

Arquitetura Mobile React Native

Última edição 21 dias atrás por WikiSysop

Índice

- 1 INTRODUÇÃO
- 2 REPRESENTAÇÃO ARQUITETURAL
- 3 METAS E RESTRIÇÕES DA ARQUITETURA
- 4 VISÃO DE CASOS DE USO
 - 4.1 Realizações de Casos de Uso
- 5 VISÃO LÓGICA
 - 5.1 Visão Geral
 - 5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura
 - 5.3 Camadas
 - 5.3.1 Component
 - 5.3.2 Service
 - 5.3.3 Controller
 - 5.3.4 Service (Backend)
 - 5.3.5 Repository
 - 5.3.6 Model
- 6 VISÃO DE IMPLANTAÇÃO
 - 6.1 Visão Geral
 - 6.2 Exemplo de Configuração do Keycloak
- 7 REFERÊNCIAS
- 8 GUIA DE IMPLEMENTAÇÃO

INTRODUÇÃO

Este documento oferece uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

REPRESENTAÇÃO ARQUITETURAL

Seguindo o modelo 4+1 de representação Arquitetural definido por Philippe Kruchten, este documento cobrirá as seguintes visões Arquiteturais:

- Visão Lógica: Abrange as classes, interfaces e colaborações que formam o vocabulário do problema e de sua solução;
- Visão de Implementação: Abrange os componentes e os artefatos utilizados para a montagem e fornecimento do sistema físico;
- Visão do Processo: Mostra o fluxo de controle entre as várias partes, incluindo mecanismos de concorrência e sincronização;
- Visão de Implantação: Abrange os nós que formam a topologia de hardware em que o sistema é executado.

METAS E RESTRIÇÕES DA ARQUITETURA

Visando atender as necessidades do MCTIC, algumas restrições tiveram que ser observadas na instanciação da solução arquitetural. A “Tabela 1 - Restrições do cliente” descreve uma relação exemplificativa de restrições que foram levadas em consideração para a solução arquitetural.

Restrições significativas para a solução arquitetural	
1	Os sistemas devem ser responsivos para serem acessados por vários tipos de dispositivos via WEB.
2	Os sistemas podem ter um aumento significativo de acesso de usuários.
3	Os sistemas devem responder as requisições em menos de 3 segundos em 99% dos casos.
4	Os processos de negócio dos sistemas devem possuir facilidade para evoluções contínuas.
5	A autenticação deverá ser centralizada em apenas um local com múltiplos provedores.
6	Os dados corporativos devem ser coesos e de fácil evolução e correção.
7	Os sistemas devem possuir a flexibilidade de se comunicar com banco de dados.

VISÃO DE CASOS DE USO

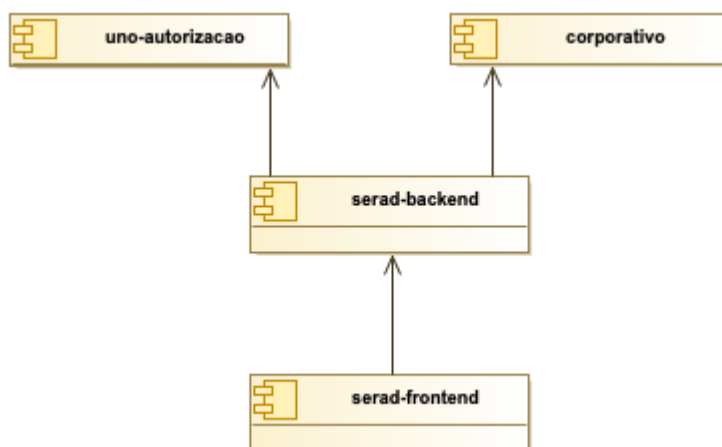
Para representação da arquitetura foi escolhido como exemplo o caso de uso “Consultar Canal Radiofusão”.

Realizações de Casos de Uso

Abaixo segue uma ilustração de como é feita a execução de uma operação do caso de uso “Visualizar Canais”, seguindo a arquitetura definida para o projeto.



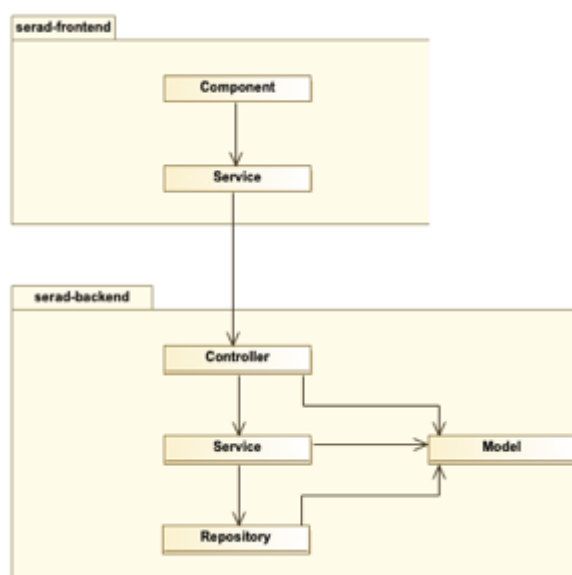
VISÃO LÓGICA



Visão Geral

O sistema utiliza o framework React Native, que possibilita ao desenvolvedor construir as funcionalidades para os dispositivos móveis utilizando a biblioteca React. A integração entre o aplicativo e os dados é feita utilizando uma arquitetura REST, que permite a troca de mensagens através dos métodos do protocolo HTTP. A verificação de identidade e acesso é feita através da ferramenta keycloak que implementa os protocolos open OpenID Connect e Oauth2.

Pacotes de Design Significativos do Ponto de Vista da Arquitetura



Camadas

Component

Esta camada é responsável pela disponibilização de uma funcionalidade para um dispositivo móvel. Através dela, o usuário tem a possibilidade de interagir com o aplicativo e enviar informações para o servidor backend.

Service

Esta camada é responsável por enviar informações para o backend da aplicação, e também de tratar e processar regras de negócio associadas a um domínio de dados específico.

Controller

Esta camada do backend é responsável por receber os dados da requisição enviado pelo aplicativo, e repassar as informações para a camada responsável pelo tratamento e processamento das regras de negócio.

Service (Backend)

Esta camada é responsável por receber dados de domínio e realizar processamentos associados as regras de negócio de uma aplicação. Nela é realizado todo o controle transacional necessário para que a aplicação respeite o conceito de atomicidade das operações. Quando necessário, esta camada enviará informações para repositórios externos à aplicação através da camada Repository.

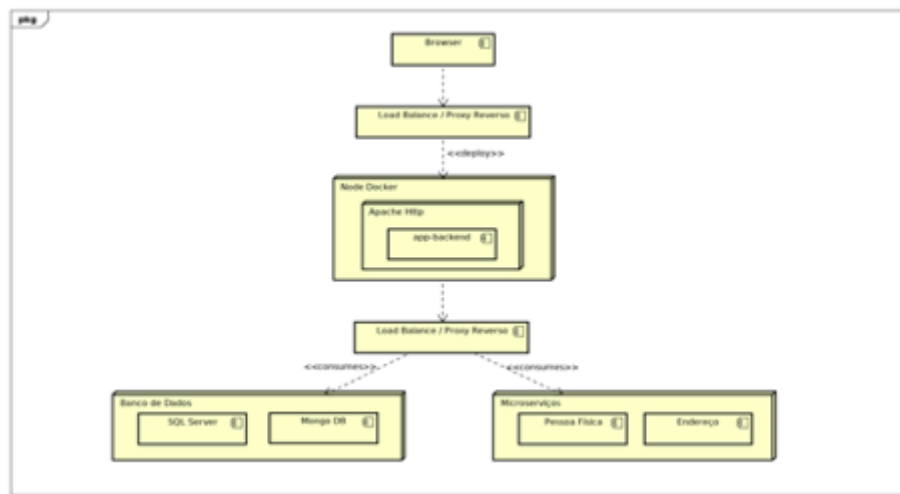
Repository

Esta camada é responsável pela comunicação com toda e qualquer infraestrutura externa à aplicação. Como por exemplo, um banco de dados e um microserviço corporativo.

Model

Esta camada é responsável por mapear toda e qualquer estrutura que represente um objeto de dados da aplicação. Nenhuma regra de negócio deve estar associada a este grupo de classes, apenas mapeamentos de entidades e definições de restrições de informação.

VISÃO DE IMPLANTAÇÃO



Visão Geral

Para a verificação de identidade está sendo utilizado a lib react-native-auth que implementa uma ponte para a lib AppAuth que implementa os protocolos OpenID Connect e OAuth2 e segue as diretrizes propostas na RFC8252, onde essa RFC propõe a estrutura necessária para a implementação dos protocolos OAuth2 e OpenIDConnect no ambiente de aplicações mobile nativas.

Exemplo de Configuração do Keycloak

Para o correto funcionamento da verificação de identidade é necessário que exista uma configuração no projeto para guardar os seguintes atributos requeridos pela lib react-native-auth:

```
const config = {
  warmAndPrefetchChrome: true,
  issuer: 'https://sso.dsv.app.mctic.gov.br/auth/realms/mctic',
  clientId: 'mobile',
  redirectUrl: 'br.gov.mctic:/callback',
  scopes: ['openid'],
};
```

Para cada sistema operacional mobile existe uma configuração necessária para o correto funcionamento:

Para aplicações IOS:

1. Entrar na pasta iOS do projeto e executar o comando pod install.
2. Registrar as URL's de redirect no Info.plist:

```

<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleURLName</key>
    <string>$(PRODUCT_BUNDLE_IDENTIFIER)</string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>br.gov.mctic</string>
    </array>
  </dict>
</array>

```

3. Para reter a sessão de autenticação é necessário retirar e adicionar os seguintes trechos de código no AppDelegate.h e AppDelegate.m.

AppDelegate.h:

```

+ #import "RNAppAuthAuthorizationFlowManager.h"
- @interface AppDelegate : UIResponder <UIApplicationDelegate, RCTBridgeDele-
+ @interface AppDelegate : UIResponder <UIApplicationDelegate, RCTBridgeDele-
+ @property(nonatomic, weak)id<RNAppAuthAuthorizationFlowManagerDelegate>autho

```

AppDelegate.m:

```

+ (BOOL)application:(UIApplication *)app openURL:(NSURL *)url op-tions:(NSDict
+ return [self.authorizationFlowManagerDelegate resumeExternalUserAgentFlow-V
+ }

```

Para aplicações Android:

1. Registrar a url de redirect no android/app/build.gradle

```
android {  
  defaultConfig {  
    manifestPlaceholders = [  
      appAuthRedirectScheme: 'br.gov.mctic'  
    ]  
  }  
}
```

REFERÊNCIAS

RFC8252: [rfc:8252 <https://tools.ietf.org/html/rfc8252>]

React-native-auth: <https://github.com/FormidableLabs/react-native-app-auth>

GUIA DE IMPLEMENTAÇÃO

As informações sobre implementação estão disponibilizadas na página [Guia de Implementação REACT NATIVE](#).