tabela “provas” possui relacionamento um para muitos com a tabela “bancas”, ou seja, uma prova pode ser aplicada por somente uma banca, porém, uma banca pode aplicar várias provas. O relacionamento é efetivado pela presença da chave estrangeira “banca_id” na tabela “provas”.		
perguntas	A tabela “provas” possui relacionamento um para muitos com a tabela “perguntas”, ou seja, uma prova pode conter várias perguntas, porém uma pergunta só pode estar presente em uma prova. O relacionamento é efetivado pela presença da chave estrangeira “prova_id” na tabela “perguntas”.		

Fonte: BARBOSA, 2022.

2.3.8 Tabela *tentativas*

Esta tabela visa armazenar dados sobre as tentativas dos usuários em responderem uma pergunta. É importante ressaltar que é uma tabela que implementa o relacionamento muitos para muitos de usuários para perguntas. Uma vez que um usuário responda uma pergunta na aplicação, deve ser registrada, na tabela “*tentativas*”, as chaves estrangeiras necessárias (*pergunta_id* e *usuario_id*), e se a tentativa foi bem sucedida ou não.

Tabela 8 – Tabela “*tentativas*”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da tentativa (chave primária).
usuario_id	BIGINT	S	Este é o campo que guardará o identificador único do usuário (chave estrangeira).
pergunta_id	BIGINT	S	Este é o campo que guardará o identificador único da pergunta (chave estrangeira).
tentativa	TINYINT(1)	S	Este é o campo que guardará um valor booleano indicando o sucesso da tentativa. (0 corresponde a mal sucedida, 1 corresponde a bem sucedida)
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa tentativa. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa tentativa. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa tentativa. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
usuarios	A tabela “ <i>tentativas</i> ” possui relacionamento um para muitos com a tabela “ <i>usuarios</i> ”, implementando o relacionamento muitos para muitos de “ <i>usuarios</i> ” para “ <i>perguntas</i> ”. Conclui-se que, uma tentativa está associada somente a um usuário, porém um usuário pode tentar várias vezes, ou seja, possuir mais de uma tentativa. Quanto à esse relacionamento, verifica-se a presença da chave estrangeira “ <i>usuario_id</i> ” na tabela “ <i>tentativas</i> ”.		
perguntas	A tabela “ <i>tentativas</i> ” possui relacionamento um para muitos com a tabela “ <i>perguntas</i> ”, implementando o relacionamento muitos para muitos de “ <i>perguntas</i> ” para “ <i>usuarios</i> ”. Conclui-se que, uma tentativa está associada somente a uma pergunta, porém uma pergunta pode possuir várias tentativas de ser respondida corretamente, ou seja, possuir mais de uma tentativa. Quanto à esse relacionamento, verifica-se a presença da chave estrangeira “ <i>pergunta_id</i> ” na tabela “ <i>tentativas</i> ”.		

Fonte: BARBOSA, 2022.

2.3.9 Tabela *usuarios*

Esta tabela visa armazenar dados sobre os usuários do sistema. Seu CPF deve passar pela validação dos dois últimos dígitos verificadores e seu e-mail deve possuir a configuração “nome@domínio.com”.

Tabela 9 – Tabela “usuarios”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único do usuário
nome	VARCHAR(100)	S	Este é o campo que guardará o nome do usuário
senha	VARCHAR(100)	S	Este é o campo que guardará a senha criptografada do usuário.
data_de_nascimento	DATE	S	Este é o campo que guardará a data de nascimento do usuário.
apelido	VARCHAR(50)	S	Este é o campo que guardará o apelido do usuário.
email	VARCHAR(100)	S	Este é o campo que guardará o endereço de e-mail do usuário.
CPF	VARCHAR(11)	S	Este é o campo que guardará o CPF do usuário sem pontos e traço.
cidade	VARCHAR(100)	S	Este é o campo que guardará a cidade do usuário
created_at	TIMESTAMP	N	Este é o campo que guardará a data de criação do cadastro do usuário ou a data da inserção do registro correspondente à um certo usuário. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data de alteração do cadastro do usuário ou a data da edição do registro correspondente à um certo usuário. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data de exclusão do cadastro do usuário ou a data da exclusão do registro correspondente à um certo usuário. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
tentativas	A tabela “usuarios” possui relacionamento muitos para muitos com a tabela “perguntas”, que é implementada via tabela “tentativas” com um relacionamento um para muitos. Ou seja, um usuário pode possuir várias tentativas, porém, uma tentativa está associada, por vez, a um único usuário.		

	O relacionamento é efetivado pela presença da chave estrangeira "usuario_id" na tabela "tentativas".
--	--

Fonte: BARBOSA, 2022.

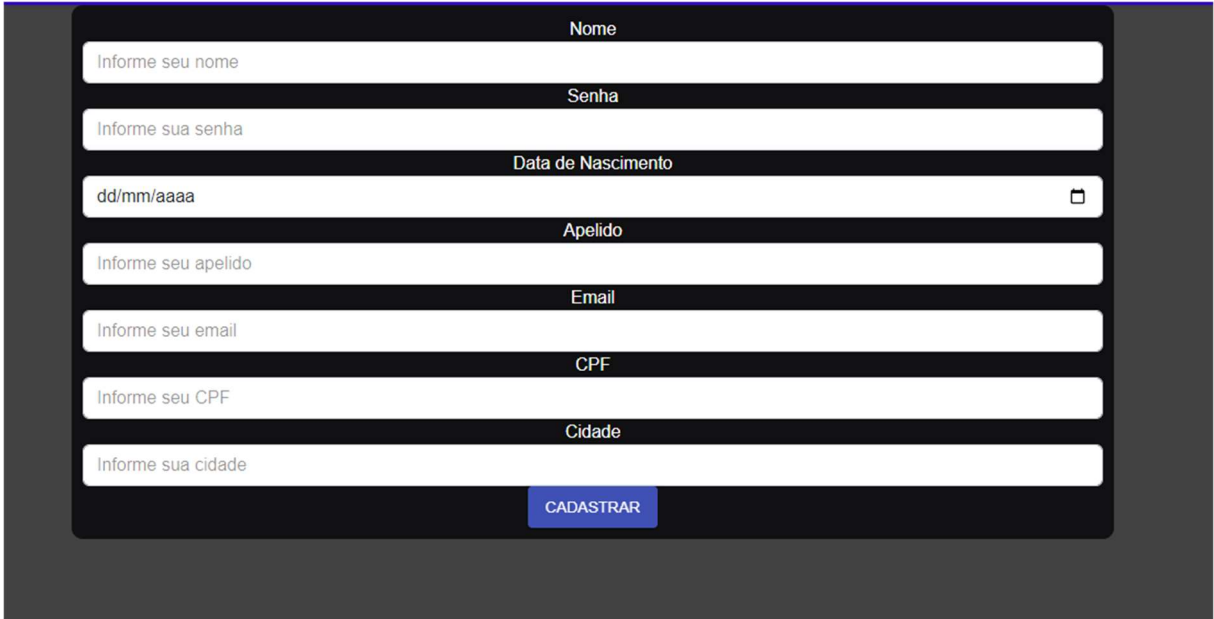
CAPÍTULO III

O Sistema Athena Web consiste em ser um quiz de perguntas de TI e pretende possibilitar que o usuário, uma vez cadastrado e autenticado, tenha acesso à perguntas de TI para que possa respondê-las e controlá-las. O sistema deve ser visualizado por meio de um navegador, como o Google Chrome, já que usa tecnologias como o HTML, que é uma linguagem de marcação que é processada em navegadores.

A criação do sistema se inspira no fato de que há pessoas que demandam uma melhor capacitação na área de Tecnologia da Informação devido aos movimentos do mercado e às mudanças na sociedade. A demanda por desenvolvedores de sistema só tem aumentado, e os profissionais de tecnologia que estão no mercado podem não ser suficientes em termos quantitativos ou não estão preparados ainda em termos qualitativos.

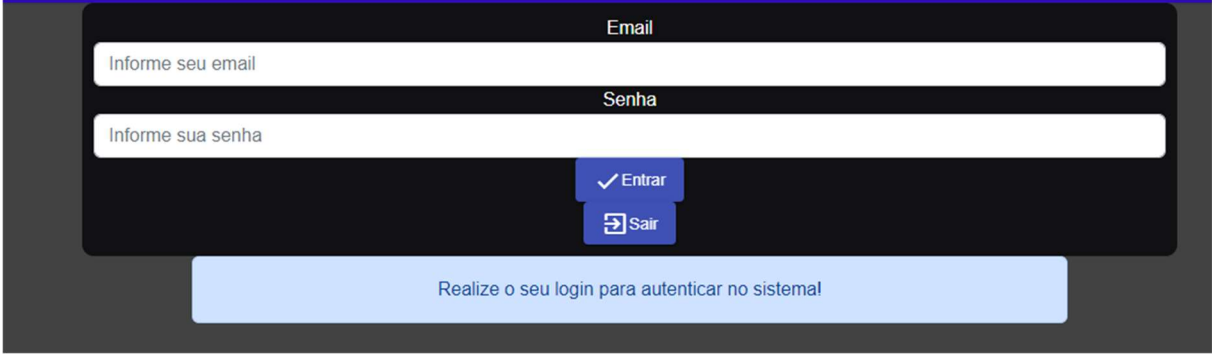
3.1. Conjunto de Tabelas

Tabela 1 – Manter Cadastro

Manter Cadastro
Caso de Uso: UC001 – Manter Cadastro

Funcionalidades:
RF001 – Manter Cadastro
<p>Esta tela tem como funcionalidade servir um formulário destinado à entrada de dados relacionados ao cadastro de usuário</p> <ol style="list-style-type: none"> 1. No campo “Nome” deve ser informado um nome com no mínimo 3 caracteres e no máximo 40 caracteres. Além desses requisitos, o nome de usuário deve ser único no banco de dados e é um campo obrigatório. 2. No campo “Senha” deve ser informado uma senha com no mínimo 3 dígitos e no máximo 40 dígitos, além de ser um campo obrigatório. 3. No campo “Data de Nascimento” deve ser informado uma data e é um campo obrigatório. 4. No campo “Apelido” deve ser informado um apelido com no mínimo 3 caracteres e no máximo 40 caracteres, além de ser um campo obrigatório. 5. No campo “Email” deve ser informado um e-mail possuindo os caracteres “@” e “.” para delimitar o nome e o domínio. Além desses requisitos, o e-mail do usuário deve ser único no banco de dados e é um campo obrigatório. 6. No campo “CPF” deve ser informado um número de CPF possua os dois últimos dígitos verificadores válidos. Além desses requisitos, o CPF do usuário deve ser um CPF único no banco de dados e é um campo obrigatório. 7. No campo “Cidade” deve ser informada uma cidade com no mínimo 3 caracteres e no máximo 40 caracteres, além de ser um campo obrigatório.

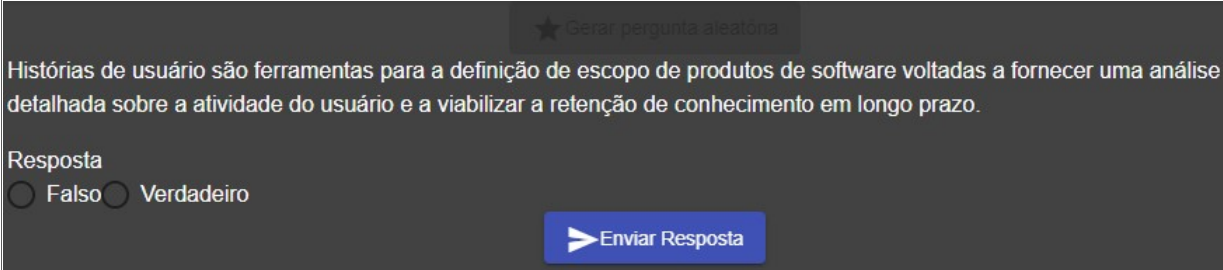
Fonte: BARBOSA, 2022.

Tabela 2 – Manter Login

Manter Login
Caso de Uso: UC002 – Manter Login

Funcionalidades:
RF002 – Manter Login
<p>Esta tela tem como funcionalidade servir um formulário destinado à entrada de dados relacionados à autenticação de um certo usuário para que possa utilizar o sistema.</p> <ol style="list-style-type: none"> 1. No campo “Email” deve ser informado um e-mail possuindo os caracteres “@” e “.” para delimitar o nome e o domínio. Além desses requisitos, o e-mail do usuário deve ser único no banco de dados e é um campo obrigatório. 2. No campo “Senha” deve ser informado uma senha com no mínimo 3 dígitos e no máximo 40 dígitos, além de ser um campo obrigatório.

Fonte: BARBOSA, 2022.

Tabela 3 – Responder Pergunta

Responder Pergunta
<p>Caso de Uso: UC003 – Responder Pergunta</p> 
<p>Funcionalidades:</p> <p>RF003 – Responder Pergunta</p> <p>Esta tela tem como funcionalidade servir um formulário, composto de “radio buttons” e um botão de submissão, destinado à entrada de dados relacionados à resposta de uma pergunta por um usuário específico.</p> <ol style="list-style-type: none"> 1. Ao entrar na tela, o usuário somente verá o botão “Gerar pergunta aleatória” preenchido com uma cor azul de fundo. 2. Após clicar no botão, uma pergunta aleatória será apresentada na tela e o botão será desativado. A pergunta é constituída de um enunciado e um formulário para marcar a resposta. 3. O usuário, após isso, deve marcar a resposta que acredita ser correta e clicar no botão “Enviar Resposta”. 4. Após isso, aparecerá um aviso sobre se a resposta foi correta ou não.

Fonte: BARBOSA, 2022.

Tabela 4 – Manter Pergunta

A imagem mostra a interface de usuário do sistema de perguntas, especificamente a tela "Manter Pergunta". No topo, há uma barra de navegação com o título "Manter Pergunta" e o caso de uso "Caso de Uso: UC004 – Manter Pergunta". Abaixo, há uma barra de ferramentas com os botões "Listar Pergunta" e "Criar Pergunta".

O título principal da tela é "Listar Pergunta", seguido por um ícone de interrogação amarelo. Abaixo dele, há uma tabela com as seguintes colunas: ID da Pergunta, ID da Prova, Nome da Prova, ID da Disciplina, Nome da Disciplina, Enunciado, Tipo de Pergunta, Resposta, Editar e Excluir.

A tabela contém duas linhas de dados:

ID da Pergunta	ID da Prova	Nome da Prova	ID da Disciplina	Nome da Disciplina	Enunciado	Tipo de Pergunta	Resposta	Editar	Excluir
1	1	Petrobras - Analista de Sistemas - Engenharia de Software	2	Engenharia de Software	O conceito de sprint tem sua origem no RUP a partir da execução das fases, cada uma delas com seu marco; cada ciclo no RUP tinha uma sprint considerada, assim como um projeto curto.	Verdadeiro ou Falso	Falso	Atualizar Pergunta 1	Deletar Pergunta 1
2	1	Petrobras - Analista de Sistemas - Engenharia de Software	2	Engenharia de Software	No Scrum, todo o trabalho necessário para atingir a meta do produto está embutido nas sprints, inclusive	Verdadeiro ou Falso	Verdadeiro	Atualizar Pergunta 2	Deletar Pergunta 2

Abaixo da tabela, há uma seção intitulada "Funcionalidades:" com o subtítulo "RF004 – Manter Pergunta". O texto descreve a funcionalidade de servir uma tabela destinada ao controle de dados relacionados às perguntas existentes no sistema.

- Caso o usuário possua dúvida sobre o que representa cada coluna, como ID da Pergunta, Nome da Prova, e as demais colunas, há um botão com um sinal de interrogação ao lado do título da tabela. Ao clicar no botão, é aberta uma janela com explicações sobre as operações e as colunas.
- Ao final de um registro, há dois botões, um responsável pela edição da pergunta e outro pela exclusão.
- Caso o usuário opte por atualizar a pergunta, ele pode clicar no botão "Atualizar Pergunta" e será aberta uma janela disponibilizando um formulário para entrada de dados relacionadas à atualização, ou completa ou parcial, daquela pergunta em específico.
- Já caso o usuário deseje excluir a pergunta, ele pode clicar no botão "Deletar Pergunta" e será aberta uma janela relacionada à exclusão daquela pergunta em específico.
- Caso o usuário queira criar uma pergunta nova, ele deve clicar no botão "Criar Pergunta", que se encontra ao lado de "Listar Pergunta". Nesse botão, estará disposto um formulário destinado à entrada de dados relacionados ao cadastro de uma pergunta. Os campos que devem ser preenchidos são também explicados em um

botão com sinal de interrogação caso o usuário necessite das explicações no momento de execução do aplicativo.

6. Quanto ao formulário de criação de pergunta:

a No campo “Identificador da Prova” deve ser informado um número de identificador único que já exista no banco de dados, além de ser um campo obrigatório.

b No campo “Identificador da Disciplina” deve ser informado um número de identificador único que já exista no banco de dados, além de ser um campo obrigatório.

c No campo “Enunciado” deve ser informado um texto que represente o enunciado da pergunta, tendo no mínimo 10 caracteres, além de ser um campo obrigatório.

d No campo “Tipo da Pergunta” deve ser selecionado a opção “Verdadeiro ou Falso” ou “Múltipla Escolha”, além de ser um campo obrigatório.

e No campo “Resposta” deve ser informado a resposta, que pode ser 1 para verdadeiro ou 0 para falso. Se o tipo de pergunta for “Verdadeiro ou Falso”, deve ser somente um dígito, porém, se for “Múltipla Escolha”, deve ser uma cadeia de 5 dígitos, os quais são compostos de quatro dígitos 0 e um dígito 1. O campo é obrigatório.

f Caso em “Tipo de Pergunta” seja selecionado “Múltipla Escolha”, serão exibidos 5 campos para serem preenchidas os enunciados das opções. São campos obrigatórios.

Fonte: BARBOSA, 2022.

3.2. Diagrama de Caso de Uso

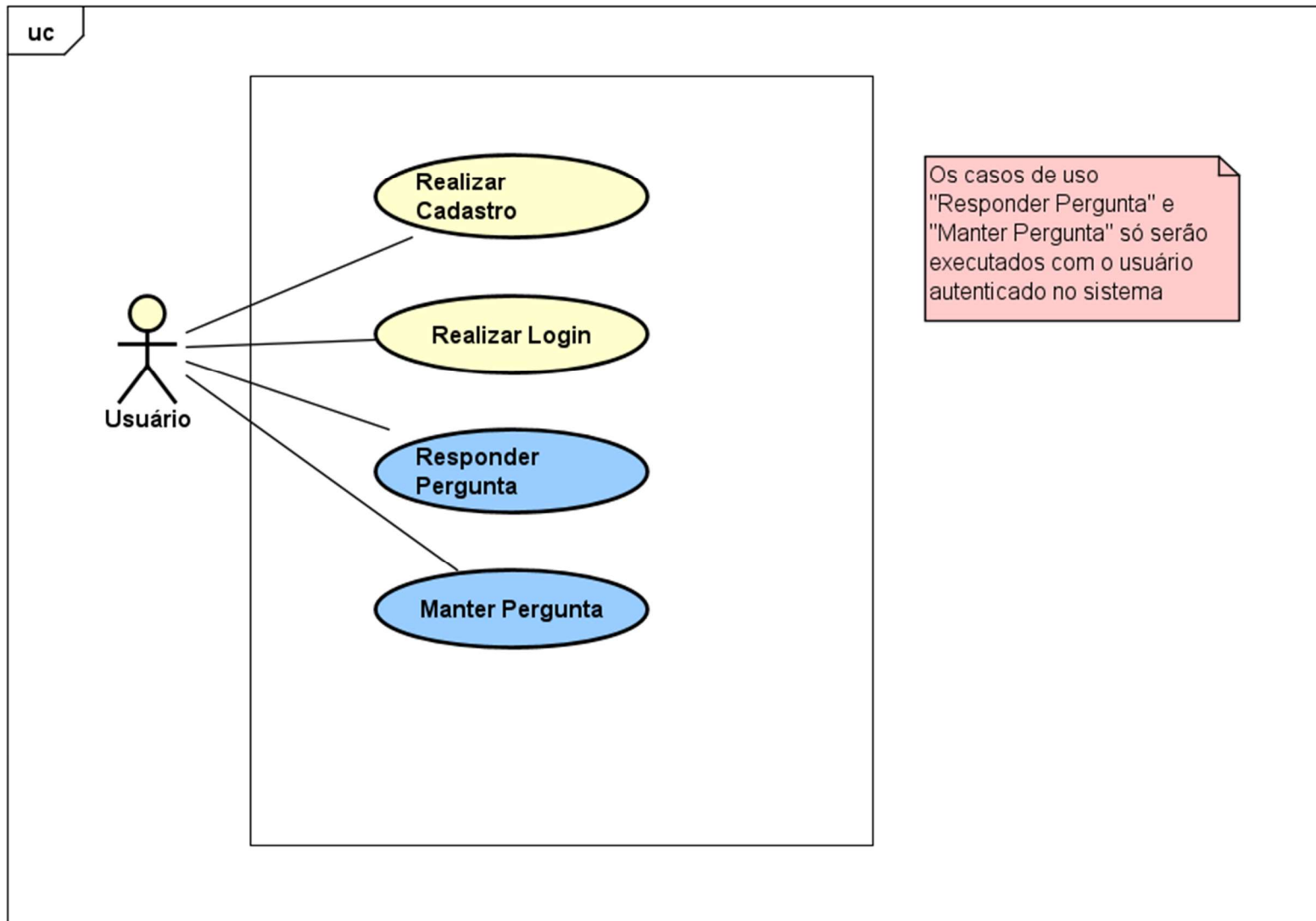


Figura 03– Diagrama de Caso de Uso
Fonte: BARBOSA, 2022.

3.3. Conjunto de Descrições de Caso de Uso

3.3.1 Realizar Cadastro

Nome UC	<i>Realizar Cadastro</i>
Atores	<i>Usuário</i>
Participação do ator	Permite o registro em banco de dados das informações de um certo usuário
Pré-condições	
O <i>Usuário</i> deve ter conexão com a Internet para acessar a página Web do Sistema Athena com a finalidade de fazer um cadastro no sistema para acessar mais funcionalidades. Uma vez na página “/home” do sistema, o usuário deve clicar em “Cadastro” para ser redirecionado à página “/realizar-cadastro”. É nessa página que o cadastro pode ser realizado.	
Pós-condições	
O <i>Usuário</i> está cadastrado no banco de dados do Sistema Athena.	
Fluxo Principal	
1.	O <i>Usuário</i> seleciona no cabeçalho a opção “Cadastro”
2.	O sistema carrega uma tela contendo o formulário de cadastro
3.	O <i>Usuário</i> entra com seu nome, CPF, senha, e-mail, apelido, data de nascimento e cidade.
4.	O <i>Usuário</i> submete o cadastro das informações pessoais
5.	O sistema valida a entrada de dados (CPF, senha, e-mail, ...).
6.	O sistema emite a mensagem de indicação de sucesso
7.	O caso de uso é finalizado
Fluxos Alternativos	
<u>A01 - O usuário cancela o cadastro</u>	
1.	O <i>Usuário</i> fecha o navegador ou sai da página que contém o formulário de cadastro do usuário.
2.	O sistema cancela a operação
3.	O caso de uso é finalizado
Fluxos de Exceção	
<u>E01 – O usuário já está cadastrado</u>	
1.	O sistema verifica que o CPF e/ou o e-mail inserido já foi utilizado alguma vez.
2.	O sistema envia a mensagem referente ao campo já utilizado.”

3.	O sistema retorna ao fluxo principal (P3) para reentrada do(s) dado(s) do <i>Usuário</i>
E02 – <u>Campo requerido não fornecido</u>	
1.	O <i>Usuário</i> não entra com algum campo requerido
2.	Após o processo de validação, o sistema impede a requisição referente ao cadastro de <i>usuário</i> .
3.	O sistema retorna ao fluxo principal (P3) para reentrada do(s) dado(s) do <i>Usuário</i> que está faltando
E03 – <u>O CPF do usuário é inválido</u>	
1.	O <i>Usuário</i> entra com número de CPF inválido;
2.	Após o processo de validação, o sistema impede a requisição referente ao cadastro do <i>Usuário</i> .”
3.	O sistema retorna ao fluxo principal (P3) para entrada do CPF do <i>Usuário</i> que está inválido

3.3.2 Realizar Login

Nome UC	<i>Realizar Login</i>
Atores	<i>Usuário</i>
Participação do ator	Permite a autenticação de um certo usuário no sistema de acordo com as informações persistidas no banco de dados
Pré-condições	
O <i>Usuário</i> deve ter conexão com a Internet para acessar a página Web do Sistema Athena com a finalidade de fazer login no sistema para acessar mais funcionalidades. Uma vez na página “/home” do sistema, o usuário deve clicar em “Login” para ser redirecionado à página “/realizar-login”. É nessa página que a autenticação pode ser realizada.	
Pós-condições	
O <i>Usuário</i> está autenticado no Sistema Athena, o que lhe permitirá acessar mais funcionalidade.	
Fluxo Principal	
1.	O <i>Usuário</i> seleciona no cabeçalho a opção “Login”
2.	O sistema carrega uma tela contendo o formulário de login
3.	O <i>Usuário</i> entra com seu e-mail e senha.
4.	O <i>Usuário</i> submete o formulário de login
5.	O sistema valida a entrada de dados.
6.	O sistema emite a mensagem de indicação de sucesso
7.	O caso de uso é finalizado
Fluxos Alternativos	
A01 - <u>O usuário cancela o login</u>	
1.	O <i>Usuário</i> fecha o navegador ou sai da página que contém o formulário de login
2.	O sistema cancela a operação
3.	O caso de uso é finalizado
Fluxos de Exceção	
E01 – <u>Campo requerido não fornecido</u>	
1.	O <i>Usuário</i> não entra com algum campo requerido
2.	Após o processo de validação, o sistema impede a requisição referente ao login
3.	O sistema retorna ao fluxo principal (P3) para entrada do(s) dado(s) do <i>Usuário</i> que está faltando
E02 – <u>As credenciais são inválidas</u>	

1.	O <i>Usuário</i> entra com uma combinação de e-mail e senha que não há no sistema
2.	Após o processo de validação, o sistema responde a requisição apontando que a combinação não há no sistema
3.	O sistema retorna ao fluxo principal (P3) para uma nova entrada de credenciais que sejam válidas.

3.3.3 Responder Pergunta

Nome UC	<i>Responder Pergunta</i>
Atores	<i>Usuário</i>
Participação do ator	Permite, ao usuário, que responda à uma pergunta aleatória. Sua resposta ficará armazenada no sistema.
Pré-condições	
O <i>Usuário</i> deve ter conexão com a Internet para acessar a página Web do Sistema Athena com a finalidade de responder à pergunta aleatória. Uma vez na página “/responder-pergunta” do sistema e devidamente autenticado, o <i>Usuário</i> pode prosseguir.	
Pós-condições	
O <i>Usuário</i> respondeu à uma pergunta aleatória, armazenando seu resultado em um banco de dados.	
Fluxo Principal	
1.	O <i>Usuário</i> seleciona no cabeçalho a opção “Quiz”
2.	O sistema carrega uma tela contendo um botão para gerar uma pergunta aleatória.
3.	O <i>Usuário</i> clica no respectivo botão.
4.	O sistema carrega uma pergunta aleatória, juntamente com um formulário para submeter a resposta.
5.	O <i>Usuário</i> marca a opção que pensa ser a correta.
6.	O <i>Usuário</i> submete o formulário.
7.	O sistema exibe uma mensagem referente ao fato da pergunta ter sido respondida corretamente ou não.
8.	O caso de uso é finalizado.
Fluxos Alternativos	
A01 - <u>O usuário cancela a operação</u>	
1.	O <i>Usuário</i> fecha o navegador ou sai da página que contém a pergunta gerada.
2.	O sistema cancela a operação.
3.	O caso de uso é finalizado.
Fluxos de Exceção	
E01 – <u>Opção não fornecida</u>	
1.	O <i>Usuário</i> não seleciona qualquer opção.
2.	Após o processo de validação, o sistema impede a requisição referente à resposta de pergunta.
3.	O sistema retorna ao fluxo principal (P5) para que o <i>Usuário</i> selecione uma opção.

3.3.4 Manter Pergunta

Nome UC	<i>Manter Pergunta</i>
Atores	<i>Usuário</i>
Participação do ator	Permite que um certo usuário controle as perguntas no sistema, podendo criá-las, alterá-las, deletá-las e fazer uma leitura de todas as perguntas.
Pré-condições	
O <i>Usuário</i> deve ter conexão com a Internet para acessar a página Web do Sistema Athena com a finalidade de controlar todas as perguntas do sistema. Uma vez na página “/manter-pergunta” do sistema e devidamente autenticado, o <i>Usuário</i> pode prosseguir com suas operações.	
Pós-condições	
O <i>Usuário</i> modificou a(s) pergunta(s) do Sistema Athena como desejou. As mudanças serão visíveis também ao passar pelo Caso de Uso 3, <i>Responder Pergunta</i> .	
Fluxo Principal	
1.	O <i>Usuário</i> seleciona no cabeçalho a opção “Controlar Pergunta”.
2.	O sistema carrega uma tela contendo uma tabela com todas as perguntas e dois botões, um para listar todas as perguntas e outro para criar uma nova pergunta.
3.	O <i>Usuário</i> clica no botão “Criar Pergunta”.
4.	O sistema apresenta um formulário destinado à criação de uma nova pergunta.
5.	O <i>Usuário</i> preenche todos os campos necessários.
6.	O <i>Usuário</i> submete o formulário.
7.	O sistema exibe uma mensagem referente à criação da pergunta.
8.	O caso de uso é finalizado.
Fluxos Alternativos	
A01 - <u>O usuário cancela o cadastro de pergunta</u>	
4.	O <i>Usuário</i> fecha o navegador ou sai da página que contém ou a tabela listando todas as perguntas ou o formulário de criação de pergunta.
5.	O sistema cancela a operação.
6.	O caso de uso é finalizado.
A02 - <u>O usuário deseja ver a listagem das perguntas</u>	
1.	O <i>Usuário</i> não clica no botão “Criar Pergunta”, mantendo-se na tela responsável pela listagem de perguntas.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha a pergunta de seu interesse e lê seus dados.
3.	O caso de uso é finalizado.

A03 - O usuário deseja atualizar uma pergunta

1.	O <i>Usuário</i> não clica no botão “Criar Pergunta”, mantendo-se na tela responsável pela listagem de perguntas.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha a pergunta de seu interesse e lê seus dados.
3.	O <i>Usuário</i> , no registro referente à pergunta desejada, clica no botão responsável pela atualização.
4.	O sistema disponibiliza um formulário para a atualização de perguntas.
5.	O <i>Usuário</i> preenche os campos conforme é solicitado e submete as alterações.
6.	O caso de uso é finalizado.

A04 - O usuário deseja excluir uma pergunta

1.	O <i>Usuário</i> não clica no botão “Criar Pergunta”, mantendo-se na tela responsável pela listagem de perguntas.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha a pergunta de seu interesse e lê seus dados.
3.	O <i>Usuário</i> , no registro referente à pergunta desejada, clica no botão responsável pela exclusão.
4.	O sistema sinaliza qual é a pergunta a ser deletada em questão.
5.	O <i>Usuário</i> pressiona o botão referente à exclusão.
6.	O caso de uso é finalizado.

Fluxos de Exceção**E01 – Campo requerido não fornecido**

1.	O <i>Usuário</i> não entra com algum campo requerido
2.	Após o processo de validação, o sistema impede a requisição referente à criação de perguntas
3.	O sistema retorna ao fluxo principal (P5) para entrada do(s) dado(s) do <i>Usuário</i> que está faltando

E02 – Os dados inseridos são inválidos

1.	O <i>Usuário</i> entra com uma entrada de dados que não passa pelas verificações que o sistema faz.
2.	Após o processo de validação, o sistema responde a requisição apontando os erros relativos à entrada de dados.
3.	O sistema retorna ao fluxo principal (P5) para uma nova entrada de dados que sejam válidos.

CONCLUSÃO

O projeto foi desenvolvido, tanto na parte documental quanto na parte computacional, no contexto da gamificação, no aprendizado de Tecnologia da Informação e da experiência que as pessoas têm ao adquirirem conhecimento de Tecnologia da Informação por meios de sistemas Web com aspectos de gamificação, principalmente estudantes de informática. É feita a menção em diversos artigos científicos os benefícios da gamificação associados ao aprendizado, o que reforça a aliança do aprendizado com aspectos de gamificação, além desse aprendizado se dar por meios de aplicações em dispositivos.

É perceptível que os objetivos específicos foram alcançados com sucesso. O usuário, ao entrar no sistema, pode efetuar seu cadastrado, e, após isso, pode efetuar seu login. Estando autenticado no sistema, pode controlar perguntas. Ao controlar as perguntas, o sistema possibilita que o usuário faça operações de banco de dados já bem conhecidas, como inserir um registro, alterar esse mesmo registro, seja de forma completa ou incompleta, deletar esse registro e ler todos os registros presentes no banco de dados. Além disso, o usuário autenticado pode responder às perguntas em um formato de quiz, e todas suas alterações no banco de dados refletirão no seu processo de responder perguntas.

A relevância do projeto reside no fato dele existir enquanto aplicação web, ou seja, um sistema que roda em um navegador e oferece funcionalidades ao usuário por meio de um ambiente digital, ao invés de analógico. O mercado e a sociedade estão passando por mudanças e soluções que não consideram o ambiente virtual estão possuindo menos relevância do que aquelas que o consideram.

O projeto não apresentou dificuldades significativas, mas sim, oportunidades significativas. Foi por meio do projeto que pude exercitar o uso do framework Laravel, que utiliza a linguagem PHP e possui muitos recursos para lidar com API's RESTful, por meio das "Controllers", e com tabelas em bancos de dados, por meio do recurso de "ORM". Consegui exercitar também o uso básico de criptografia de senha e a aplicação de uma arquitetura MVC.

O sistema, apesar de já apresentar um protótipo funcional de um sistema de quis de perguntas de TI, está aberto para melhorias, o que se traduziriam em

implementação de mais funcionalidades por meio de inserção de códigos ou refatoração dos códigos que já estão no sistema visando uma melhor manutenibilidade do código que já está escrito ou aprimoramento e expansão das funcionalidades já existentes. Tanto o Angular quanto o Laravel suportam autenticação via JWT, o que não é utilizado no presente sistema. A melhoria que poderia ser implementada poderia ser uma autenticação mais rigorosa para evitar acessos indesejados via JWT.

BIBLIOGRAFIA

SILVA, J. B.; GILVANDENYS, L. S.; JUSCILEIDE, B. C. Gamificação como estratégia de aprendizagem ativa no ensino de Física. **Revista Brasileira de Ensino de Física**, São Paulo, v. 41, n. 4, 28 de Abril de 2019. Disponível em: < <https://www.scielo.br/j/rbef/a/Tx3KQcf5G9PvcgQB4vswPbq/>>. Acesso em: 15 set de 2022.

SENAC Divinópolis. Aquecido no Brasil, mercado de TI indica aumento de vagas de trabalho. **G1**, 2022. Disponível em: <<https://g1.globo.com/mg/centro-oeste/especial-publicitario/senac/senac-em-divinopolis/noticia/2022/06/15/aquecido-no-brasil-mercado-de-ti-indica-aumento-de-vagas-de-trabalho.ghtml>>. Acesso em: 15 set de 2022.

OLIVEIRA, A. M.; FREITAS A. A.; LIMA C. S.; ARRUDA J. T.; JAIME J. C.; MOURA L. R.; Aplicação combinada de metodologia ativa e tecnologia de informação e comunicação no ensino médico: um relato de experiência. **37º Seminário de Atualização de Práticas Docentes**. Anápolis, v. 37, p. 311-515. jan de 2019. Disponível em: < https://www.researchgate.net/publication/339971631_APLICACAO_COMBINADA_DE_METODOLOGIA_ATIVA_E_TECNOLOGIA_DE_INFORMACAO_E_COMUNICACAO_NO_ENSI_NO_MEDICO_UM_RELATO_DE_EXPERIENCIA>. Acesso em: 15 set de 2022.

MIRI, D. H.; SANTOS J. G.; MATTE J.; CHAIS C.; OLEA P. M.; Gamificação na área de Tecnologia da Informação de uma indústria metalomecânica: um estudo de caso. **DESENVOLVE: Revista de Gestão do Unilasalle**. Canoas, v. 10, n. 1, p. 01-22, mar. 2021. Disponível em: < <https://revistas.unilasalle.edu.br/index.php/desenvolve/article/view/7653>>. Acesso em: 15 set de 2022.