Strexam

Sistema de Gerenciamento de Exames

Documentação Técnica

Akles Pires Camoleze

3 de julho de 2025

Conteúdo

1 Visão Geral

Strexam é uma aplicação completa para criação, gerenciamento e participação em exames online. O sistema permite que professores e instrutores criem exames com diferentes tipos de questões, enquanto os alunos podem participar dos exames e receber feedback imediato sobre seu desempenho.

2 Arquitetura

O projeto é composto por duas partes principais:

1. **Backend (exam-api)**: Uma API REST desenvolvida com Java e Spring Boot, seguindo o paradigma de programação reativa.
2. **Frontend (exam_app)**: Uma aplicação móvel multiplataforma desenvolvida com Flutter.

2.1 Detalhes da Arquitetura

2.1.1 Backend (exam-api)

O backend do Strexam é construído com uma arquitetura moderna e reativa, utilizando Spring WebFlux para fornecer uma API REST não-bloqueante. A aplicação segue o padrão de arquitetura em camadas:

- **Controladores**: Responsáveis por receber as requisições HTTP e delegar o processamento para os serviços apropriados.
- **Serviços**: Contêm a lógica de negócio da aplicação.
- **Repositórios**: Responsáveis pela comunicação com o banco de dados.
- **Modelos**: Representam as entidades do domínio.
- **DTOs (Data Transfer Objects)**: Utilizados para transferência de dados entre as camadas.

2.1.2 Frontend (exam_app)

O frontend é uma aplicação móvel multiplataforma desenvolvida com Flutter, que permite a execução em dispositivos Android e iOS a partir de uma única base de código. A arquitetura do frontend segue o padrão de gerenciamento de estado Provider, que facilita a comunicação entre os componentes da aplicação. A aplicação é organizada em:

- **Screens**: Telas da interface do usuário.
- **Widgets**: Componentes reutilizáveis da interface.
- **Providers**: Gerenciadores de estado que fornecem dados para as telas.

- **Services**: Serviços para comunicação com a API e armazenamento local.
- **Models**: Representações dos dados da aplicação.

2.1.3 Comunicação entre Backend e Frontend

A comunicação entre o backend e o frontend é realizada através de requisições HTTP REST e Server-Sent Events (SSE) para streaming de dados em tempo real. O frontend utiliza o padrão de autenticação JWT (JSON Web Tokens) para autenticar as requisições ao backend.

2.2 Tecnologias Utilizadas

2.2.1 Backend

- **Linguagem**: Java 17
- **Framework**: Spring Boot 3.x
- **Programação Reativa**: Project Reactor (Mono/Flux)
- **Autenticação**: JWT (JSON Web Tokens)
- **Banco de Dados**: PostgreSQL com suporte a operações reativas via R2DBC
- **Migração de Banco de Dados**: Flyway
- **Logging**: SLF4J
- **Testes**: JUnit 5, Mockito, WebTestClient

2.2.2 Frontend

- **Framework**: Flutter 3.x
- **Linguagem**: Dart 3.x
- **Gerenciamento de Estado**: Provider
- **Armazenamento Local**: StorageService (Shared Preferences)
- **HTTP Client**: Dio
- **Injeção de Dependência**: GetIt
- **Testes**: Flutter Test, Mockito

3 Banco de Dados

O Strexam utiliza PostgreSQL como banco de dados relacional, com acesso reativo através do R2DBC (Reactive Relational Database Connectivity). A estrutura do banco de dados é composta pelas seguintes tabelas principais:

3.1 Tabelas Principais

3.1.1 users

Armazena informações dos usuários do sistema.

- `id`: Identificador único do usuário (chave primária)
- `username`: Nome de usuário único
- `email`: Email único do usuário
- `full_name`: Nome completo do usuário
- `password`: Senha criptografada do usuário
- `created_at`: Data de criação do registro
- `updated_at`: Data de atualização do registro

3.1.2 exams

Armazena informações sobre os exames criados.

- `id`: Identificador único do exame (chave primária)
- `title`: Título do exame
- `description`: Descrição detalhada do exame
- `host_user_id`: ID do usuário que criou o exame (chave estrangeira para users)
- `join_code`: Código único para acesso ao exame
- `status`: Status do exame (DRAFT, ACTIVE, COMPLETED, CANCELLED)
- `time_limit`: Tempo limite para realização do exame (em minutos)
- `allow_retake`: Indica se o exame permite múltiplas tentativas
- `created_at`: Data de criação do registro
- `updated_at`: Data de atualização do registro

3.1.3 questions

Armazena as questões dos exames.

- `id`: Identificador único da questão (chave primária)
- `exam_id`: ID do exame ao qual a questão pertence (chave estrangeira para exams)
- `question_text`: Texto da questão
- `type`: Tipo da questão (MULTIPLE_CHOICE, TRUE_FALSE, SHORT_ANSWER)
- `order_index`: Ordem da questão no exame
- `points`: Pontuação da questão

3.1.4 answers

Armazena as respostas possíveis para questões de múltipla escolha.

- `id`: Identificador único da resposta (chave primária)
- `question_id`: ID da questão à qual a resposta pertence (chave estrangeira para questions)
- `answer_text`: Texto da resposta
- `is_correct`: Indica se a resposta está correta
- `order_index`: Ordem da resposta na questão

3.1.5 exam_sessions

Armazena as sessões de exame dos usuários.

- `id`: Identificador único da sessão (chave primária)
- `exam_id`: ID do exame (chave estrangeira para exams)
- `user_id`: ID do usuário (chave estrangeira para users)
- `status`: Status da sessão (STARTED, IN_PROGRESS, COMPLETED, ABANDONED)
- `started_at`: Data de início da sessão
- `completed_at`: Data de conclusão da sessão
- `total_score`: Pontuação total obtida
- `max_score`: Pontuação máxima possível
- `created_at`: Data de criação do registro
- `updated_at`: Data de atualização do registro

3.1.6 user_responses

Armazena as respostas dos usuários às questões.

- `id`: Identificador único da resposta (chave primária)
- `session_id`: ID da sessão de exame (chave estrangeira para exam_sessions)
- `question_id`: ID da questão (chave estrangeira para questions)
- `answer_id`: ID da resposta escolhida (chave estrangeira para answers, para questões de múltipla escolha)
- `response_text`: Texto da resposta (para questões de resposta curta)

- `is_correct`: Indica se a resposta está correta
- `points_earned`: Pontos obtidos com a resposta
- `responded_at`: Data da resposta

3.2 Índices

O banco de dados utiliza diversos índices para otimizar o desempenho das consultas:

- Índices em chaves estrangeiras para melhorar o desempenho de junções
- Índices em campos frequentemente utilizados em filtros, como `join_code` em exams
- Índices compostos para consultas específicas de alto desempenho

3.3 Migrações

O esquema do banco de dados é gerenciado através do Flyway, que executa scripts de migração em ordem sequencial para criar e atualizar o esquema. Os principais scripts de migração são:

1. `V1__create_schema.sql`: Cria as tabelas principais e índices
2. `V2__create_functions.sql`: Cria funções SQL personalizadas
3. `V3__create_data_and_views.sql`: Cria dados iniciais e views
4. `V4__add_password_column.sql`: Adiciona a coluna de senha à tabela de usuários

4 Funcionalidades Principais

4.1 Gerenciamento de Usuários

- **Registro e autenticação de usuários**: Sistema completo de registro e login com autenticação JWT.
- **Perfis de usuário**: Suporte a diferentes perfis (alunos e professores/instrutores) com permissões específicas.
- **Gerenciamento de informações de perfil**: Atualização de dados pessoais, email e senha.
- **Segurança**: Senhas armazenadas com criptografia bcrypt e autenticação via tokens JWT.

4.2 Criação e Gerenciamento de Exames

- **Criação de exames**: Interface intuitiva para criação de exames com título, descrição e configurações avançadas.
- **Editor de questões**: Suporte a diferentes tipos de questões:
  - Questões de múltipla escolha
  - Questões de verdadeiro ou falso
  - Questões abertas com correção manual
- **Configurações de exame**: Definição de tempo limite, pontuação por questão e possibilidade de múltiplas tentativas.
- **Ativação/desativação de exames**: Controle sobre quando o exame está disponível para os participantes.
- **Compartilhamento de exames**: Geração automática de códigos de acesso para compartilhamento fácil com os participantes.
- **Gerenciamento de sessões**: Visualização e gerenciamento de todas as sessões de exame em andamento ou concluídas.

4.3 Participação em Exames

- **Ingresso em exames**: Acesso simplificado através de código de acesso único.
- **Interface adaptativa**: Interface intuitiva que se adapta ao tipo de questão sendo respondida.
- **Navegação entre questões**: Facilidade para navegar entre as questões do exame.
- **Submissão de respostas**: Envio de respostas em tempo real com feedback imediato quando aplicável.
- **Controle de tempo**: Exibição do tempo restante e alertas de tempo.
- **Finalização de sessões**: Processo seguro para finalizar o exame e submeter todas as respostas.
- **Visualização de resultados**: Acesso imediato aos resultados após a conclusão do exame.

4.4 Estatísticas e Análise

- **Dashboard em tempo real**: Estatísticas atualizadas em tempo real durante a realização do exame.
- **Análise de questões**:
  - Identificação de questões mais difíceis com base na taxa de erro

  - Identificação de questões com maior taxa de acerto
  - Tempo médio gasto por questão
- **Análise de desempenho**:
  - Ranking de desempenho dos participantes
  - Distribuição de pontuações
  - Progresso individual em exames
- **Visualizações gráficas**: Gráficos e visualizações para facilitar a interpretação dos dados.
- **Exportação de dados**: Possibilidade de exportar estatísticas para análise externa.

4.5 Correção e Feedback

- **Correção automática**: Avaliação instantânea para questões de múltipla escolha e verdadeiro/falso.
- **Correção manual**: Interface para professores corrigirem questões de resposta curta.
- **Feedback detalhado**: Informações sobre respostas corretas e incorretas.
- **Comentários personalizados**: Possibilidade de adicionar comentários específicos às respostas dos alunos.
- **Revisão de exame**: Alunos podem revisar suas respostas e feedback após a conclusão do exame.

5 API REST

Abaixo estão os principais endpoints disponíveis:

5.1 Autenticação

- `POST /api/auth/register`: Registra um novo usuário no sistema.
- `POST /api/auth/login`: Autentica um usuário e retorna um token JWT.

5.2 Usuários

- `GET /api/users/{id}`: Obtém informações de um usuário específico.
- `GET /api/users/username/{username}`: Busca um usuário pelo nome de usuário.
- `GET /api/users`: Lista todos os usuários (com paginação).
- `PUT /api/users/{id}`: Atualiza informações de um usuário.
- `DELETE /api/users/{id}`: Remove um usuário do sistema.

5.3 Exames

- `POST /api/exams`: Cria um novo exame.
- `GET /api/exams/{examId}`: Obtém detalhes de um exame específico.
- `GET /api/exams/host/{hostUserId}`: Lista exames criados por um usuário específico.
- `GET /api/exams/participant/{userId}`: Lista exames em que um usuário participou.
- `POST /api/exams/join`: Permite um usuário ingressar em um exame usando um código de acesso.
- `PUT /api/exams/{examId}/activate`: Ativa um exame para participação.

5.4 Sessões de Exame

- `GET /api/exams/{examId}/sessions`: Lista todas as sessões de um exame.
- `GET /api/exams/participant/{userId}/sessions`: Lista sessões de exame de um participante.
- `PUT /api/exams/sessions/{sessionId}/complete`: Finaliza uma sessão de exame.
- `GET /api/exams/sessions/{sessionId}/responses`: Obtém as respostas de uma sessão de exame.

5.5 Respostas

- `POST /api/exams/answer`: Submete uma resposta para uma questão.
- `PUT /api/exams/responses/{responseId}/correct`: Atualiza a correção de uma resposta de texto livre.

5.6 Estatísticas

- `GET /api/exams/{examId}/statistics`: Stream de estatísticas gerais do exame (SSE).
- `GET /api/exams/{examId}/statistics/difficult-questions`: Stream das questões mais difíceis (SSE).
- `GET /api/exams/{examId}/statistics/correct-questions`: Stream das questões com maior taxa de acerto (SSE).
- `GET /api/exams/{examId}/statistics/top-performers`: Stream dos participantes com melhor desempenho (SSE).
- `GET /api/exams/sessions/{sessionId}/progress`: Obtém o progresso de uma sessão de exame.

5.7 Streaming de Eventos

- `GET /api/stream/exams/{examId}`: Stream de eventos de um exame específico (SSE).
- `GET /api/stream/exams`: Stream de eventos de todos os exames (SSE).

6 Estrutura do Projeto

6.1 Backend (exam-api)

O backend segue uma estrutura de pacotes organizada por funcionalidade:

```
exam-api/
          src/
                main/
                      java/
                            com/
                                camoleze/
                                    examapi/
                                        config/           # Configura  es da aplica  o
                                        controller/       # Controladores REST
                                        dto/              # Objetos de transfer ncia de dados
                                        exception/        # Exce  es personalizadas
                                        model/            # Entidades de dom nio
                                        repository/       # Reposit rios para acesso a dados
                                        security/         # Configura  es de seguran a
                                        service/          # Servi os com l gica de neg cio
                                        ExamApiApplication.java  # Classe principal
                      resources/
                          application.yml               # Configura  es da aplica  o
                          db/
                              migrations/               # Scripts de migra  o do Flyway
                test/                                 # Testes automatizados
          pom.xml                                   # Configura  o do Maven
```

Listing 1: Estrutura do Backend

6.1.1 Controladores

- **AuthController**: Gerencia autenticação e registro de usuários.

- **UserController**: Gerencia operações relacionadas a usuários (CRUD).
- **ExamController**: Gerencia operações relacionadas a exames, questões, respostas e sessões.
- **StreamController**: Gerencia streams de eventos em tempo real usando Server-Sent Events (SSE).
- **GlobalExceptionHandler**: Tratamento centralizado de exceções da API.

6.1.2 Serviços

- **UserService**: Lógica de negócio para gerenciamento de usuários, incluindo autenticação.
- **ExamService**: Lógica de negócio para gerenciamento de exames, questões, respostas e sessões.
- **StatisticsService**: Cálculo e fornecimento de estatísticas em tempo real sobre exames e participantes.

6.1.3 Repositórios

- **UserRepository**: Acesso a dados de usuários.
- **ExamRepository**: Acesso a dados de exames.
- **QuestionRepository**: Acesso a dados de questões.
- **AnswerRepository**: Acesso a dados de respostas possíveis.
- **ExamSessionRepository**: Acesso a dados de sessões de exame.
- **UserResponseRepository**: Acesso a dados de respostas dos usuários.

6.1.4 Segurança

- **SecurityConfig**: Configuração de segurança da aplicação.
- **JwtUtil**: Utilitário para geração e validação de tokens JWT.
- **JwtAuthenticationConverter**: Conversor para autenticação baseada em JWT.
- **ReactiveUserDetailsService**: Serviço para carregamento de detalhes de usuário de forma reativa.

6.2 Frontend (exam_app)

O frontend segue uma estrutura de diretórios organizada por funcionalidade:

```
exam_app/
          lib/
                config/           # Configura  es da aplica  o
                core/             # Funcionalidades e utilidades
   principais
                      exceptions/   # Exce  es personalizadas
                      types/        # Tipos e enums
                mixins/           # Mixins reutiliz veis
                models/           # Modelos de dados
                providers/        # Provedores de estado
                screens/          # Telas da interface do usu rio
                services/         # Servi os para comunica  o com API
                transformers/     # L gica de transforma  o de dados
                utils/            # Fun  es utilit rias
                widgets/          # Componentes de UI reutiliz veis
                      common/       # Widgets comuns
                      exam/         # Widgets espec ficos para exames
                      statistics/   # Widgets para estat sticas
                main.dart         # Ponto de entrada da aplica  o
          pubspec.yaml          # Configura  o de depend ncias
          test/                 # Testes automatizados
```

Listing 2: Estrutura do Frontend

6.2.1 Telas Principais

- **LoginScreen**: Tela de login e registro.
- **HomeScreen**: Tela principal com acesso às funcionalidades.
- **CreateExamScreen**: Interface para criação e edição de exames.
- **ExamDetailsScreen**: Visualização detalhada de um exame.
- **JoinExamScreen**: Interface para ingressar em um exame.
- **ExamScreen**: Interface para realização de um exame.
- **ExamResultsScreen**: Visualização de resultados de um exame.
- **StatisticsScreen**: Visualização de estatísticas em tempo real.
- **SessionsListScreen**: Lista de sessões de exame.
- **SessionResponsesScreen**: Visualização de respostas de uma sessão.
- **SessionCorrectionScreen**: Interface para correção manual de respostas.
- **UserSessionsScreen**: Lista de sessões de exame de um usuário.

6.2.2 Provedores de Estado

- **AuthProvider**: Gerenciamento de estado de autenticação.
- **ExamProvider**: Gerenciamento de estado relacionado a exames.
- **StatisticsProvider**: Gerenciamento de estado para estatísticas.

6.2.3 Serviços

- **ApiService**: Comunicação com a API backend.
- **StorageService**: Armazenamento local de dados.
- **AuthService**: Serviço de autenticação.
- **ExamService**: Serviço para operações relacionadas a exames.
- **StatisticsService**: Serviço para obtenção de estatísticas.

7 Configuração e Instalação

7.1 Requisitos

7.1.1 Backend

- Java 17 ou superior
- Maven 3.8 ou superior
- PostgreSQL 13 ou superior

7.1.2 Frontend

- Flutter 3.x
- Dart 3.x
- Android Studio ou VS Code com plugins Flutter

7.2 Configuração do Banco de Dados

Crie um banco de dados PostgreSQL:

```
CREATE DATABASE examdb;
CREATE USER examuser WITH ENCRYPTED PASSWORD 'exampass';
GRANT ALL PRIVILEGES ON DATABASE examdb TO examuser;
```

Listing 3: Criação do Banco de Dados

As migrações do Flyway serão executadas automaticamente na inicialização da aplicação.

7.3 Configuração do Backend

Clone o repositório:

```
git clone https://github.com/camoleze/strexam.git
cd strexam
```

Listing 4: Clone do Repositório

Configure as propriedades da aplicação em `exam-api/src/main/resources/application.yml`:

```
spring:
  r2dbc:
    url: r2dbc:postgresql://localhost:5432/examdb
    username: examuser
    password: exampass
```

Listing 5: Configuração da Aplicação

Compile e execute o backend:

```
cd exam-api
mvn clean install
mvn spring-boot:run
```

Listing 6: Execução do Backend

O backend estará disponível em http://localhost:9000.

7.4 Configuração do Frontend

Configure o arquivo de ambiente em `exam_app/lib/config/app-config.dart` com o endereço do backend:

```
class AppConfig {
  static const String baseUrl = 'http://10.0.2.2:9000/api' // localhost;
}
```

Listing 7: Configuração do Frontend

Instale as dependências e execute o aplicativo:

```
cd exam_app
flutter pub get
flutter run
```

Listing 8: Execução do Frontend

7.5 Execução de Testes

7.5.1 Backend

```
cd exam-api
mvn test
```

7.5.2 Frontend

```
cd exam_app
flutter test
```

8 Fluxo de Uso Típico

8.1 Professor

1. Registra-se e faz login no sistema
2. Cria um novo exame com questões e respostas
3. Ativa o exame e obtém um código de acesso
4. Compartilha o código com os alunos
5. Monitora estatísticas em tempo real durante o exame
6. Corrige questões de resposta curta após a conclusão

8.2 Aluno

1. Registra-se e faz login no sistema
2. Ingressa em um exame usando o código de acesso
3. Responde às questões do exame
4. Recebe feedback imediato para questões de múltipla escolha
5. Completa o exame e visualiza seu desempenho

9 Conclusão

Strexam é uma proposta para o gerenciamento de exames online, oferecendo funcionalidades tanto para professores quanto para alunos. A arquitetura moderna, com backend reativo e frontend em Flutter, proporciona uma experiência responsiva e eficiente para todos os usuários.

9.1 Possíveis Melhorias Futuras

- **Suporte a Mais Tipos de Questões**: Adicionar suporte para questões de arrastar e soltar, preenchimento de lacunas, etc.
- **Integração com LMS**: Integração com sistemas de gerenciamento de aprendizagem como Moodle, Canvas, etc.
- **Análise Avançada de Dados**: Implementação de algoritmos de aprendizado de máquina para análise de desempenho e recomendações personalizadas.
- **Modo Offline**: Permitir a realização de exames sem conexão com a internet, sincronizando os dados quando a conexão for restabelecida.

- **Acessibilidade**: Melhorias na acessibilidade para usuários com necessidades especiais.
- **Internacionalização**: Suporte a múltiplos idiomas.
- **Versão Web**: Desenvolvimento de uma versão web do aplicativo para acesso via navegador.
- **Integração com IA**: Integrar inteligência artificial para correção automática de questões abertas e feedback personalizado.