### **Levantamento de Requisitos**

#### **1. Objetivo do Jogo**

Proporcionar uma experiência divertida e educativa onde os jogadores testem e ampliem seu conhecimento sobre processos seletivos do SENAI. Além de responder perguntas, os jogadores podem explicar seu raciocínio, ver e avaliar as explicações dos outros, e analisar seu próprio desempenho.

#### **2. Requisitos Funcionais**

##### **2.1. Tela Inicial**

* **Descrição**: Tela com opções de iniciar o jogo, acessar o perfil do usuário, consultar processos salvos e ajustar configurações.
* **Funcionalidades**:
  + **Botão “Iniciar Jogo”**: Leva o usuário para a Tela de Seleção de Processo.
  + **Botão “Perfil”**: Permite que o usuário acesse e edite seu perfil.
  + **Botão “Processos Salvos”**: Exibe os processos seletivos que o usuário salvou ou consultou anteriormente.
  + **Botão “Configurações”**: Acesso às opções de configuração do jogo (som, notificações, etc.).

##### **2.2. Tela de Seleção de Processo**

* **Descrição**: Tela onde o usuário seleciona o processo seletivo do SENAI para responder perguntas.
* **Funcionalidades**:
  + Lista de processos seletivos disponíveis
  + Descrição breve de cada processo seletivo (opcional)
  + Botão para confirmar a seleção (podendo ser o Card)
  + Opção para retornar à Tela Inicial

##### **2.3. Tela de Jogo**

* **Descrição**: Tela principal do jogo onde o jogador responde às perguntas sobre o processo seletivo escolhido.
* **Funcionalidades**:
  + Exibição da pergunta atual e opções de resposta
  + Imagem relacionada a pergunta (se houver)
  + Temporizador para cada pergunta
  + Botões para selecionar a resposta
  + Indicador de pontos acumulados e progresso do jogo
  + Sistema de feedback imediato após a resposta (certa/errada)
  + Interface amigável e responsiva

##### **2.4. Tela de Explicação**

* **Descrição**: Após responder uma pergunta, o usuário pode fornecer uma explicação para sua resposta e visualizar explicações de outros usuários.
* **Funcionalidades**:
  + **Campo de Texto para Explicação**: Permite que o usuário digite uma explicação para a resposta correta.
  + **Visualização de Explicações**: Mostra explicações de outros usuários que acertaram a mesma pergunta.
  + **Sistema de Votação**: Usuários podem votar nas explicações de outros, indicando qual consideram mais clara ou correta.
  + **Ranking de Explicações**: Exibições de explicações classificadas por votos, mostrando as mais confiáveis no topo.

##### **2.5. Tela de Perfil**

* **Descrição**: Tela onde o usuário pode ver e analisar suas informações e desempenho.
* **Funcionalidades**:
  + **Dados Pessoais**: Visualizar e editar informações pessoais.
  + **Análise de Desempenho**:
    - **Estrelas de Explicação**: Exibir um resumo das estrelas recebidas pelas explicações do usuário.
    - **Taxa de Acertos e Erros**: Percentual de respostas corretas dividido pelo total de respostas.
    - **Número de Provas Concluídas**: Total de provas (ou jogos) que o usuário completou.
    - **Histórico de Pontuação**: Exibir um histórico de pontuações e desempenho em jogos anteriores.

##### **2.6. Banco de Perguntas**

* **Descrição**: Conjunto de perguntas relacionadas aos processos seletivos do SENAI.
* **Funcionalidades**:
  + Sistema para importar e atualizar perguntas
  + Classificação das perguntas por dificuldade e categorias
  + Banco de dados seguro e acessível para fácil atualização

##### **2.7. Sistema de Pontuação**

* **Descrição**: Sistema para calcular e exibir a pontuação do jogador.
* **Funcionalidades**:
  + Pontuação baseada na resposta correta e tempo de resposta
  + Exibição de pontuação total ao final de cada jogo
  + Histórico de pontuações e ranking

##### **2.8. Configurações do Jogo**

* **Descrição**: Opções para personalizar a experiência do jogo.
* **Funcionalidades**:
  + Configurações de som e música
  + Preferências de notificações
  + Ajustes de dificuldade

#### **3. Requisitos Não Funcionais**

##### **3.1. Usabilidade**

* Interface intuitiva e fácil de navegar
* Suporte para múltiplas plataformas (web, mobile)

##### **3.2. Desempenho**

* O jogo deve carregar rapidamente e ser responsivo
* A aplicação deve suportar múltiplos usuários simultaneamente sem degradação de desempenho

##### **3.3. Segurança**

* Proteção de dados pessoais dos usuários
* Sistema seguro para armazenamento e manipulação das perguntas e respostas

##### **3.4. Acessibilidade**

* O jogo deve ser acessível para pessoas com deficiências (ex.: compatibilidade com leitores de tela, contraste adequado)

### **4. Requisitos Técnicos**

#### **4.1. Plataforma**

* **Desenvolvimento Móvel**:
  + **React Native**: Framework para criar aplicativos móveis nativos para iOS e Android e Web a partir de uma única base de código.
  + **TypeScript**: Adiciona tipagem estática ao JavaScript, melhorando a robustez e a manutenção do código.
* **Desenvolvimento Backend**:
  + **Java com Spring Framework** ou **C# com .NET**:
    - **Spring Framework (Java)**: Framework robusto para construir aplicações empresariais, incluindo APIs RESTful.
    - **ASP.NET Core (C#)**: Framework para desenvolvimento de APIs RESTful e aplicações web, com forte integração com o .NET Ecosystem.
* **Banco de Dados**:
  + **MySQL**: Sistema de gerenciamento de banco de dados relacional (RDBMS) para armazenar dados estruturados como perguntas, respostas, usuários e pontuações.

#### **4.2. Tecnologias e Ferramentas**

* **Frontend**:
  + **React Native**: Utilizado para o desenvolvimento do frontend móvel.
  + **TypeScript**: Para garantir uma codificação mais segura e manutenível.
  + **React Navigation**: Biblioteca para gerenciamento de navegação no aplicativo.
  + **Axios** : Para comunicação com o backend via HTTP.
  + **Expo**: Um conjunto de ferramentas e serviços para o desenvolvimento com React Native.
* **Backend**:
  + **Spring Framework (Java)** ou **ASP.NET Core (C#)**:
    - **Spring Boot**: Para facilitar a configuração e a inicialização do projeto Spring.
    - **Spring Security**: Para implementar autenticação e autorização (para Spring Framework).
    - **Entity Framework Core**: Para manipulação de dados e mapeamento objeto-relacional (para .NET).
    - **ASP.NET Identity**: Para autenticação e gerenciamento de usuários (para .NET).
* **Banco de Dados**:
  + **MySQL**:
    - **MySQL Workbench**: Para modelagem de banco de dados e gerenciamento.
    - **JPA/Hibernate**: Para mapeamento objeto-relacional (Java).

#### **4.3. Ambiente de Desenvolvimento**

* **Frontend**:
  + **Editor/IDE**: Visual Studio Code, ou outro editor com suporte para TypeScript e React Native.
  + **Emuladores/Simuladores**: Android Studio, para a simulção de um mobile.
* **Backend**:
  + **IDE**: IntelliJ IDEA (para Java), Visual Studio (para C#), ou outro IDE preferido.
  + **Servidor de Aplicação**: Tomcat (para Spring).
  + **Testes de API**: Postman ou Insomnia para testar endpoints da API.
* **Banco de Dados**:
  + **MySQL Server**: Configuração do servidor de banco de dados.
  + **Ferramentas de Backup e Recuperação**: Para garantir a integridade e a recuperação de dados.
  + **Ferramentas de Monitoramento**: Monitorar a performance e o uso do banco de dados.

#### **4.4. Integração**

* **APIs**:
  + **RESTful APIs**: Para comunicação entre o frontend e o backend.
  + **Swagger/OpenAPI**: Para documentação das APIs e testes interativos.
* **Autenticação e Autorização**:
  + **JWT (JSON Web Tokens)**: Para troca segura de informações entre cliente e servidor.
* **Versionamento de Código**:
  + **Git**: Para controle de versão.
  + **GitHub/GitLab/Bitbucket**: Para hospedagem e colaboração em código.

#### **4.5. Performance e Escalabilidade**

* **Frontend**:
  + **Otimização de Performance**: Uso de técnicas como lazy loading e otimização de imagens.
  + **Análise de Performance**: Ferramentas como React DevTools e perfis de desempenho em dispositivos móveis.
* **Backend**:
  + **Escalabilidade**: Implementar práticas para escalar horizontalmente, como uso de balanceadores de carga e microservices (se necessário).
  + **Cache**: Utilizar caching para melhorar a performance de acesso aos dados frequentemente consultados.
* **Banco de Dados**:
  + **Indexação**: Criar índices apropriados para melhorar a performance das consultas.
  + **Sharding e Replicação**: Considerar estratégias para distribuir a carga e melhorar a disponibilidade dos dados.

#### **5. Testes e Validação**

##### **5.1. Testes Funcionais**

* Verificar se todas as funcionalidades estão operando conforme especificado
* Testes de explicações e sistema de votação

##### **5.2. Testes de Usabilidade**

* Testar a interface com usuários reais para garantir que seja intuitiva e fácil de usar

##### **5.3. Testes de Desempenho**

* Avaliar o desempenho do jogo em diferentes dispositivos e condições de rede

#### **6. Documentação e Suporte**

##### **6.1. Documentação do Usuário**

* Manual de usuário com instruções sobre como jogar, fornecer explicações, votar e acessar informações do perfil

##### **6.2. Suporte Técnico**

* Sistema de suporte para resolver problemas técnicos e responder a perguntas dos usuários