



## Nome do projeto: QuizTech Challenge: Aprendendo Tecnologia de Forma Divertida e Progressiva

Beatriz Palombarini de Souza<sup>1</sup>

Disciplina: Prática Profissional em Análise e Desenvolvimento de Sistemas

<sup>1</sup> Faculdade de Computação e Informática  
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brazil

10340701@mackenzista.com.br

**Abstract.** *The QuizTech Challenge is an educational game leveraging gamification to make technology learning engaging and progressive. The game features a series of questions categorized by units, topics, and difficulty levels, covering a broad range of IT areas such as development, artificial intelligence, cybersecurity, and networking. Difficulty increases as players advance, with automatic progress saving, allowing users to resume without needing login credentials. Hosted as a static site on AWS S3, player progress is stored in AWS DynamoDB, and identification is managed locally via browser LocalStorage. This project aims to offer an interactive learning platform that motivates users to expand their IT knowledge in a structured and engaging way through immediate feedback and level-based progression.*

**Keywords:** *Gamification, technology learning, educational games, IT knowledge, progressive learning, AWS S3, AWS DynamoDB, LocalStorage, cybersecurity, artificial intelligence.*

**Resumo.** *O QuizTech Challenge é um jogo educacional que utiliza gamificação para tornar o aprendizado sobre tecnologia envolvente e progressivo. O jogo apresenta uma série de perguntas organizadas por unidades, tópicos e níveis de dificuldade, abrangendo diversas áreas da TI, como desenvolvimento, inteligência artificial, segurança da informação e redes. À medida que os jogadores avançam, a dificuldade das perguntas aumenta, e o progresso é salvo automaticamente, permitindo que retomem de onde pararam sem necessidade de login ou senha. O projeto é hospedado como um site estático no AWS S3, com o progresso armazenado no AWS DynamoDB e a identificação dos jogadores gerenciada localmente por meio do LocalStorage do navegador. O objetivo é fornecer uma plataforma de aprendizado interativa que incentive os jogadores a ampliarem seus conhecimentos em TI de forma estruturada e envolvente, com feedback imediato e progresso baseado em níveis.*

**Palavras-chave:** *Gamificação, aprendizado em tecnologia, jogos educacionais, conhecimento em TI, aprendizado progressivo, AWS S3, AWS DynamoDB, LocalStorage, segurança da informação, inteligência artificial.*

## 1. Introdução

O QuizTech Challenge é um jogo educacional inovador, projetado para tornar o aprendizado sobre tecnologia acessível, interativo e envolvente. Utilizando os princípios da gamificação, o jogo proporciona uma experiência imersiva, na qual os usuários enfrentam desafios progressivos por meio de perguntas organizadas por unidades, tópicos e níveis de dificuldade. Essas perguntas abrangem uma ampla gama de áreas da tecnologia da informação (TI), incluindo desenvolvimento de software, inteligência artificial, segurança cibernética, redes e muitas outras disciplinas fundamentais para o cenário tecnológico atual.

O diferencial do QuizTech Challenge está em sua estrutura amigável e flexível, que permite aos usuários aprenderem de maneira autônoma e contínua. O sistema salva automaticamente o progresso do jogador, eliminando a necessidade de logins ou senhas, garantindo que a experiência de aprendizado seja prática e descomplicada. Essa funcionalidade é possível graças à integração com tecnologias modernas: o jogo é implementado como um site estático hospedado na infraestrutura robusta do AWS S3, enquanto os dados de progresso dos jogadores são armazenados no banco de dados escalável AWS DynamoDB. A identificação única de cada jogador é gerenciada diretamente no navegador por meio do LocalStorage, mantendo o foco na simplicidade e na acessibilidade.

Além disso, o QuizTech Challenge foi concebido para atender tanto iniciantes quanto profissionais da área de TI, oferecendo conteúdos organizados de forma estruturada e com feedback imediato. Isso permite que os jogadores identifiquem rapidamente suas áreas de oportunidade e consolidem seus conhecimentos enquanto avançam por desafios que se tornam gradualmente mais complexos. Dessa forma, o projeto não apenas auxilia no aprendizado de conceitos técnicos, mas também promove uma experiência engajante e divertida que incentiva o desenvolvimento contínuo de habilidades cruciais para o mundo da tecnologia.

Com o avanço acelerado da transformação digital e a crescente demanda por profissionais qualificados, o QuizTech Challenge surge como uma ferramenta poderosa para preencher lacunas de conhecimento e inspirar novos aprendizes a explorarem os campos vastos e dinâmicos da TI.

## 2. Interessados no Projeto QuizTech Challenge

Os interessados no projeto **QuizTech Challenge** são indivíduos, grupos ou organizações que têm algum nível de envolvimento, impacto ou interesse nos resultados e funcionalidades do jogo. Abaixo estão os principais interessados e suas respectivas expectativas:

### 1. Jogadores

- **Descrição:** Indivíduos que utilizam o jogo para aprender sobre tecnologia e expandir seus conhecimentos.
- **Expectativas:**
  - Aprendizado eficiente e divertido sobre diferentes tópicos de TI.
  - Interface amigável e acessível em diversos dispositivos.
  - Progressão clara, organizada e motivadora no conteúdo.
  - Garantia de privacidade, sem necessidade de criar contas ou fornecer informações pessoais.

### 2. Desenvolvedores e Equipe Técnica

- **Descrição:** Equipe responsável pelo desenvolvimento, manutenção e aprimoramento do jogo.

- **Expectativas:**
  - Feedback contínuo dos usuários para melhorias do sistema.
  - Código bem estruturado e modular, permitindo expansão futura do jogo.
  - Utilização de tecnologias escaláveis e de baixo custo (como AWS S3 e DynamoDB) para sustentar o projeto.
  - Facilidade de monitoramento e suporte técnico para assegurar a estabilidade do sistema.

### 3. Educadores e Instituições de Ensino

- **Descrição:** Professores, instituições ou organizações que podem adotar o QuizTech Challenge como uma ferramenta complementar de ensino.
- **Expectativas:**
  - Conteúdo técnico preciso e de qualidade.
  - Organização didática dos tópicos e níveis de dificuldade.
  - Possibilidade de adaptar ou sugerir conteúdos para personalizar a experiência de aprendizado.
  - Relatórios opcionais para acompanhamento de desempenho (em futuras implementações).

### 4. Patrocinadores e Investidores

- **Descrição:** Organizações ou indivíduos que financiam ou apoiam o projeto, buscando retorno financeiro ou impacto social.
- **Expectativas:**
  - Visibilidade do projeto e impacto positivo no aprendizado de TI.
  - Retorno sobre o investimento através da expansão do público-alvo e adoção do jogo.
  - Utilização eficiente de recursos financeiros e tecnológicos.

### 5. Comunidade de Tecnologia

- **Descrição:** Grupos e comunidades ligados à área de TI que podem apoiar, divulgar ou se beneficiar do projeto.
- **Expectativas:**
  - Promoção do aprendizado e do crescimento de novos profissionais na área.
  - Contribuição para a democratização do conhecimento em tecnologia.
  - Envolvimento com sugestões ou melhorias técnicas.

### 6. Entusiastas de Gamificação

- **Descrição:** Pessoas ou grupos interessados no uso de gamificação para fins educacionais.
- **Expectativas:**
  - Uso criativo e eficaz da gamificação para engajar usuários no aprendizado.
  - Inspiração para outros projetos educacionais baseados em jogos.

## 7. Provedores de Infraestrutura Tecnológica

- **Descrição:** Plataformas e serviços que sustentam o funcionamento do jogo, como AWS (S3, DynamoDB).
- **Expectativas:**
  - Uso adequado de seus serviços para garantir a performance e segurança do sistema.
  - Potencial de expansão no uso de mais serviços à medida que o projeto cresce.

## Conexão Entre os Interessados

Cada grupo tem interesses diferentes, mas todos contribuem de forma interdependente para o sucesso do projeto. Jogadores fornecem feedback para melhorias, desenvolvedores criam a experiência ideal, educadores amplificam o impacto educacional, e patrocinadores e investidores tornam a continuidade do projeto viável. O alinhamento das expectativas desses grupos é essencial para o avanço do QuizTech Challenge como uma ferramenta de aprendizado inovadora e sustentável.

## 3. Objetivos:

### Geral

Criar uma plataforma educacional interativa baseada em gamificação que permita aos usuários aprenderem sobre tecnologia de forma envolvente, estruturada e acessível, promovendo o desenvolvimento progressivo de conhecimentos em TI e habilidades práticas por meio de desafios organizados e feedback imediato.

### Específicos

- Proporcionar um ambiente de aprendizado dinâmico e adaptativo, cobrindo tópicos essenciais da tecnologia da informação, como desenvolvimento, inteligência artificial, redes e segurança da informação.
- Incentivar o engajamento contínuo dos jogadores por meio de um sistema de progressão baseado em níveis e desafios crescentes de dificuldade.
- Garantir acessibilidade e simplicidade no uso, permitindo que os jogadores retomem seu progresso a qualquer momento sem complicações, como cadastros ou autenticações.
- Introduzir diferentes formatos de perguntas (flashcards, múltipla escolha, completar frases) para tornar o aprendizado mais interativo e diversificado.

### Objetivos Funcionais

- Exibir perguntas organizadas por unidade de aprendizado, tópico e níveis de dificuldade, abrangendo diferentes formatos interativos como flashcards, múltipla escolha e completar frases.
- Fornecer feedback imediato para cada resposta dada pelos jogadores, indicando se está correta ou incorreta, promovendo aprendizado ativo.
- Implementar um sistema de progressão baseado no desempenho do jogador, permitindo que ele suba de nível ao concluir unidades e acertar perguntas.

- Salvar automaticamente o progresso do jogador, utilizando AWS DynamoDB para armazenamento remoto e LocalStorage para identificação local, sem necessidade de login ou senha.

### **Objetivos Não Funcionais**

- Garantir que a interface do usuário seja responsiva, proporcionando uma experiência fluida e consistente em dispositivos móveis e desktops.
- Otimizar o desempenho do sistema, garantindo carregamento rápido das perguntas e respostas, mesmo em conexões limitadas.
- Manter o site estático hospedado no AWS S3, utilizando os benefícios do AWS Free Tier para uma solução de baixo custo e alta disponibilidade.
- Assegurar a privacidade e a segurança dos dados dos jogadores, armazenando apenas informações essenciais como progresso e identificações anônimas, sem a coleta de dados pessoais.

URL de acesso ao repositório de código-fonte do grupo:

[https://github.com/10340701/QuizTechChallenge\\_ADS\\_PraticaProfissionalEmADS](https://github.com/10340701/QuizTechChallenge_ADS_PraticaProfissionalEmADS)

URL de acesso ao quadro de acompanhamento do projeto do grupo:

<https://trello.com/invite/b/67421a6c757eb0b6081ad0bd/ATTI7521d3756c16f6939cafc6083170144A56054AC/quiztech-challenge>

Link do vídeo no youtube: <https://youtu.be/qDVIuS50frg>

Link do site no s3: <http://beatriz-quiztech.s3-website-us-east-1.amazonaws.com/>

## **4. Descrição do Projeto**

### **Estrutura do Jogo:**

O QuizTech Challenge é organizado em unidades que cobrem diversos tópicos de tecnologia, com perguntas de dificuldade crescente. Cada unidade é composta de múltiplos tópicos, e as perguntas variam em formato (flashcards, múltipla escolha, completar frase).

A estrutura é composta por:

- Unidades: Seções de aprendizado organizadas por níveis de dificuldade, indo de 0 a 10.
- Tópicos: Dentro de cada unidade, os tópicos abrangem áreas como Conhecimentos Gerais, Desenvolvimento, Testes de Software, Inteligência Artificial, entre outros.
- Níveis de Dificuldade: Cada unidade tem um nível de dificuldade associado (0-2 para iniciante, 8-10 para especialista).

As perguntas evoluem à medida que o jogador progride no jogo, com tópicos e dificuldades sendo desbloqueados conforme o jogador avança para novos níveis.

### **Tipos de Perguntas:**

O jogo possui três formatos principais de perguntas:

1. Flashcards: Perguntas de memorização, onde o jogador vê um conceito ou termo e deve responder ou memorizar a resposta.

Exemplo: O que significa HTTP?

2. Múltipla Escolha: Perguntas onde o jogador escolhe a resposta correta entre várias opções.

Exemplo: Qual linguagem de programação é mais usada para desenvolvimento web? (Python, JavaScript, Java, Ruby)

3. Completar Frase: O jogador preenche a lacuna em uma frase para completar uma definição ou conceito técnico.

Exemplo: O comando SQL usado para recuperar dados de um banco de dados é \_\_\_\_\_.

### **Sistema de Níveis e Progressão:**

O jogador sobe de nível conforme acerta perguntas e completa unidades. Cada unidade contém tópicos com dificuldades específicas, e o progresso é salvo automaticamente utilizando AWS DynamoDB e LocalStorage.

Os níveis são organizados da seguinte forma:

- Unidade 1 (Dificuldade 0-2): Tópicos iniciais como Conhecimentos Gerais de TI, Testes de Software e Desenvolvimento.
- Unidade 2 (Dificuldade 2-4): Tópicos mais avançados como DevOps, Computação em Nuvem e Segurança da Informação.
- Unidade 3 (Dificuldade 4-6): Perguntas sobre metodologias ágeis, IoT, Big Data, entre outros.
- Unidade 4 (Dificuldade 6-8): Conteúdo especializado como Blockchain, IA Avançada, e Segurança em Infraestruturas Críticas.
- Unidade 5 (Dificuldade 8-9): Tópicos de alta especialização como Kubernetes, Computação Quântica, e Redes Neurais Profundas.
- Unidade 6 (Dificuldade 10): Perguntas desafiadoras e complexas sobre todas as áreas de TI.

## **5. Diagrama de casos de uso**

Iniciar Sessão: O jogador entra no sistema e recebe uma ID anônima no navegador (sem login).

Carregar Perguntas: O sistema carrega perguntas do arquivo JSON com base no nível do jogador.

Responder Pergunta: O jogador responde às perguntas e o sistema fornece feedback em tempo real.

Subir de Nível: O jogador sobe de nível conforme completa as perguntas e tópicos.

Salvar Progresso: O sistema salva o progresso automaticamente usando AWS DynamoDB.

Salvar Progresso: O sistema restaura o progresso do jogador quando ele retorna ao jogo.

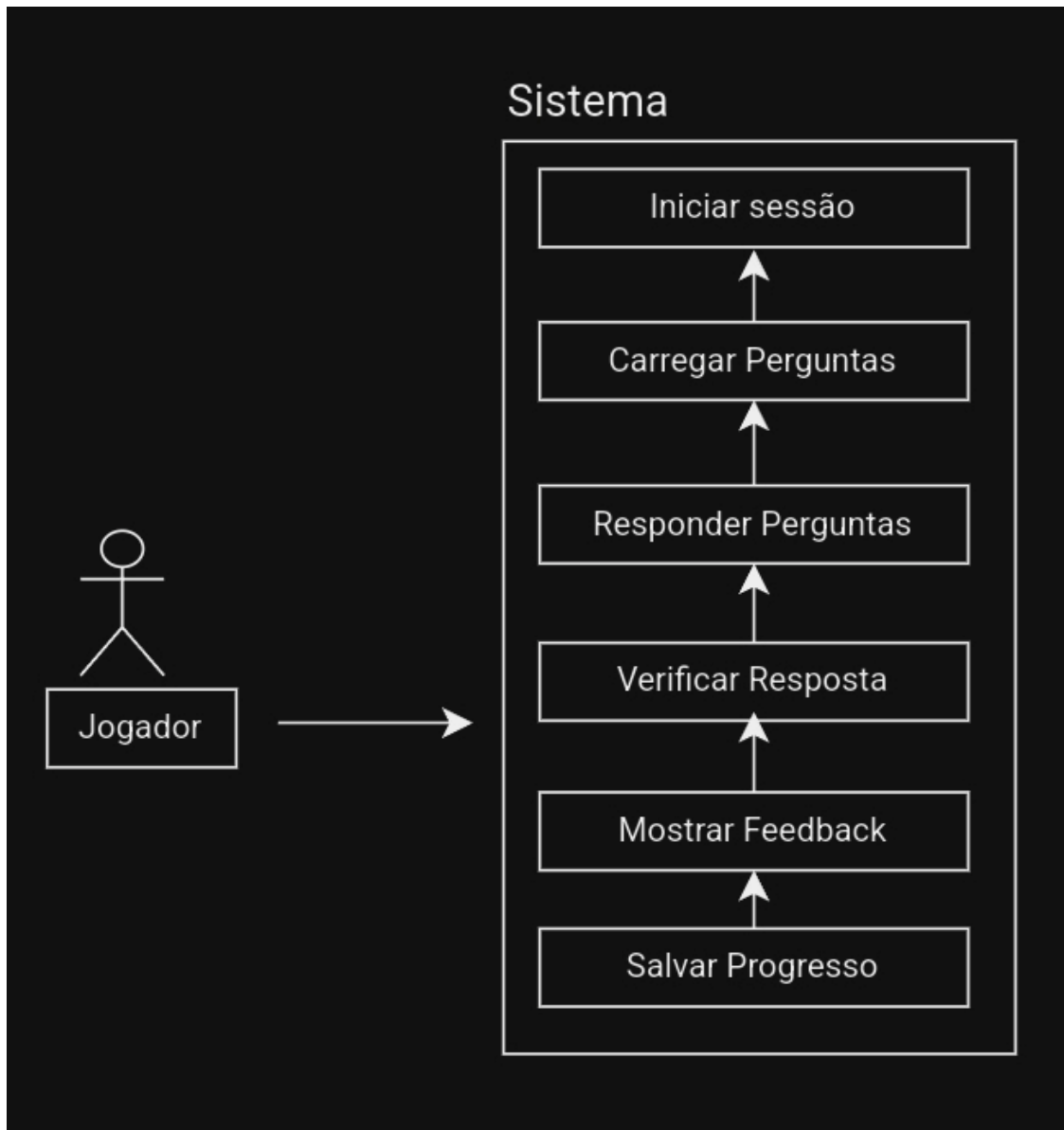


Figura 1: Caso de Uso - Jogar QuizTech Challenge

## 6. Descrição Detalhada do Caso de Uso: Jogar QuizTech Challenge

### Ator Principal:

- **Jogador:** Pessoa que interage com o sistema para responder às perguntas do quiz e aprender sobre tecnologia.

### Fluxo de Eventos:

#### Iniciar Sessão

- **Descrição:** O jogador acessa o sistema. O sistema cria ou recupera uma **ID anônima** no navegador do jogador, utilizando o recurso de LocalStorage.
- **Detalhes Técnicos:**
  - Se o jogador é novo, uma ID única é gerada e armazenada localmente.
  - Se já existe uma ID armazenada, ela é reutilizada para carregar o progresso do jogador.

### Carregar Perguntas

- **Descrição:** O sistema carrega um conjunto de perguntas baseado no nível e no progresso atual do jogador.
- **Detalhes Técnicos:**
  - As perguntas são armazenadas em um arquivo **JSON** local ou em uma API configurada.
  - O sistema filtra as perguntas de acordo com:
    - Unidade ou tópico em que o jogador está.
    - Nível de dificuldade correspondente ao nível atual do jogador.

### Responder Perguntas

- **Descrição:** O jogador visualiza uma pergunta e seleciona uma resposta entre as opções apresentadas.
- **Detalhes Técnicos:**
  - A pergunta pode estar em diferentes formatos:
    - Múltipla escolha.
    - Flashcards.
    - Completar a frase.
  - A resposta do jogador é enviada ao sistema para validação.

### Verificar Resposta

- **Descrição:** O sistema avalia se a resposta dada pelo jogador está correta ou incorreta.
- **Detalhes Técnicos:**
  - A validação é feita comparando a resposta fornecida com a resposta correta armazenada no arquivo JSON ou banco de dados.

### Mostrar Feedback

- **Descrição:** O sistema exibe ao jogador um feedback em tempo real indicando se a resposta está correta ou errada.
- **Detalhes Técnicos:**
  - Mensagens como "Resposta correta!" ou "Resposta incorreta. Tente novamente!" são exibidas na interface.
  - O feedback pode incluir explicações adicionais para as perguntas, ajudando o jogador a aprender com os erros.

### Subir de Nível

- **Descrição:** Ao concluir um conjunto de perguntas dentro de um tópico ou unidade, o jogador sobe de nível.
- **Detalhes Técnicos:**
  - O sistema verifica a quantidade de perguntas respondidas corretamente e, ao atingir um limite definido, o jogador avança para o próximo nível ou tópico.



- Este avanço é refletido nas perguntas carregadas futuramente, que se tornam mais desafiadoras.

### Salvar Progresso

- **Descrição:** O progresso do jogador (como nível, perguntas respondidas, e tópicos concluídos) é salvo automaticamente após cada interação.
- **Detalhes Técnicos:**
  - Utiliza o **AWS DynamoDB** para armazenar o progresso no servidor, permitindo recuperação futura.
  - Localmente, o progresso é associado à ID anônima armazenada no navegador (LocalStorage), garantindo continuidade da experiência sem login.

### Restaurar Progresso

- **Descrição:** Quando o jogador retorna ao sistema, seu progresso é recuperado automaticamente com base na ID anônima.
- **Detalhes Técnicos:**
  - O sistema consulta o AWS DynamoDB utilizando a ID armazenada no navegador para obter os dados de progresso.
  - Se o progresso for encontrado, ele é restaurado e exibido ao jogador, permitindo que continue exatamente de onde parou.

### Fluxo Alternativo

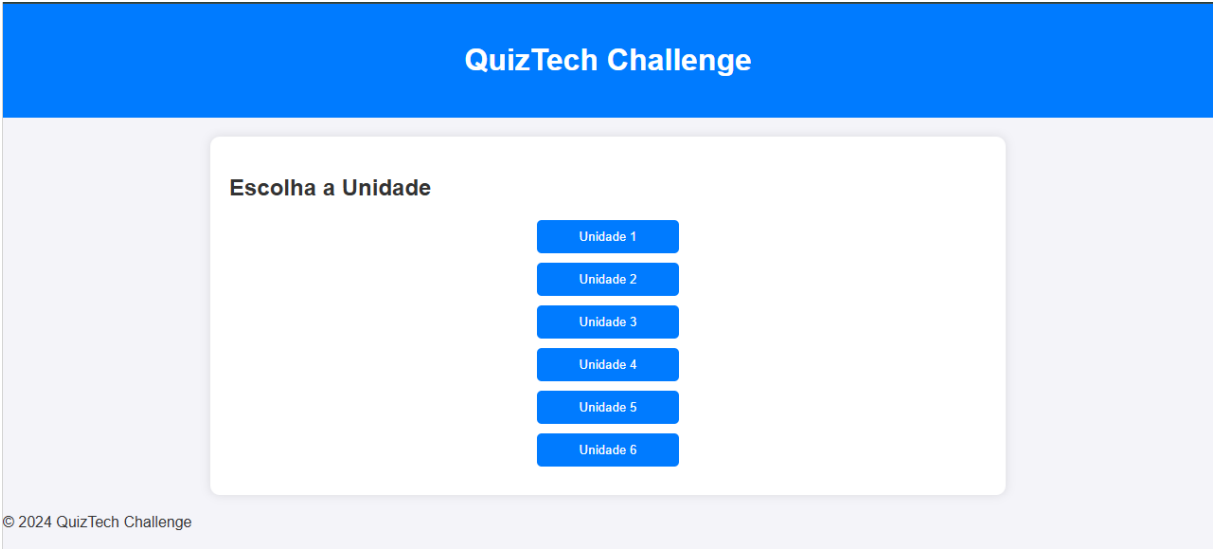
- **Erro ao Salvar Progresso:** Se houver uma falha ao salvar no AWS DynamoDB (ex.: ausência de conexão com a internet), o progresso será armazenado temporariamente no LocalStorage do navegador e sincronizado posteriormente.

### Considerações Gerais

- **Requisitos de Interface:** A interface deve ser intuitiva e responsiva para que o jogador consiga interagir facilmente em dispositivos móveis e desktops.
- **Privacidade:** Nenhuma informação pessoal é coletada; o jogador é identificado apenas pela ID anônima.
- **Segurança:** Os dados salvos são limitados ao progresso no jogo, minimizando riscos.

Esse caso de uso reflete a jornada do jogador ao interagir com o sistema, desde a entrada até a finalização de uma sessão, garantindo continuidade, interatividade e segurança durante toda a experiência.

## 7. Protótipos de tela



## 8. Modelo de Domínio:

### Entidades e Atributos

#### 1. Jogador

- ID (único, gerado automaticamente)
- Nome (opcional, pode ser usado para personalização futura)
- Nível Atual (indica o nível do jogador)
- Unidade Atual (indica a unidade que o jogador está explorando)
- Progresso (indica o percentual concluído do jogo)
- Data de Criação (quando o jogador começou a jogar)
- Data de Última Atualização (última vez que jogou)

#### 2. Pergunta

- ID (único)
- Texto (o enunciado da pergunta)
- Tipo (múltipla escolha, flashcard, completar frase)
- Opções (lista de possíveis respostas, no caso de múltipla escolha)
- Resposta Correta (a resposta esperada)
- Explicação (opcional, feedback educativo para o jogador)
- Nível de Dificuldade (fácil, médio, difícil)
- Tópico (a que área ou unidade pertence)

#### 3. Unidade

- ID (único)
- Nome (nome da unidade, como "Redes" ou "Segurança da Informação")
- Descrição (breve explicação do que será abordado)
- Lista de Tópicos (referências aos tópicos incluídos na unidade)

#### 4. **Tópico**

- ID (único)
- Nome (nome do tópico, como "Fundamentos de Redes")
- Lista de Perguntas (referências às perguntas relacionadas)
- Unidade Relacionada (referência à unidade associada)

#### 5. **Progresso**

- ID (único)
- Jogador (referência ao jogador associado)
- Unidade Atual (unidade que está sendo concluída)
- Tópico Atual (tópico que está sendo trabalhado)
- Perguntas Respondidas (lista de IDs de perguntas já respondidas)
- Acertos (número total de acertos)
- Erros (número total de erros)
- Percentual Concluído (com base nas perguntas respondidas)

---

#### **Relacionamentos**

1. **Jogador** possui um **Progresso**.
2. **Unidade** contém vários **Tópicos**.
3. **Tópico** inclui várias **Perguntas**.
4. **Progresso** está vinculado a um **Jogador**, uma **Unidade**, e um **Tópico**.
5. **Pergunta** pertence a um **Tópico** e está associada a um **Nível de Dificuldade**.

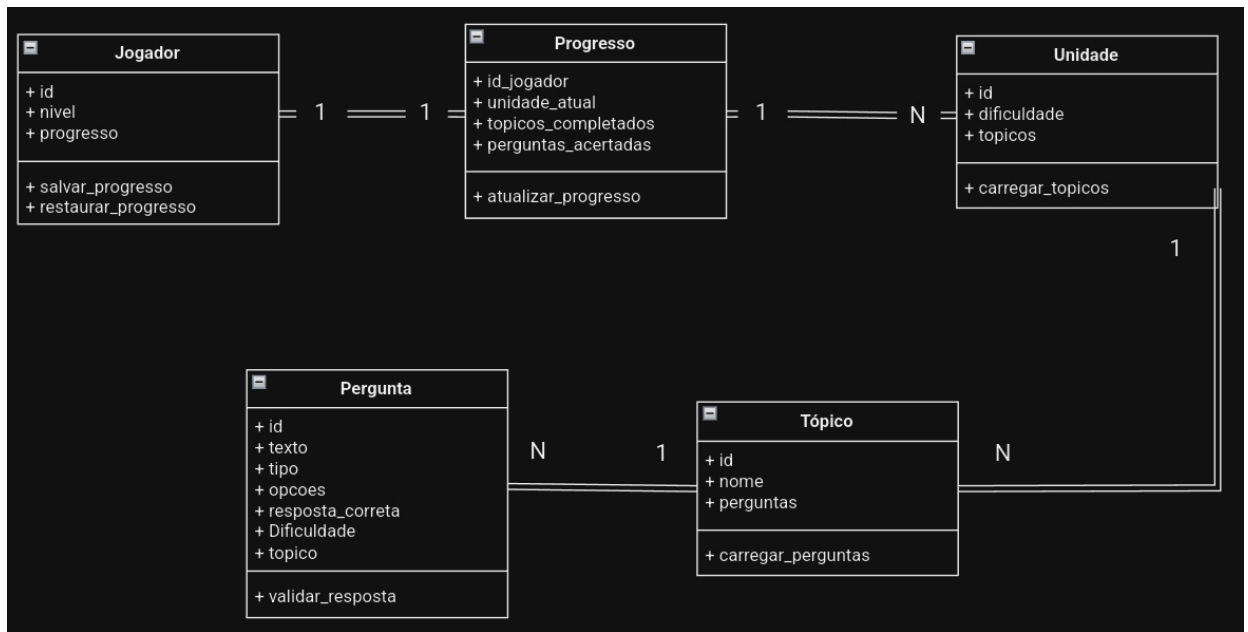
---

#### **Regras de Negócio**

1. Um jogador não pode subir de nível até completar 80% das perguntas do tópico atual.
2. O progresso do jogador é salvo automaticamente após responder cada pergunta.
3. Perguntas respondidas corretamente contam para a conclusão do progresso.
4. O jogador só pode avançar para um novo tópico ou unidade quando conclui todas as perguntas do tópico atual.

#### 9. **Diagrama de Classes e Sequência**

#### **Diagrama de Classes:**



### 1. Classe Pergunta

Atributos:

id: Identificador único da pergunta.

texto: O conteúdo da pergunta.

tipo: O tipo de pergunta (flashcard, múltipla escolha, completar frase).

opcoes: Uma lista de opções para perguntas de múltipla escolha.

resposta\_correta: A resposta correta para a pergunta.

dificuldade: O nível de dificuldade da pergunta (0 a 10).

topico: O tópico ao qual a pergunta pertence (ex: Desenvolvimento, Testes, etc.).

Métodos:

validar\_resposta(resposta): Verifica se a resposta do jogador está correta.

### 2. Classe Jogador

Atributos:

id: Identificador único do jogador (gerado e salvo no LocalStorage).

nivel: O nível atual do jogador.

progresso: Progresso do jogador em termos de unidades e tópicos completados.

Métodos:

salvar\_progresso(): Salva o progresso atual do jogador no DynamoDB.

restaurar\_progresso(): Restaura o progresso do jogador do DynamoDB.

### 3. Classe Progresso

Atributos:

id\_jogador: Identificador do jogador ao qual o progresso pertence.

unidade\_atual: A unidade que o jogador está completando.

topicos\_completados: Lista dos tópicos que o jogador completou.

perguntas\_acertadas: Lista das perguntas que o jogador já acertou.

Métodos:

atualizar\_progresso(): Atualiza o progresso com novas perguntas respondidas.

#### 4. Classe Unidade

Atributos:

id: Identificador da unidade.

dificuldade: Nível de dificuldade da unidade.

topicos: Lista de tópicos dentro dessa unidade.

Métodos:

carregar\_topicos(): Carrega os tópicos disponíveis nessa unidade.

#### 5. Classe Tópico

Atributos:

id: Identificador do tópico.

nome: Nome do tópico (ex: Desenvolvimento, Segurança da Informação, etc.).

perguntas: Lista de perguntas pertencentes ao tópico.

Métodos:

carregar\_perguntas(): Carrega as perguntas do tópico.

### **Explicação do Diagrama:**

1. Jogador: A classe que contém os dados sobre o jogador, como o nível e o progresso. Ela interage com a classe Progresso para salvar e restaurar o progresso no DynamoDB.

2. Progresso: Armazena informações sobre a unidade atual, os tópicos completados e as perguntas que o jogador acertou. Esta classe é responsável por atualizar o progresso conforme o jogador avança no jogo.

3. Pergunta: A classe que define as perguntas do quiz. Cada pergunta tem atributos como o tipo (flashcard, múltipla escolha, etc.), as opções de resposta (caso seja múltipla escolha) e a resposta correta.

4. Unidade: Agrupa os tópicos em diferentes níveis de dificuldade e organiza o jogo por etapas. Cada unidade tem tópicos associados a ela.

5. Tópico: Cada tópico pertence a uma unidade e contém uma lista de perguntas relacionadas ao assunto específico (como desenvolvimento, testes, etc.).

### Relacionamento Explicado em Texto:

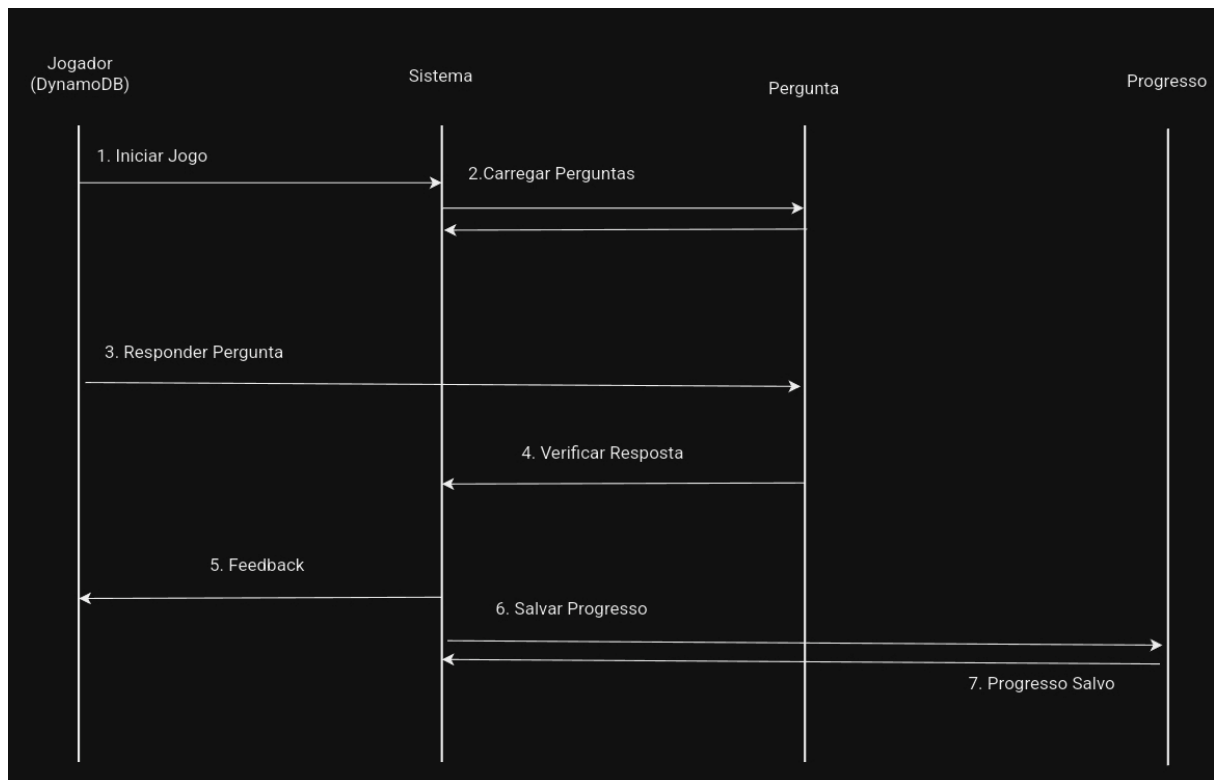
Jogador inicia o jogo, e um Progresso é criado para registrar suas atividades. Este progresso armazena as Unidades que ele já completou.

Cada Unidade é composta por vários Tópicos (ex: Unidade 1 pode conter tópicos sobre Conhecimentos Gerais, Desenvolvimento, etc.).

Os Tópicos possuem um conjunto de Perguntas, que o jogador responde.

O sistema então valida a resposta do Jogador com o método `validar_resposta()` da Pergunta, atualizando o Progresso e permitindo ao jogador avançar para novas unidades e tópicos.

### Diagrama de Sequência:



O diagrama representa a interação entre diferentes elementos do sistema **QuizTech Challenge**, ilustrando como as ações do jogador desencadeiam processos no sistema, desde o início de uma sessão até o salvamento do progresso. Abaixo, cada etapa é detalhada:

#### 1. Iniciar Jogo

- **Descrição:** O jogador inicia o jogo, acessando a interface pela primeira vez ou retornando a uma sessão anterior.
- **Fluxo:**
  - O sistema verifica se há uma **ID anônima** no navegador (LocalStorage).
    - Caso exista, a sessão é recuperada e o progresso salvo anteriormente é carregado.
    - Caso contrário, uma nova ID anônima é gerada e armazenada localmente.

#### 2. Carregar Perguntas

- **Descrição:** O sistema solicita as perguntas apropriadas para o nível e o progresso atual do jogador.
- **Fluxo:**
  - O sistema acessa uma fonte de perguntas (JSON local ou banco de dados).
  - As perguntas são filtradas com base em:
    - Nível do jogador.
    - Unidade e tópico atual.
    - Formato da pergunta (múltipla escolha, flashcard, completar frase).
  - As perguntas selecionadas são carregadas na interface para que o jogador possa respondê-las.

### 3. Responder Pergunta

- **Descrição:** O jogador interage com uma pergunta apresentada pelo sistema e envia uma resposta.
- **Fluxo:**
  - A resposta selecionada ou digitada pelo jogador é enviada ao sistema para validação.
  - O sistema registra a interação para processamento subsequente.

### 4. Verificar Resposta

- **Descrição:** O sistema compara a resposta do jogador com a resposta correta.
- **Fluxo:**
  - O sistema verifica no banco de dados (ou JSON local) se a resposta do jogador está correta.
  - A validação pode incluir regras adicionais, como respostas parcialmente corretas em perguntas discursivas ou preenchimento de lacunas.

### 5. Feedback

- **Descrição:** O sistema fornece um feedback imediato ao jogador com base na resposta enviada.
- **Fluxo:**
  - Se a resposta estiver correta, o sistema exibe uma mensagem de sucesso, por exemplo: "Resposta correta!".
  - Se estiver errada, o sistema exibe um erro, podendo sugerir dicas ou explicações adicionais para ajudar no aprendizado.

### 6. Salvar Progresso

- **Descrição:** Após o jogador responder à pergunta, o sistema registra automaticamente o progresso.
- **Fluxo:**
  - O progresso do jogador (como perguntas respondidas, tópicos concluídos e nível atual) é salvo em duas camadas:

1. **AWS DynamoDB:** O progresso é armazenado no banco de dados remoto, permitindo recuperação em diferentes dispositivos.
2. **LocalStorage:** O progresso é sincronizado localmente no navegador para rápida recuperação em sessões futuras.

## 7. Progresso Salvo

- **Descrição:** O sistema confirma que o progresso foi salvo corretamente.
- **Fluxo:**
  - Uma resposta de sucesso do AWS DynamoDB é recebida pelo sistema.
  - O jogador pode continuar respondendo perguntas sem se preocupar com a perda de progresso.

## Considerações Técnicas

1. **Confiabilidade do Salvamento:** Caso o salvamento no DynamoDB falhe (por exemplo, devido a problemas de conexão), o sistema mantém o progresso localmente no navegador até que a conexão seja restabelecida.
2. **Desempenho:** O carregamento de perguntas e o salvamento do progresso são projetados para serem rápidos e eficientes, otimizando a experiência do jogador.
3. **Feedback Imediato:** Garantir que o jogador receba feedback rápido é fundamental para manter o aprendizado engajante e motivador.

## Resumo

O diagrama reflete o fluxo cíclico e contínuo das interações do jogador com o sistema, onde cada ação é seguida de validação, feedback e salvamento do progresso. Essa estrutura assegura uma experiência fluida, motivadora e tecnicamente robusta, promovendo aprendizado eficiente e seguro.

## 10. Modelos de Dados (JSON)

Exemplo:

```
{
  "unidades": [
    {
      "unidade": 1,
      "dificuldade": [0, 2],
      "topicos": [
        {
          "nome": "Conhecimentos Gerais",
          "perguntas": [
            {
              "id": 1,
              "tipo": "flashcard",
              "pergunta": "O que significa HTTP?",
            }
          ]
        }
      ]
    }
  ]
}
```



```
    "resposta": "HyperText Transfer Protocol"
  },
  {
    "id": 2,
    "tipo": "multipla_escolha",
    "pergunta": "Qual linguagem é mais usada para desenvolvimento web?",
    "opcoes": ["Python", "JavaScript", "Java", "C#"],
    "resposta_correta": "JavaScript"
  },
  {
    "id": 3,
    "tipo": "completar_frase",
    "pergunta": "O comando SQL utilizado para recuperar dados é _____.",
    "resposta_correta": "SELECT"
  }
]
},
{
  "nome": "Testes de Software",
  "perguntas": [
    {
      "id": 4,
      "tipo": "flashcard",
      "pergunta": "O que é TDD?",
      "resposta": "Test Driven Development"
    },
    {
      "id": 5,
      "tipo": "multipla_escolha",
      "pergunta": "Qual ferramenta é usada para automação de testes?",
      "opcoes": ["Selenium", "Postman", "JMeter", "All of the above"],
      "resposta_correta": "All of the above"
    }
  ]
}
```

```
]
}
]
}
}
```

## 11. Guia do Usuário

Bem-vindo ao **QuizTech Challenge**, o jogo educacional que utiliza gamificação para ajudar você a aprender sobre tecnologia de forma divertida e interativa! Este guia explicará como navegar e utilizar o sistema, além de descrever as funcionalidades disponíveis.

### 1. Requisitos para Utilização

- **Navegador Recomendado:** Utilize navegadores atualizados, como Google Chrome, Mozilla Firefox, Microsoft Edge ou Safari.
- **Dispositivo:** Compatível com desktops, laptops, tablets e smartphones.
- **Conexão à Internet:** Necessária para carregar perguntas e salvar progresso.

### 2. Acessando o Jogo

1. Abra o navegador no seu dispositivo.
2. Acesse o site do **QuizTech Challenge**.
3. Clique em **“Iniciar Jogo”** para começar.
  - Não é necessário criar uma conta ou fornecer informações pessoais.
  - O sistema gera automaticamente uma **ID anônima** e salva localmente no navegador para acompanhar seu progresso.

### 3. Navegando pelo Jogo

#### Tela Inicial

- **Botão "Iniciar Jogo":** Clique para acessar as perguntas e começar seu aprendizado.
- **Descrição do Jogo:** Uma breve explicação sobre como o QuizTech Challenge funciona.
- **Configurações** (opcional): Ajuste preferências, como reiniciar progresso.

#### Tela de Perguntas

- Aqui você responderá às perguntas do quiz.
- **Componentes:**
  - Enunciado da pergunta: Aparece no centro da tela.
  - Respostas possíveis:
    - **Múltipla Escolha:** Clique na resposta que você acredita ser correta.
    - **Completar a Frase:** Digite a resposta no campo de texto.
    - **Flashcards:** Clique para visualizar a resposta.
  - **Botão "Enviar Resposta":** Após selecionar ou digitar sua resposta, clique neste botão para continuar.

- **Barra de Progresso:** Mostra seu avanço na unidade atual.

#### **Tela de Feedback**

- Exibe se a resposta está correta ou incorreta.
- Se errar, o sistema pode oferecer dicas ou explicações adicionais.
- Clique em “**Próxima Pergunta**” para continuar.

#### **Tela de Progresso**

- Mostra suas estatísticas gerais:
  - Nível atual.
  - Unidades e tópicos concluídos.
  - Percentual de progresso total.
  - Número de acertos e erros.
- Acesse essa tela automaticamente ao completar um tópico ou manualmente no menu.

### **4. Funcionalidades Especiais**

#### **Subir de Nível**

- Você sobe de nível ao completar perguntas em uma unidade ou tópico.
- As perguntas tornam-se mais desafiadoras à medida que você avança.

#### **Salvar Progresso**

- Seu progresso é salvo automaticamente após cada interação.
  - **LocalStorage:** Salva no navegador, permitindo que você continue mesmo offline.
  - **AWS DynamoDB:** Sincroniza seus dados com o servidor, garantindo que seu progresso esteja seguro, mesmo que troque de dispositivo.

#### **Configurações**

- **Reiniciar Jogo:** Apaga todo o progresso e recomeça do zero.
- **Configurações de Experiência:** Ajustes de volume, dicas ou notificações (se implementado no futuro).

### **5. Dicas para Jogadores**

- **Leia o Feedback:** Mesmo se errar, leia as explicações para aprender com seus erros.
- **Avance no Seu Ritmo:** O jogo é projetado para ser flexível e acessível; você pode parar e retomar quando quiser.
- **Explore Diferentes Tópicos:** Aprenda sobre áreas que você ainda não conhece, como Inteligência Artificial, Redes ou Segurança da Informação.
- **Concentre-se nos Níveis Fáceis Inicialmente:** Para iniciantes, comece em níveis mais baixos antes de avançar para os mais desafiadores.

### **6. Perguntas Frequentes (FAQ)**

#### **Posso jogar sem conexão com a internet?**

- O jogo exige conexão para carregar perguntas e salvar no servidor. No entanto, o progresso local é salvo no navegador mesmo offline.

#### **E se eu perder meu progresso?**

- Desde que você use o mesmo navegador e dispositivo, seu progresso é salvo automaticamente. Caso troque de dispositivo, a sincronização com o servidor (AWS DynamoDB) restaura o progresso.

#### **Preciso criar uma conta?**

- Não. O sistema utiliza uma ID anônima gerada automaticamente.

#### **Como saber meu nível atual?**

- Seu nível é exibido na tela de progresso ou ao final de cada unidade.

#### **O jogo é gratuito?**

- Sim, o **QuizTech Challenge** é gratuito e não exige qualquer pagamento.

### **7. Suporte Técnico**

Caso encontre problemas:

- Verifique sua conexão com a internet.
- Atualize seu navegador para a versão mais recente.
- Entre em contato com o suporte técnico através do e-mail fornecido na página inicial.

### **12. Diagrama de Implantação**

Back End (Lógica) → Cliente – Front End (Navegador) → AWS S3 (Site Estático) → AWS EC2 (Progresso)

\*LocalStorage (ID único) no AWS S3

\*\*JSON (Perguntas e Respostas)

### **13. Plano de Testes**

#### **Objetivo do Plano de Testes:**

Garantir que a aplicação **QuizTech Challenge** funcione corretamente em computadores, validando a funcionalidade, usabilidade e integridade dos dados. Este plano descreve os casos de teste, abordagens, ferramentas e critérios necessários para validar o sistema, focado exclusivamente em ambientes desktop.

#### **1. Estratégia de Testes**

Os testes serão realizados nos seguintes níveis:

1. **Testes de Unidade:** Validar componentes isolados, como carregamento de perguntas e validação de respostas.
2. **Testes de Integração:** Verificar a interação entre diferentes módulos, como carregamento de perguntas e salvamento de progresso.
3. **Testes de Interface (End-to-End):** Testar a navegação e a funcionalidade do sistema completo, utilizando o **Cypress** para automação.

4. **Testes de Aceitação:** Validar se o sistema atende aos requisitos definidos.

## 2. Escopo dos Testes

Os testes cobrirão os seguintes aspectos:

1. Funcionalidades principais:
  - Exibição de perguntas.
  - Envio de respostas e feedback imediato.
  - Salvamento e restauração de progresso.
  - Subida de nível e desbloqueio de tópicos.
2. Usabilidade:
  - Interface funcional e intuitiva em computadores.

## 3. Casos de Teste

### 3.1 Testes Funcionais

#### 1. Carregar Perguntas

- **Cenário:** O sistema deve carregar perguntas baseadas no nível e tópico do jogador.
- **Passos:**
  1. Iniciar o jogo.
  2. Selecionar o nível/tópico atual.
  3. Observar o carregamento das perguntas.
- **Critério de Sucesso:** As perguntas corretas aparecem na interface, com formatação adequada e opções interativas.

#### 2. Enviar Resposta

- **Cenário:** O jogador envia uma resposta válida.
- **Passos:**
  1. Escolher ou digitar uma resposta para a pergunta.
  2. Clicar no botão "**Enviar Resposta**".
- **Critério de Sucesso:** O sistema valida a resposta e exibe o feedback correto (certo ou errado).

#### 3. Subir de Nível

- **Cenário:** O jogador completa todas as perguntas de um tópico.
- **Passos:**
  1. Responder corretamente a todas as perguntas de um tópico.
  2. Observar o sistema atualizar o nível do jogador.
- **Critério de Sucesso:** O nível do jogador aumenta e as perguntas do próximo tópico são carregadas automaticamente.

#### 4. Salvar Progresso

- **Cenário:** O jogador responde a uma pergunta e o progresso é salvo automaticamente.
- **Passos:**
  1. Responder a uma pergunta.
  2. Reiniciar o jogo.
  3. Verificar se o progresso foi recuperado.
- **Critério de Sucesso:** O progresso aparece corretamente ao reiniciar o jogo no mesmo navegador.

### 3.2 Testes de Usabilidade

#### 1. Interface Funcional

- **Cenário:** O jogador utiliza a interface em um navegador de computador.
- **Passos:**
  1. Navegar entre diferentes telas (inicial, perguntas, progresso).
  2. Interagir com botões, links e inputs.
- **Critério de Sucesso:** Todas as funcionalidades são acessíveis e responsivas em telas de computador (resolução mínima 1366x768).

#### 2. Navegação Intuitiva

- **Cenário:** O jogador acessa diferentes funcionalidades sem instruções explícitas.
- **Passos:**
  1. Iniciar o jogo.
  2. Navegar pelas perguntas e pela tela de progresso.
- **Critério de Sucesso:** Todas as ações podem ser realizadas facilmente, com botões e links visíveis.

### 4. Ferramentas e Recursos

- **Ferramentas de Teste Automatizado:**

- **Cypress:** Para realizar testes de interface (end-to-end) e validação da navegação do sistema.
  - Exemplo de cenários testados pelo Cypress:
    - Carregamento da tela inicial.
    - Responder a uma pergunta e verificar o feedback.
    - Verificar se o progresso é salvo e recuperado corretamente.
    - Navegar entre as telas de perguntas e progresso.

- **Ambiente de Teste:**

- Sistemas Operacionais: Windows 10/11, macOS.
- Navegadores Compatíveis: Google Chrome, Mozilla Firefox, Microsoft Edge.
- Resoluções de Tela: 1366x768 (mínimo), 1920x1080 (recomendado).

### 5. Cronograma

1. **Testes de Unidade:** Semana 1.
2. **Testes de Integração:** Semana 2.
3. **Testes de Interface (End-to-End):** Semana 3 (com Cypress).
4. **Testes de Aceitação:** Semana 4.

## 6. Relatórios

Os resultados de cada teste serão documentados em um formato padrão, incluindo:

- Nome do teste.
- Objetivo.
- Passos realizados.
- Resultados esperados.
- Resultados reais.
- Status (Aprovado/Reprovado).
- Comentários adicionais.

## 7. Critérios de Aceitação

O sistema será considerado funcional e pronto para uso se:

1. Todas as funcionalidades principais (carregar perguntas, salvar progresso, etc.) funcionarem sem falhas.
2. A interface for totalmente funcional em navegadores compatíveis.
3. Não houver travamentos ou erros graves durante sessões de uso prolongado.

## 14. Considerações Finais

O **QuizTech Challenge** é uma ferramenta inovadora que combina aprendizado e diversão, utilizando gamificação para ensinar tecnologia de forma desafiadora e envolvente. Com perguntas cuidadosamente organizadas por tópicos e níveis de dificuldade, os jogadores podem progredir gradualmente, consolidando seu conhecimento em áreas essenciais da tecnologia da informação, como desenvolvimento de software, inteligência artificial e segurança cibernética.

A escolha por tecnologias como **AWS S3** para hospedagem e **AWS DynamoDB** para o salvamento automático do progresso oferece uma solução robusta, escalável e acessível, eliminando a necessidade de cadastros ou logins complicados. Essa abordagem garante que o jogo seja simples de usar, leve e acessível a todos os jogadores, sem comprometer a privacidade ou a experiência do usuário.

O potencial de crescimento do **QuizTech Challenge** é vasto. Expansões futuras podem incluir a introdução de novos tópicos como computação quântica e ciência de dados, a implementação de funcionalidades como rankings de jogadores, desafios multiplayer e a personalização do conteúdo para diferentes níveis de conhecimento. Essas melhorias prometem tornar a experiência ainda mais rica, aumentando o engajamento dos jogadores e atendendo a uma audiência ainda maior.

Com sua proposta única, o **QuizTech Challenge** se destaca como uma plataforma educativa moderna, capaz de atender tanto iniciantes quanto profissionais da área de TI, incentivando o aprendizado contínuo de forma prática e divertida. Este projeto demonstra o poder da

gamificação e das tecnologias em nuvem na transformação do aprendizado em algo acessível e motivador.

## 15. Referências

Amazon Web Services (AWS). **Amazon S3 Documentation**. Disponível em: <https://aws.amazon.com/s3/>. Acesso em: 23/11/2024.

Amazon Web Services (AWS). **DynamoDB Documentation**. Disponível em: <https://aws.amazon.com/dynamodb/>. Acesso em: 23/11/2024.

□ W3Schools. **SQL Tutorial**. Disponível em: <https://www.w3schools.com/sql/>. Acesso em: 23/11/2024.

MDN Web Docs. **LocalStorage - Web APIs**. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>. Acesso em: 23/11/2024.

Google Developers. **Web Fundamentals: Responsive Design**. Disponível em: <https://web.dev/responsive-web-design-basics/>. Acesso em: 23/11/2024.

Cypress Documentation. **End-to-End Testing with Cypress**. Disponível em: <https://docs.cypress.io/>. Acesso em: 23/11/2024.

Lighthouse Documentation. **Performance Metrics and Reports**. Disponível em: <https://developers.google.com/web/tools/lighthouse>. Acesso em: 23/11/2024.

AWS Free Tier. **Overview of AWS Free Tier Services**. Disponível em: <https://aws.amazon.com/free/>. Acesso em: 23/11/2024.

Medium. **Introduction to Gamification in Education**. Disponível em: <https://medium.com>. Acesso em: 23/11/2024.

Coursera. **Gamification in Learning and Education**. Disponível em: <https://www.coursera.org/>. Acesso em: 23/11/2024.