

Métodos de Desenvolvimento de Software

Integração Email

Professora: Carla Rocha

Alunos:

Nathan Abreu 221022696

Maria Helena Carvalho 222006982

Marlon 222025914

Thales Henrique - 222006178

Victor Hugo 222021924

Otavio 211043692

Marcelo Adrian Ribeiro de Araújo 202016909

12 de junho de 2024

Brasília - DF

Objetivo

O objetivo deste projeto é criar uma ferramenta onde os usuários possam fazer login usando suas contas do Google, sem a necessidade de armazenar suas senhas e emails diretamente. Após o login, os emails dos usuários serão armazenados no Firestore, e mensalmente será enviado um newsletter com atualizações sobre as licitações de Brasília que ocorreram no mês anterior.

Uso e Funcionamento

O sistema será composto por três componentes principais:

1. **Autenticação de Usuário com Google:** Permitir que os usuários façam login no site usando suas contas do Google.
2. **Armazenamento de Emails:** Armazenar os emails dos usuários no Firestore.
3. **Envio de Newsletters:** Usar Firebase Functions para enviar um email mensalmente a todos os usuários com as atualizações das licitações.

Implementação

1. Configuração do Firebase

Criação do Projeto no Firebase:

1. Acesse o Firebase Console.
2. Crie um novo projeto.
3. Adicione um aplicativo web ao projeto e copie as configurações do Firebase SDK.

Configuração da Autenticação:

1. No Firebase Console, vá para a seção "Authentication".
2. Habilite o provedor de login do Google.

2. Configuração do Frontend (React)

Instalação do Firebase:

```
npm install firebase
```

Configuração do Firebase no Aplicativo React:

```
// src/firebase.js
import firebase from 'firebase/app';
import 'firebase/auth';
import 'firebase/firestore';

const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "YOUR_AUTH_DOMAIN",
  projectId: "YOUR_PROJECT_ID",
  storageBucket: "YOUR_STORAGE_BUCKET",
  messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
  appId: "YOUR_APP_ID"
};

firebase.initializeApp(firebaseConfig);

export const auth = firebase.auth();
export const googleProvider = new firebase.auth.GoogleAuthProvider();
export const firestore = firebase.firestore();
```

Implementação do Login com Google e Armazenamento do Email:

```
// src/Login.js
import React from 'react';
import { auth, googleProvider, firestore } from '../firebase';

function Login() {
  const handleLogin = async () => {
    try {
      const result = await auth.signInWithPopup(googleProvider);
      const user = result.user;
      const email = user.email;

      // Armazene o email no Firestore
      const usersRef = firestore.collection('users');
      const userDoc = await usersRef.doc(user.uid).get();
      if (!userDoc.exists) {
        await usersRef.doc(user.uid).set({ email });
      }

      console.log('Login bem-sucedido! Usuário:', user);
    } catch (error) {
      console.error('Erro no login:', error);
    }
  };

  return (
    <div>
      <button onClick={handleLogin}>Login com Google</button>
    </div>
  );
}

export default Login;
```

3. Envio de Emails com Firebase Functions

Configuração do Firebase Functions:

Instale a CLI do Firebase:

```
npm install -g firebase-tools
```

Inicie o Firebase Functions no projeto:

```
firebase init functions
```

Instale os pacotes necessários:

```
cd functions  
npm install nodemailer googleapis
```

Configuração da Função para Enviar Emails:

```
// functions/index.js
const functions = require('firebase-functions');
const admin = require('firebase-admin');
const nodemailer = require('nodemailer');
const { google } = require('googleapis');

admin.initializeApp();

const CLIENT_ID = 'YOUR_GOOGLE_CLIENT_ID';
const CLIENT_SECRET = 'YOUR_GOOGLE_CLIENT_SECRET';
const REDIRECT_URI = 'https://developers.google.com/oauthplayground';
const REFRESH_TOKEN = 'YOUR_REFRESH_TOKEN';

const oAuth2Client = new google.auth.OAuth2(
  CLIENT_ID,
  CLIENT_SECRET,
  REDIRECT_URI
);
oAuth2Client.setCredentials({ refresh_token: REFRESH_TOKEN });

async function sendMail(email, subject, text) {
  try {
    const accessToken = await oAuth2Client.getAccessToken();

    const transport = nodemailer.createTransport({
      service: 'gmail',
      auth: {
        type: 'OAuth2',
        user: 'your-email@gmail.com',
        clientId: CLIENT_ID,
        clientSecret: CLIENT_SECRET,
        refreshToken: REFRESH_TOKEN,
        accessToken: accessToken.token,
      },
    });

    // ... (rest of the code) ...
  }
}
```

```
const mailOptions = {
  from: 'SENDER NAME <your-email@gmail.com>',
  to: email,
  subject: subject,
  text: text,
  html: `<h1>${text}</h1>`,
};

const result = await transport.sendMail(mailOptions);
return result;
} catch (error) {
  console.error('Error sending email:', error);
  throw new functions.https.HttpsError('internal', 'Error sending email', error);
}
}

exports.sendNewsletter = functions.pubsub.schedule('0 0 1 * *').onRun(async (context) => {
  try {
    const usersSnapshot = await admin.firestore().collection('users').get();
    const promises = [];
    usersSnapshot.forEach((doc) => {
      const email = doc.data().email;
      const subject = 'Monthly Newsletter';
      const text = 'Here are the updates for this month...';
      promises.push(sendMail(email, subject, text));
    });
    await Promise.all(promises);
    console.log('Newsletter sent to all users');
  } catch (error) {
    console.error('Error sending newsletters:', error);
  }
});
```

Deploy do Firebase Functions:

```
firebase deploy --only functions
```

Hospedagem do Frontend com Firebase Hosting

Configuração do Firebase Hosting:

```
firebase init hosting
```

Construção e Deploy do Aplicativo React:

Construa o aplicativo para produção:

```
npm run build
```

Implante o aplicativo:

```
firebase deploy --only hosting
```

Conclusão

Este projeto configura um sistema onde os usuários podem fazer login usando suas contas do Google, ter seus emails armazenados no Firestore, e receber newsletters mensais automaticamente através de uma função agendada do Firebase Functions. Isso proporciona uma solução eficiente e escalável para manter os usuários informados sobre as licitações de Brasília que ocorreram durante o mês. O uso do Firebase simplifica a infraestrutura, eliminando a necessidade de gerenciar um backend separado, enquanto mantém a segurança e a privacidade dos dados dos usuários.

