



# Autenticando microsserviços com Keycloak

---

Vinicius Sanchez



# Agenda

- Principais conceitos sobre Autenticação e Autorização
- OAuth2
- OpenID Connect
- Keycloak
- Dúvidas

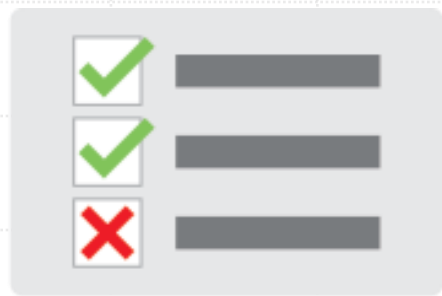
**AC**  
**Br**



# Apresentação pessoal

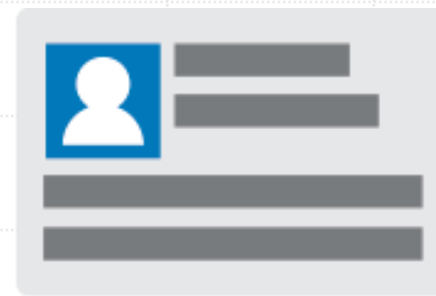
- Graduado em Sistemas de Informação
- Embarcadero MVP
- Certificação Delphi Developer
- Desenvolvedor de software na Fiorilli
- Membro da Hashload
- Criador de conteúdos

# Principais conceitos sobre Autenticação e Autorização



## Autorization

O que você pode fazer?



## Authentication

Quem é você?

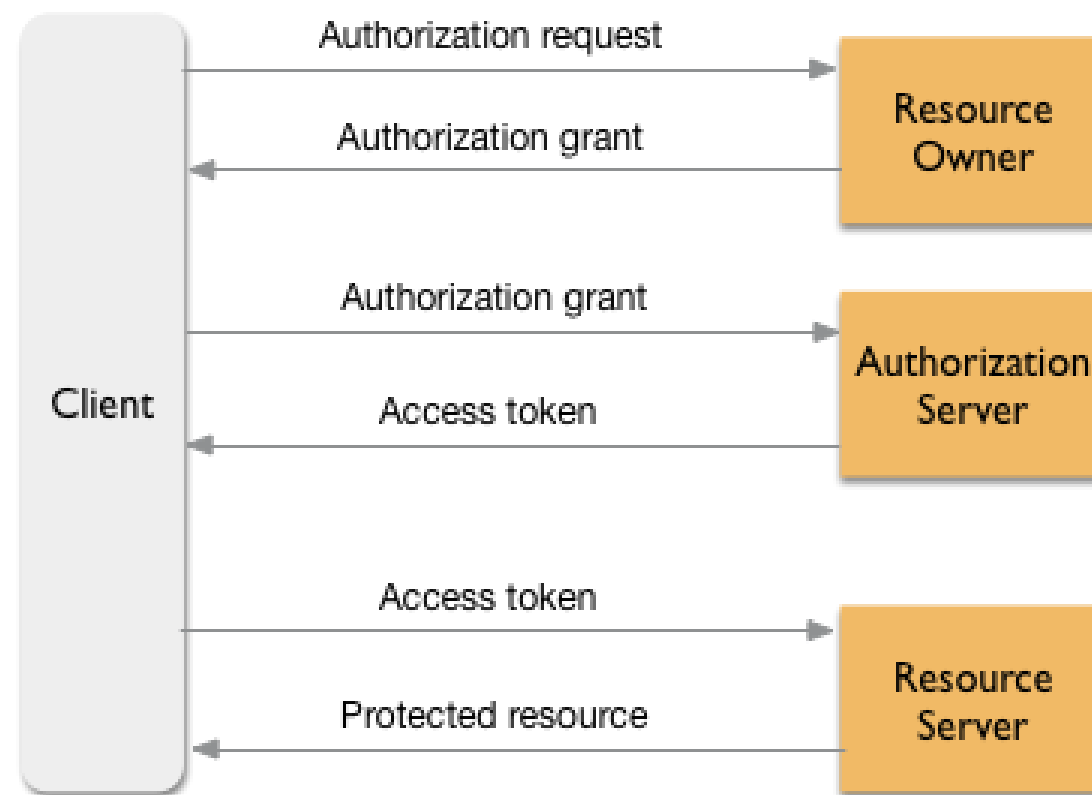


# OAuth2

- Um framework (especificações) de autorização que permite que terceiros possam obter acesso **limitado** a um serviço HTTP.
- Framework de autorização, **não é um framework de autenticação**
- Ninguém se autentica usando apenas o OAuth2

# OAuth2 “Flow”

- Abrimos um aplicativo, e ele solicita que você dê permissão para ele acessar dados como fotos, contatos, entre outros recursos do aparelho...



# Exemplificando...



- Joãozinho tem uma conta bancária
- Mariazinha quer transferir dinheiro
- Gerente nega, porque não tem permissão
- Joãozinho faz uma procuração
- Mariazinha tenta transferir novamente
- Gerente valida a procuração
- Libera a transferência

# Personagens

Joãozinho deu uma permissão **temporária** e **limitada** para Mariazinha.

- Resource Owner → Dono do recurso (Joãozinho)
- Client → Pessoa que quer acessar (Mariazinha)
- Resource Server → Onde está o nosso recurso (Banco)
- Authorization Server → Valida autorização (Cartório)

Fluxo (OAuth2)



# Como aproveitar o fluxo do OAuth2 e trabalhar com Autenticação (Login)?

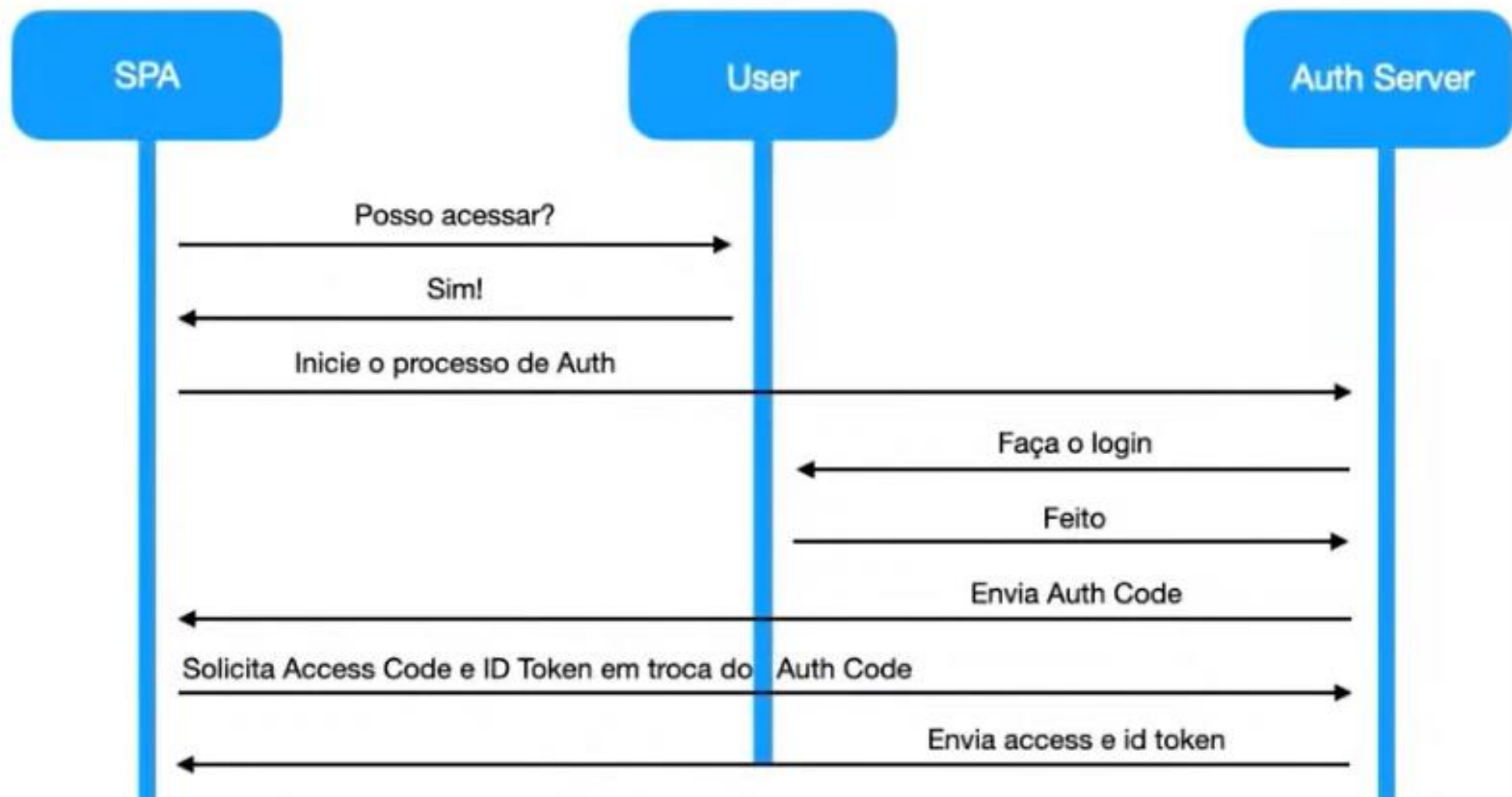
- Durante muito tempo se perguntava em como aproveitar todo o fluxograma do OAuth2 que serve para autorização, para atender também a demanda de autenticação;



# OpenID Connect

- Uma camada (protocolo) de identidade que funciona em cima do fluxo do OAuth2
- Permite usar o OAuth2 para realizar o processo de autenticação e autorização
- Exemplo: Todas as vezes que você realiza login usando o Facebook, provavelmente você está usando além do fluxograma do OAuth2, o OpenID Connect para realizar a parte de autenticação







# Id token e Access token?

- Id\_token possui informações necessárias para garantir a autenticação do usuário, como ID, E-mail e informações adicionais passadas pelo servidor de autenticação;
- O **id\_token** é focado no usuário (**autenticação**), o **access\_token** é focado em o que aquele usuário pode fazer (**autorização**), ou seja, suas permissões;
- JWT (JSON Web Token), que é um padrão aberto que representa de forma segura solicitações de informações entre duas partes;
- Token é um código Base64 que armazena um JSON
- Possui 3 partes: **Header**, **Payload** e **Signature**

# Open Source Identity and Access Management

Add authentication to applications and secure services with minimum effort.

No need to deal with storing users or authenticating users.

Keycloak provides user federation, strong authentication, user management, fine-grained authorization, and more.

[Get Started](#)[Download](#)

Latest release 26.0.5



## News

01 Nov [Keycloak 26.0.5 released](#)

30 Oct [Keycloak 26.0.4 released](#)

25 Oct [Keycloak DevDay 2025 Pre-Conf Event Announcement](#)

## Single-Sign On



# Instalação do Keycloak

- Em um terminal, digite o seguinte comando para iniciar o Keycloak:

```
docker run -p 8080:8080 -e KC_BOOTSTRAP_ADMIN_USERNAME=admin -e KC_BOOTSTRAP_ADMIN_PASSWORD=admin  
quay.io/keycloak/keycloak:26.0.5 start-dev
```

- Este comando inicia o Keycloak exposto na porta local **8080** e cria um usuário administrador inicialmente com o nome de **admin** e a senha **admin**.





## Sign in to your account

Username or email

admin



Password

.....



Sign In



# Criando um novo client

- Aplicações que queremos proteger com o Keycloak;
- Lembre-se de não remover as aplicações que já estão cadastradas por padrão;
- O que precisamos definir:
  - **Client ID:** nome da aplicação que vamos proteger
  - **Valid Redirect URI:** <https://www.getpostman.com/oauth2/callback>



# Obtendo um novo access token

- **Auth Type:** OAuth 2.0
- **Grant Type:** Authorization Code
- **Callback URL:** <https://www.getpostman.com/oauth2/callback>  
(URL que será redirecionado após a aplicação ser autorizada)
- **Auth URL:** <http://localhost:8080/realms/master/protocol/openid-connect/auth>  
(Utilizado para obter o authorization code)
- **Access Token URL:** <http://localhost:8080/realms/master/protocol/openid-connect/token>  
(Utilizado para obter o access token com base em um authorization code)
- **Client ID:** acbr-app
- **Scope:** email openid profile  
(Escopos são recursos, papéis e até mesmo ações que a aplicação cliente solicita ao servidor)

Encoded

PASTE A TOKEN HERE

eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJmaHlTdDNaZDVqYlVDWWRRReUswYkVBZnhtejVxRlJFYzdwRVFRanZiOTY4In0.eyJleHAiOiE3MzA2NTY3NjgsIm1hdCI6MTczMDY1NjcwOCwiYXV0aF90aW1lIjoxNzMwNjU2NzA4LCJqdGkiOiI0ZjNhYWU5OC1jMDIxLTQ4OTAtYWlyOS01NzZiYjA1ZTU5ZDMiLCJpc3MiOiJodHRwOi8vbG9jYXxob3N0jgwODAvcmVhbG1zL21hc3RlciIsImF1ZCI6WyJtYXN0ZXItcmVhbG0iLCJhY2NvdW50IHRsInN1YiI6IjJjYjczMjI2LTM2YmEtNDVlYS1iODk0LWUxMzY3YzMzZTBiYiIsInR5cCI6IkJ1YXJlc3R1bGciLCJmFjYnItYXBwIiwic2lkIjoiiYjgxiOGIyZmQtODYyZS00NWUwLTk5ZTYtZTk3OQ==

## Decoded

### EDIT THE PAYLOAD AND SECRET

### HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT",
  "kid": "fhySt3Zd5jbUCYdQyK0bEAfxmz5qFREc7pEQqjvb968"
}
```

PAYLOAD: DATA

```
"iat": 1730656708,
"auth_time": 1730656708,
"jti": "4f3aae98-c021-4890-ab29-576bb05e59d3",
"iss": "http://localhost:8080/realms/master",
"aud": [
  "m
```

Required user actions

Select action

Configure OTP

Update Password

Update Profile

Verify Email

Webauthn Register

Webauthn Register Passwordless

Verify Profile

Delete Credential

Update User Locale

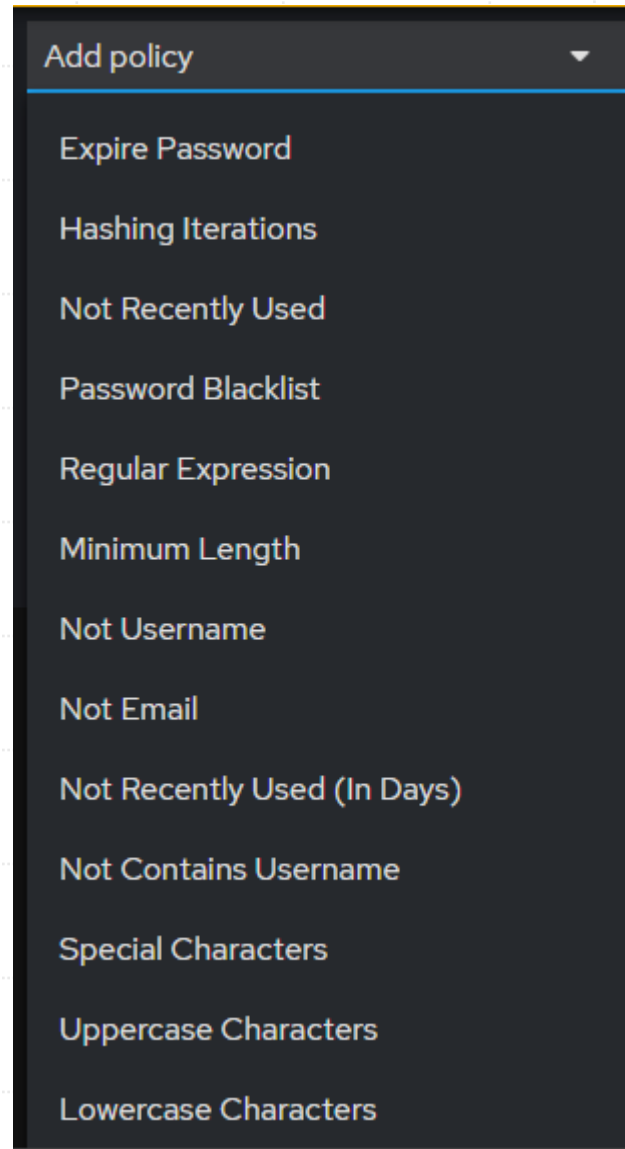
Cancel

# Criando um novo usuário

*Users > Add user*

# Password policy

- Gerenciamento de senha de usuários, para a aplicação de uma política mais forte, como quantidade mínima de caracteres por exemplo...
- *Authentication > Policies > Password policy*



# Identity providers



## Login

☐ Remember Me

[Log In](#)

[Register](#)[Lost your password?](#)

Sign in with 

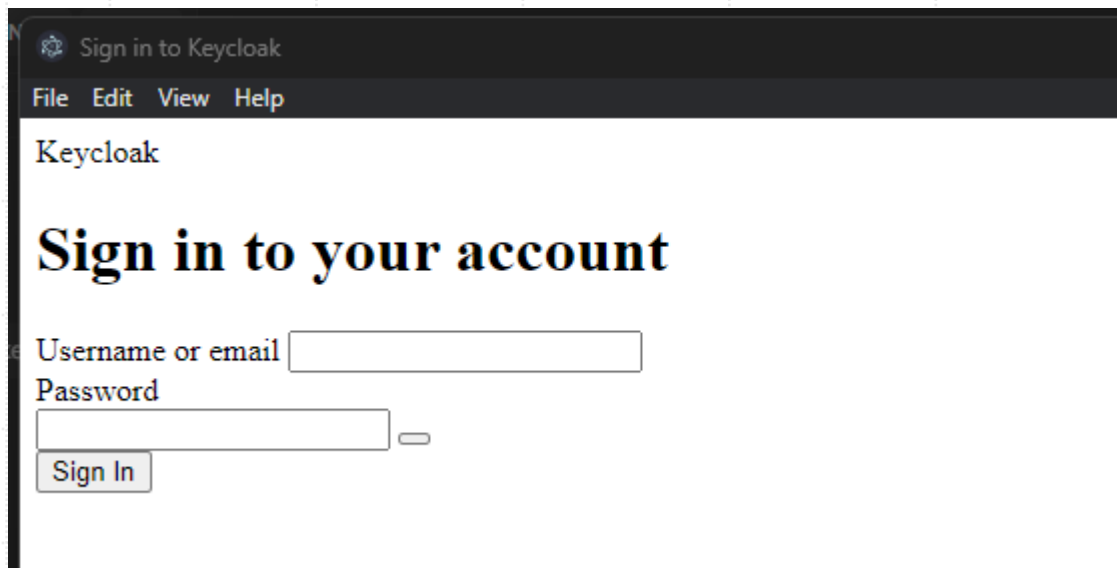
f

G

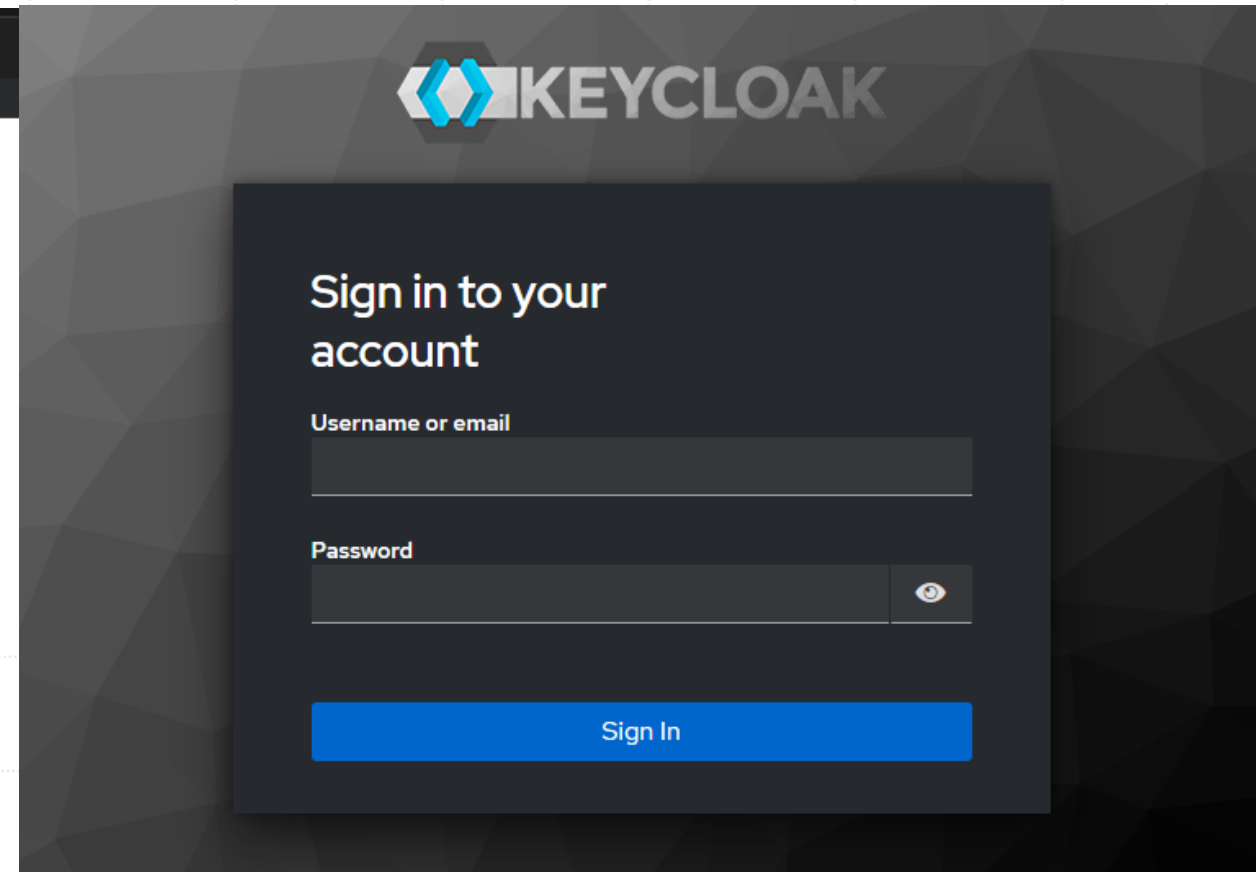
X

in

# Customizando a tela de login

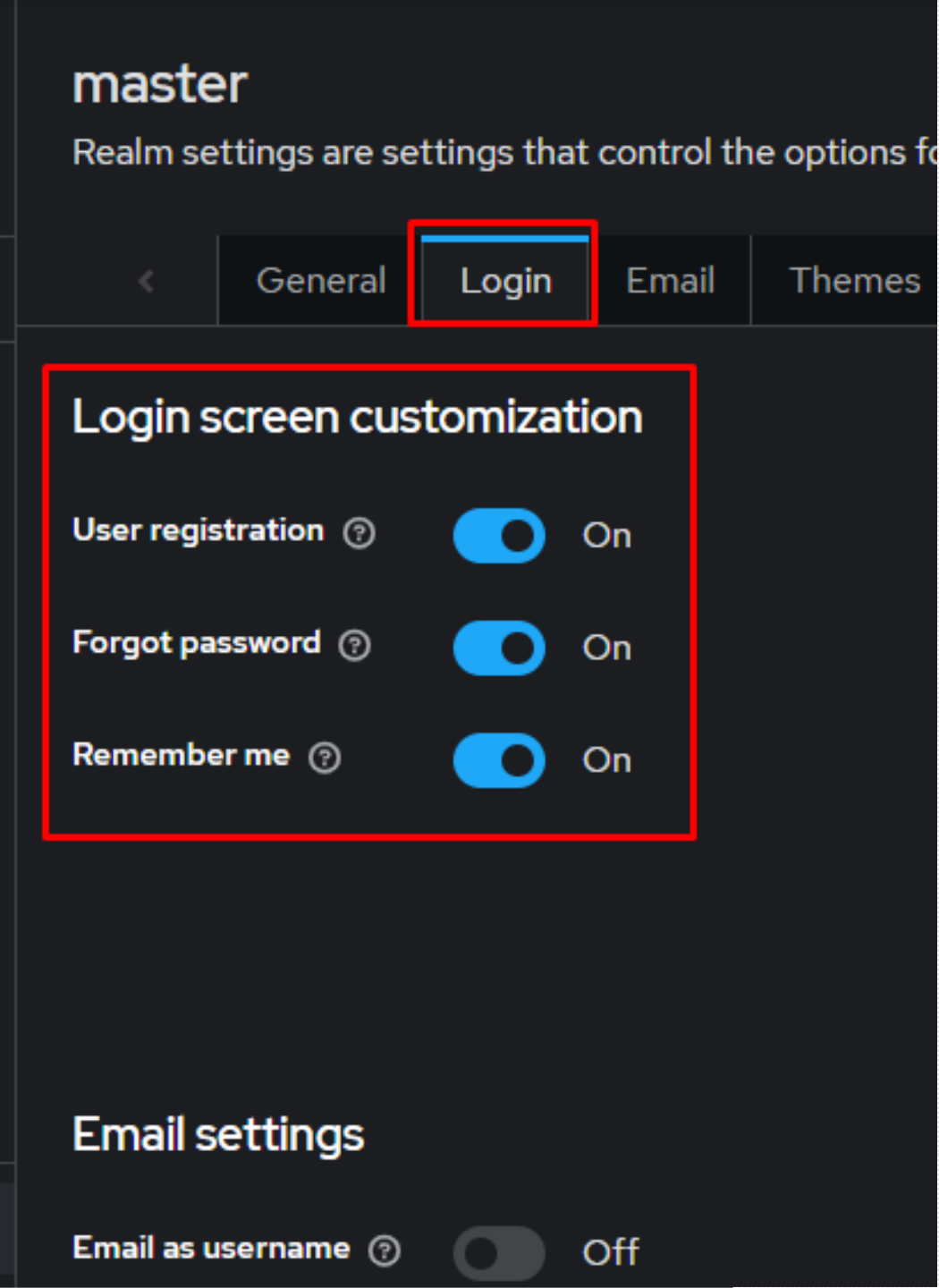


A screenshot of a web browser window showing the default Keycloak login page. The browser's address bar displays "Sign in to Keycloak". The page has a dark header with "File Edit View Help" menus. Below the header, the text "Keycloak" is visible. The main heading is "Sign in to your account". There are two input fields: "Username or email" and "Password". A "Sign In" button is located below the password field.



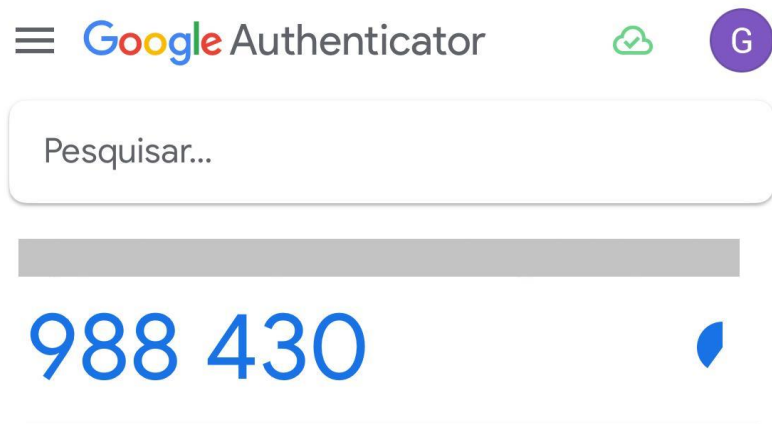
A screenshot of a customized Keycloak login page with a dark theme. The background features a dark gray geometric pattern. The Keycloak logo is at the top. The login form is a dark gray box with the heading "Sign in to your account". It contains two input fields: "Username or email" and "Password". The password field has a toggle icon (an eye) to its right. A large blue "Sign In" button is at the bottom of the form.

# Customizando a tela de login



# Two-factor authentication

- Para utilizar o segundo fator de autenticação, é necessário acessar Authentication > Policies > OTP Policy e ativar o recurso;
- Em seguida, precisamos acessar as preferências do usuário no link abaixo, e configurar o segundo fator de autenticação
- <http://localhost:8080/realms/master/account>







***Dia do ACBr***  
2 0 2 4



# Dúvidas?

[viniciuss.sanchez@gmail.com](mailto:viniciuss.sanchez@gmail.com)



@viiniciussanchez