

Sistema de Avaliação de Provas Objetivas (SAPO)

Documento de Arquitetura

LUIZ FELIPE PIRES
ROBERT CARDOSO MARTINS
VITOR LIMA RIBEIRO

**GOIÂNIA - GO
2023**

SUMÁRIO

1. Introdução	2
2. Contexto Arquitetural	2
2.1. Funcionalidades e Restrições	2
2.2. Atributos de Qualidade prioritários	3
3. Representação	3
3.1. Metamodelo de componentes	3
3.1.1. Componentes e Camadas	3
3.1.2. Diagrama	5
3.2. Data Flow Diagram (DFD)	5
3.2.1. Front-End	5
3.2.2. Back-End	6

1. Introdução

Com base nos requisitos coletados pelo Documento de Requisitos do SAPO, este documento delibera sobre seu Projeto de Arquitetura e suas particularidades. O objetivo é fornecer uma contextualização arquitetural e representações gráficas do modelo obtido. Portanto, este artefato é orientado aos Desenvolvedores.

2. Contexto Arquitetural

Neste tópico, os requisitos funcionais e não funcionais coletados são esmiuçados de modo a definir as funcionalidades e restrições do sistema. Em seguida, definem-se os atributos de qualidade a serem priorizados diante das necessidades apontadas.

Em síntese, nesta seção realizam-se as decisões de arquitetura do sistema.

2.1. Funcionalidades e Restrições

Está explícito nos *RNF-03* e *RNF-04* que a interação dos atores com o sistema se dá num contexto **Web**, enquanto que os *RF-03*, *RF-04* e *RF-05* impõe a existência de componentes independentes no sistema, e suas comunicações com o cliente. Assim, entende-se que o estilo arquitetural em camadas em módulos, e seguindo uma lógica **Cliente-Servidor** é a escolha acertada.

Os *RF-01*, *RF-02* e *RF-06* descrevem uma estrutura que seria explorada numa arquitetura **REST**, com requisições e consultas frequentes ao servidor.

Por fim, os *RNF-05* e *RF-07* tornam indispensáveis a adoção de módulos independentes no sistema, como já apontado, de forma que se possa tirar mais proveito da especialização.

2.2. Atributos de Qualidade prioritários

Por conta da natureza do problema enfrentado, os principais atributos de qualidade a serem priorizados nesse sistema são os de funcionalidade, eficiência, confiabilidade e segurança.

A corretude funcional é fundamental para módulo de reconhecimento de texto, que produzirá os resultados a serem avaliados pelo restante do sistema.

Além disso, por haver manipulação de dados sensíveis, a confidencialidade e o não repúdio também devem estar em observância.

Por fim, é necessário que se atente à tolerância a falhas e utilização de recursos no lado do servidor, já que serão processadas até uma centena de milhar de provas.

3. Representação

Partindo, pois, do referencial obtido no tópico anterior, faz-se necessário representar o sistema. Primeiramente, elabora-se um metamodelo de componentes para modelar os blocos básicos abstratos do projeto de arquitetura de modo que possam ser refinados em classes, a nível de implementação. Em seguida, é apresentado um diagrama de fluxo de dados, que especifica a comunicação entre os componentes. Na primeira parte é produzido um diagrama.

3.1. Metamodelo de componentes

É importante notar que nesse modelo dá-se uma visão mais geral da comunicação entre os componentes, sendo ela especificada posteriormente.

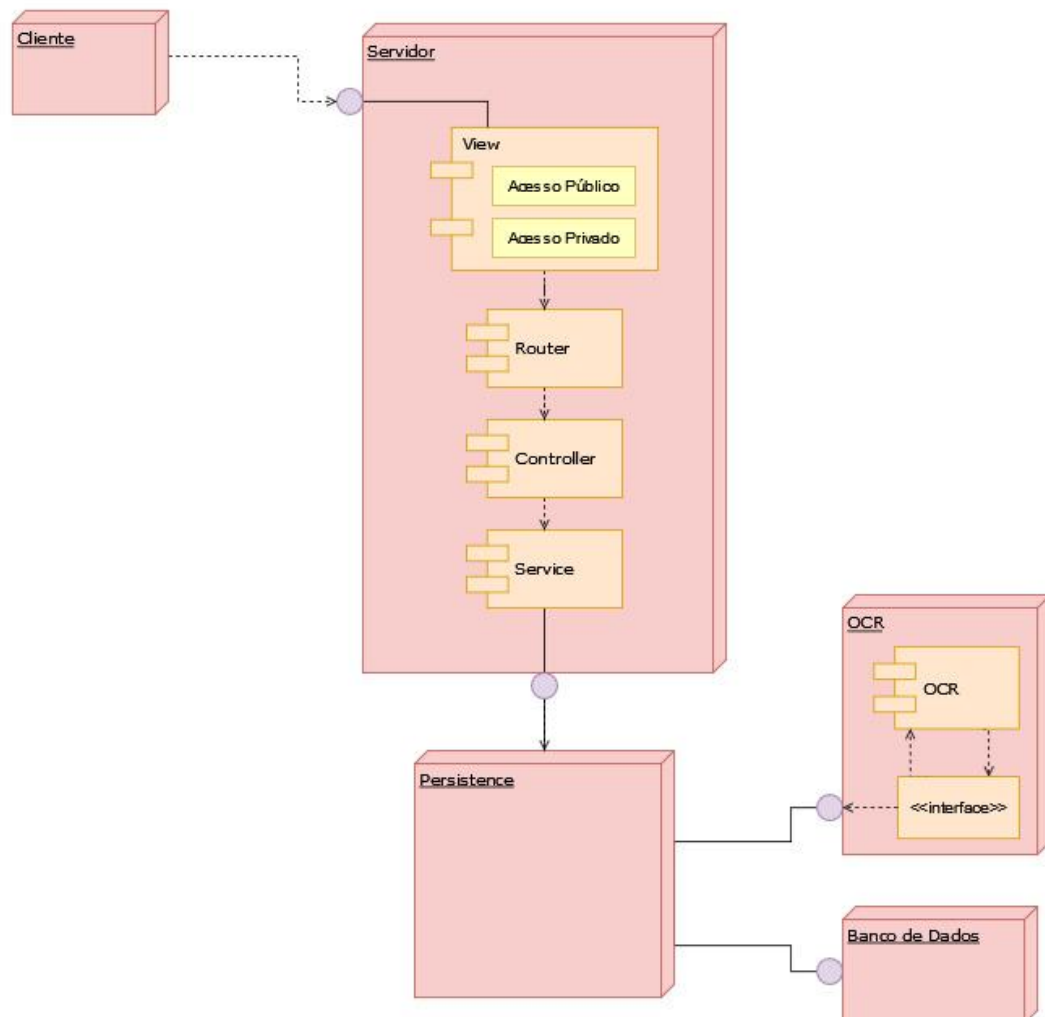
3.1.1. Componentes e Camadas

- ❖ Cliente: um navegador de um usuário (Administrador do Banco de Dados e Candidato), que possibilita o acesso ao Front-End do sistema.
- ❖ View: a aplicação estática competente por entregar a visão que permite interação com o servidor do sistema e suas funcionalidades. É dividido em acesso privado e público. Para acessar a interface privada, é necessário possuir login no sistema. O acesso público está

disponível para qualquer cliente que acessar a plataforma. Ainda, realiza a comunicação com o servidor da aplicação.

- ❖ Server: define as Regras de Negócio do sistema. Ele é acessado pelo componente View, executa as funcionalidades necessárias quando requisitado. Ainda, na requisição, pode realizar o processamento de dados enviados pela View e retornar as informações esperadas. Esse componente está dividido em Router, Controller e Service.
- ❖ Router: intercepta requisições à aplicação. O acesso ao servidor é estabelecido através de endpoints. O Router é o encarregado por interceptar a requisição e acionar o Controller em questão. Possível verificar a requisição e assegurar que é uma requisição segura. Ou seja, o Router também é responsável por parte da segurança do sistema, principalmente quando se diz respeito à autenticação de usuários.
- ❖ Controller: recebe as requisições, verifica os conteúdos do corpo de requisição, aciona os respectivos Services e retorna o resultado do processo.
- ❖ Service: processamento de uma requisição. Em um serviço, consta toda a lógica de um componente do negócio. Possui métodos específicos que aguardam acionamento por parte do Controller. Ainda, é o responsável pela comunicação com a camada de persistência de dados, a Persistence. Após processar a informação, retorna para o Controller o resultado do processamento, que por sua vez irá retornar ao View.
- ❖ Persistence: comunica-se com a base de dados e executa consultas.
- ❖ Banco de dados
- ❖ OCR: realiza a leitura de texto (Optical Character Recognition) e de marcas (Optical Mark Recognition) das avaliações e retorna os dados obtidos.

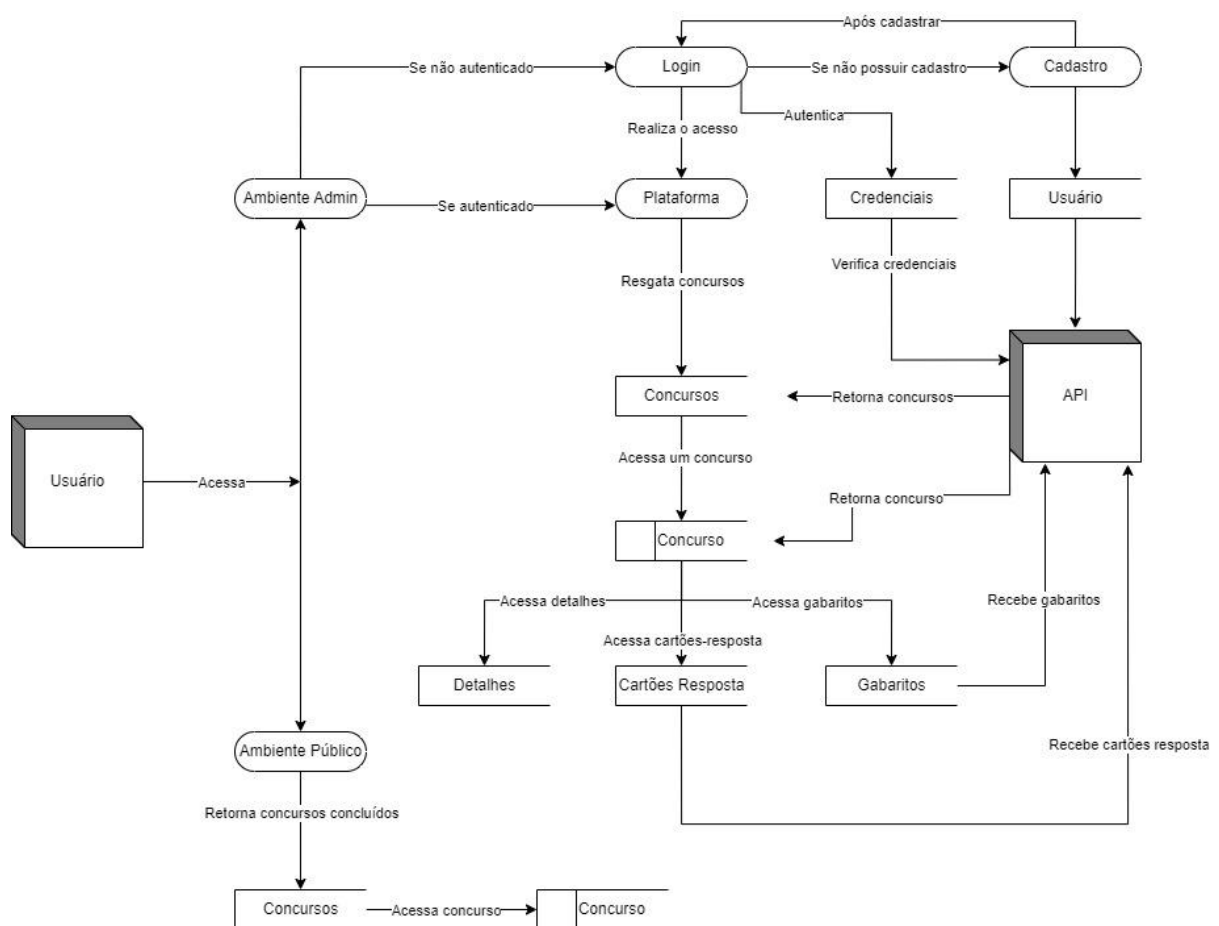
3.1.2. Diagrama



3.2. Data Flow Diagram (DFD)

Os diagramas abaixo representam o fluxo de dados a nível lógico. O primeiro fornece o esquema de User-Interface, o outro de Interface-Database.

3.2.1. Front-End



3.2.2. Back-End

