



Secure Blog Hosting

{up201905492} Cláudia Maia

{up202209374} Junior Monteiro

{up201904609} Tomás Vicente



Introdução

Design

Índice

Introdução

Neste projeto integrámos um servidor em Spring, o cliente em Flutter e ainda um servidor para o login, Keycloak.

Como comparação, configurámos uma instância de Wordpress para estudar a implementação e analisar.

Design

A escolha do Spring permitiu a criação de uma API RESTful para atender às necessidades de comunicação entre o frontend e o backend.

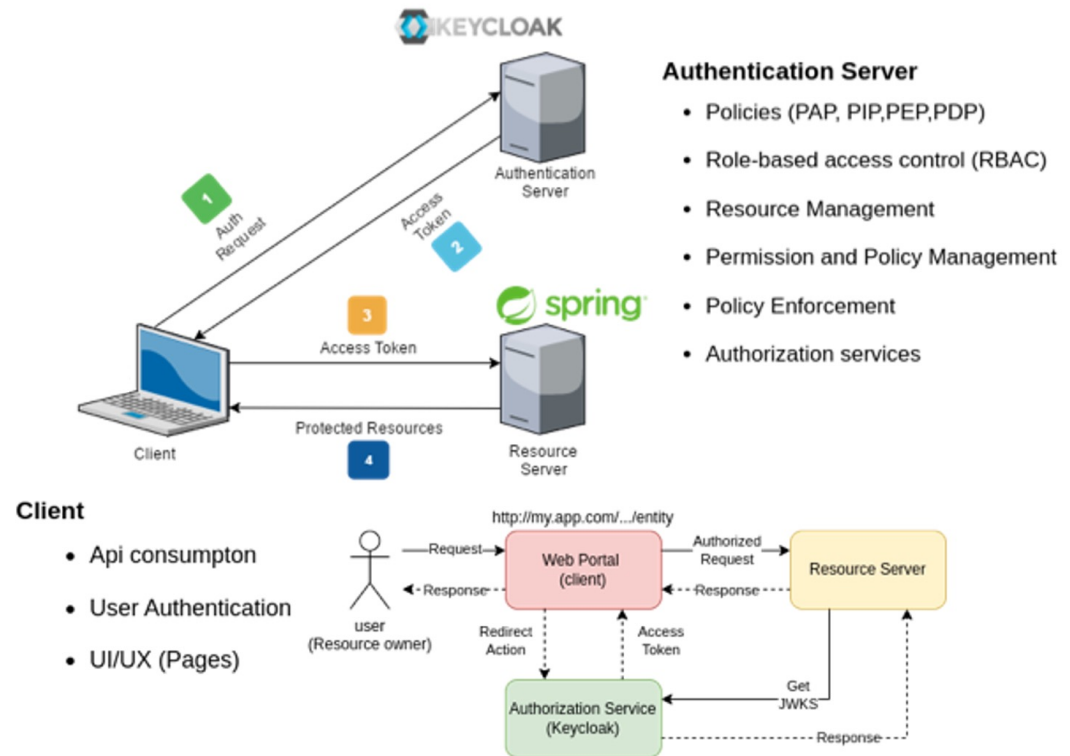
Para garantir a autenticação segura dos utilizadores, foi adotado o Keycloak como serviço de gestão de identidade e acesso.

Flutter utiliza a linguagem de programação Dart e oferece uma abordagem baseada em widgets para a criação de interfaces de usuário.

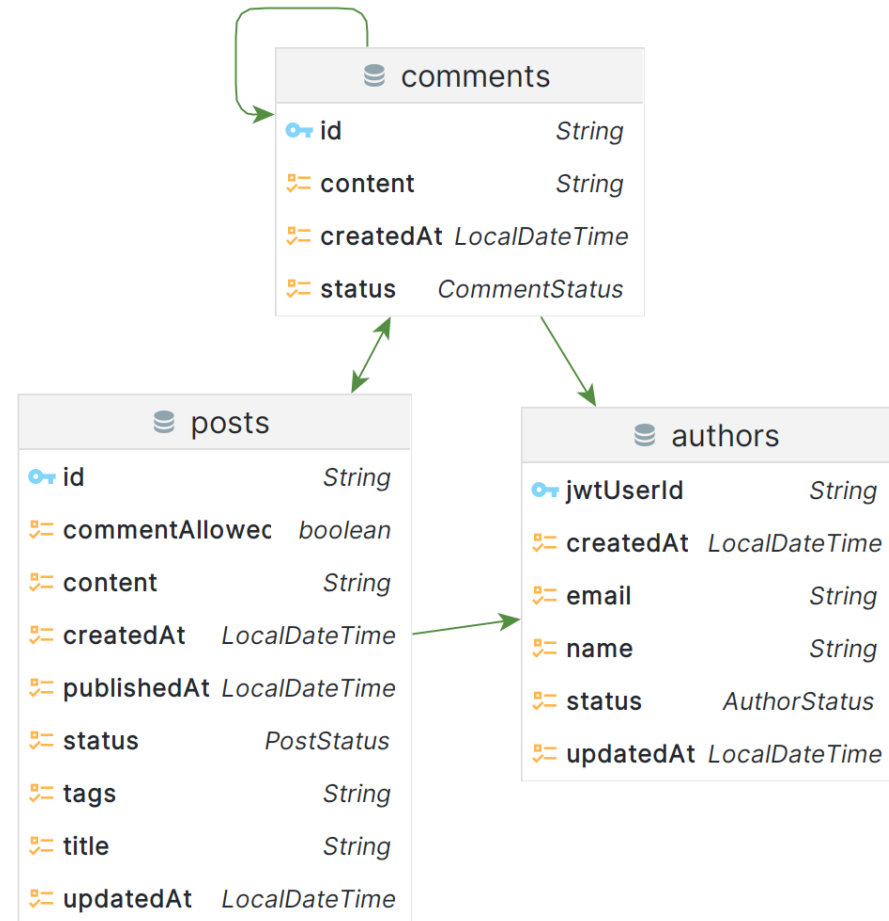
O design do blog abrangeu os seguintes aspectos: gestão de conteúdo, comentários, autenticação e access control, segurança da comunicação

Design

Secure Blog Hosting



Design



Authentication Server – Keycloak

- Autenticação Forte e controlada
- Gestão dos utilizadores
- Single-Sign On
- Identity Brokering and Social Login
- User Federation
- Standard Protocols

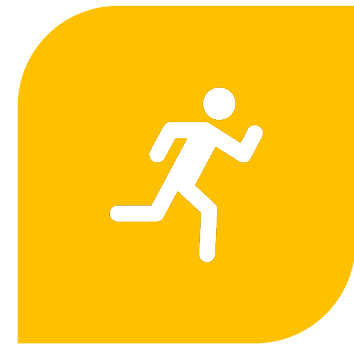
Resource Server - Spring Boot REST



SEGURO




PRODUTIVO



RÁPIDO

A large orange shape on the left side of the slide, consisting of a rectangle with a quarter-circle cutout on its right side.

Análise (Server)

- Garantias de segurança oferecidas pelas linguagens de programação
 - Padrões de design de software seguro
 - Análise de segurança das dependências e bibliotecas externas
 - Mitigações adotadas para vulnerabilidades comuns
 - Metodologias de testes de segurança (manuais ou automatizados) adotadas
 - Ferramentas de análise de código fonte integradas no desenvolvimento
- 
- A yellow dashed line in the bottom right corner, consisting of several short, curved segments.

Análise (Wordpress)

- Garantias de segurança oferecidas pela plataforma
- Aumento de segurança através de plugins (2FA, políticas password, captcha, limites login)
- Resultado: plataforma segura com mais funcionalidades



Conclusão

- Ao considerarmos a segurança durante o design e a implementação do código, podemos reduzir a ocorrência de problemas futuros e evitar a necessidade de corrigir erros de segurança posteriormente.
- Além disso, compreendemos a importância de realizar análises estáticas de código e testes de penetração para identificar possíveis vulnerabilidades que possam ter escapado durante a fase de design.
- Essas práticas permitem-nos fortalecer a segurança do nosso sistema e garantir a proteção dos dados e a confiança dos usuários.
- Plataformas como Wordpress facilitam o processo, desde que bem utilizadas (plugins, temas, etc.)