



Centro Universitário UNA

UC: Usabilidade

Professor: Diego Silva Caldeira Rocha

Introdução a Firebase

Firebase é uma plataforma desenvolvida pelo Google para ajudar os criadores de aplicativos móveis e web a construir e aprimorar suas criações de maneira eficaz. Ela oferece uma série de ferramentas e serviços que simplificam tarefas comuns no desenvolvimento de aplicativos. Por exemplo, permite que os desenvolvedores autentiquem usuários com facilidade, armazenem e sincronizem dados em tempo real, gerenciem arquivos de mídia na nuvem e enviem notificações push para os usuários.

Alguns dos principais recursos do Firebase incluem:

Autenticação de Usuário: Possibilita a autenticação de usuários de forma simples e segura, utilizando e-mail/senha, autenticação social (como Google, Facebook, etc.) e autenticação anônima.

Banco de Dados em Tempo Real: Um banco de dados NoSQL em tempo real que sincroniza dados entre os usuários em tempo real. É útil para aplicativos que necessitam de atualizações em tempo real, como chats e aplicativos colaborativos.

Firestore: Um banco de dados NoSQL de documentos que oferece escalabilidade, consultas flexíveis e sincronização em tempo real para aplicativos web e móveis.

Armazenamento em Nuvem: Fornece armazenamento para arquivos de mídia, como imagens, vídeos e outros tipos de arquivos, permitindo o armazenamento e o compartilhamento de conteúdo.

Funções na Nuvem: Permite que os desenvolvedores executem código do lado do servidor em resposta a eventos gerados por outros serviços do Firebase ou chamadas HTTP.

Notificações por Push: Permite enviar notificações por push para usuários de aplicativos Android, iOS e web.

Analytics: Fornece informações sobre o uso do aplicativo, incluindo dados demográficos, retenção de usuários e eventos personalizados.

Test Lab: Permite testar aplicativos em uma ampla variedade de dispositivos e configurações de hardware na infraestrutura do Google.

Esses são apenas alguns dos serviços oferecidos pelo Firebase. Ele é amplamente utilizado por desenvolvedores devido à sua facilidade de uso, escalabilidade e integração com outras ferramentas do Google.

Objetivo

Nesta prática utilizaremos o firebase para criar uma base de dados NOSQL através do banco de dados firestore e fazer uma página de CRUD* html/css/js.

C - Create (Criar): Refere-se à operação de criar novos registros ou entradas no banco de dados. Isso envolve inserir novos dados que não estavam anteriormente presentes no sistema.

R - Read (Ler): Envolve a operação de recuperar ou ler dados existentes do banco de dados. Isso permite que os usuários ou aplicativos obtenham informações armazenadas para visualização ou processamento posterior.

U - Update (Atualizar): Refere-se à operação de modificar ou atualizar registros existentes no banco de dados. Isso permite que os usuários ou aplicativos modifiquem dados já presentes no sistema.

D - Delete (Excluir): Envolve a operação de remover registros ou entradas existentes do banco de dados. Isso permite que os usuários ou aplicativos removam dados que não são mais necessários ou relevantes.

Siga as seguinte etapas:

- 1) Através de uma conta no Google entra no link e clique “add Project”
<https://console.firebase.google.com/>


2) Crie um projeto com nome que desejar

× Criar um projeto(Passo 1 de 2)

Vamos começar nomeando o projeto[?]

Nome do projeto

IntroFirebase

 introfirebase-d8a1f

☐ Aceito os [Termos do Firebase](#)

☐ Confirmando que vou usar o Firebase exclusivamente para fins relacionados ao meu comércio, empresa, escritório ou profissão.

× Criar um projeto(Passo 2 de 2)

O Google Analytics é uma solução de análise ilimitada e sem custos financeiros. Com ele, é possível segmentar, gerar relatórios e muito mais nos seguintes produtos: Firebase Crashlytics, Cloud Messaging, Mensagens no app, Configuração remota, Teste A/B e Cloud Functions.

O Google Analytics ativa:

× **Teste A/B**[?]

× **Segmentação de usuários em produtos do Firebase**[?]

× **Registros de navegação estrutural no Crashlytics**[?]

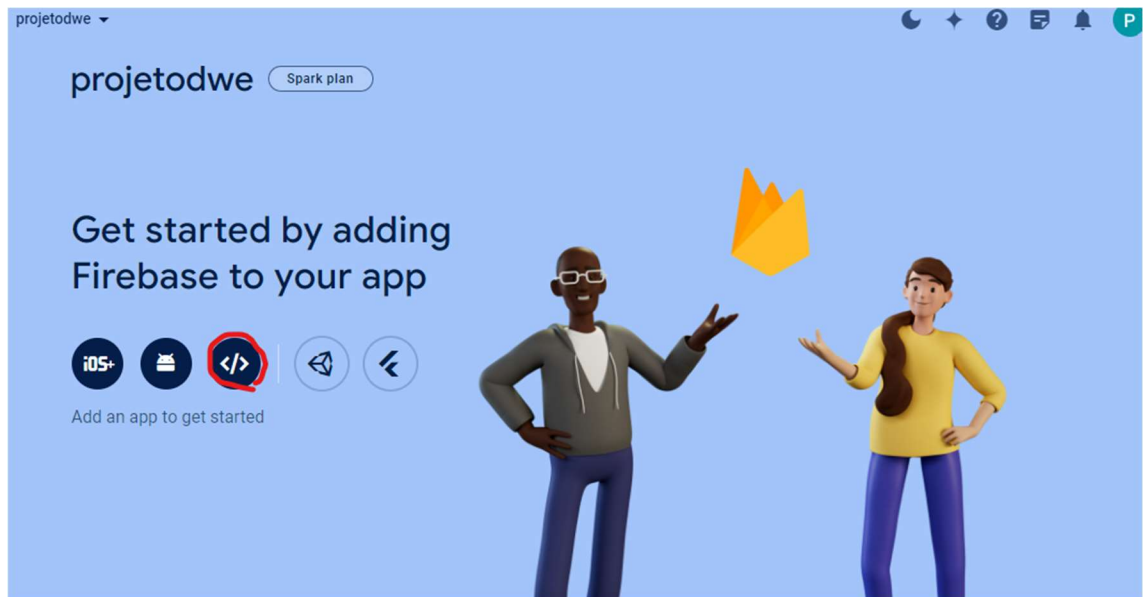
× **Gatilhos do Cloud Functions com base em eventos**[?]

× **Geração de relatórios ilimitada gratuita**[?]

☒ **Ativar o Google Analytics neste projeto**
Recomendado

[Anterior](#)[Criar projeto](#)

3) Clique no ícone da imagem a seguir.



- 4) Conforme a imagem a seguir copie os dados a seguir e deixe no bloco de notas e continue no console

×

Adicionar o Firebase ao seu app da Web

✓ Registrar app

2 Adicionar o SDK do Firebase

☐ Usar o npm

☒ Usar a tag <script>

Se você não utiliza ferramentas de builds, use essa opção para adicionar e utilizar o SDK do Firebase para JavaScript. Essa opção é recomendada no começo, mas não para apps em produção. [Saiba mais](#).

Copie e cole esses scripts na parte de baixo da tag <body>, mas antes de usar qualquer serviço do Firebase:

```
<script type="module">
// Import the functions you need from the SDKs you need
import { initializeApp } from "https://www.gstatic.com/firebasejs/10.11.0/firebase-app.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCoZW920DGBisuVd2Bra2tDZZCS5bEuh5g",
  authDomain: "introfirebase-d8a1f.firebaseio.com",
  projectId: "introfirebase-d8a1f",
  storageBucket: "introfirebase-d8a1f.appspot.com",
  messagingSenderId: "144288934361",
  appId: "1:144288934361:web:951e6cdc370e47198511c8"
};

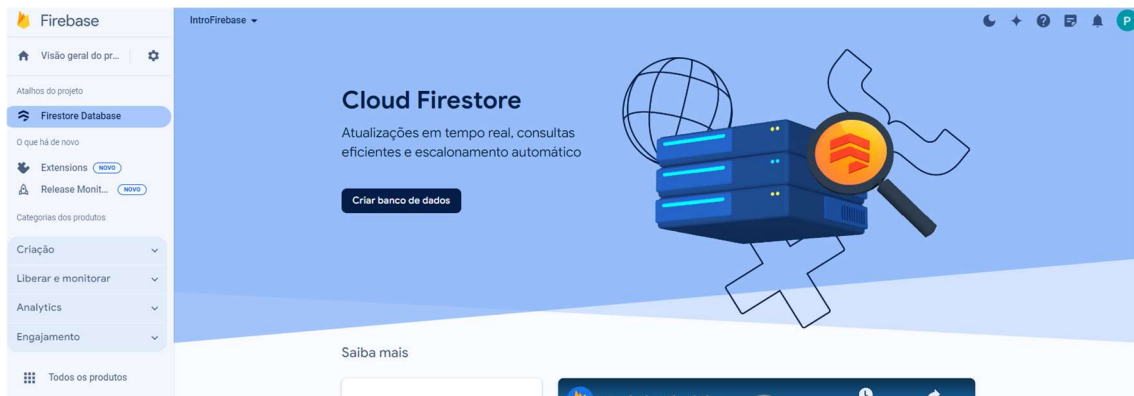
// Initialize Firebase
const app = initializeApp(firebaseConfig);
</script>
```

Você está usando npm e um bundler como Webpack ou Rollup? Consulte o [SDK modular](#).

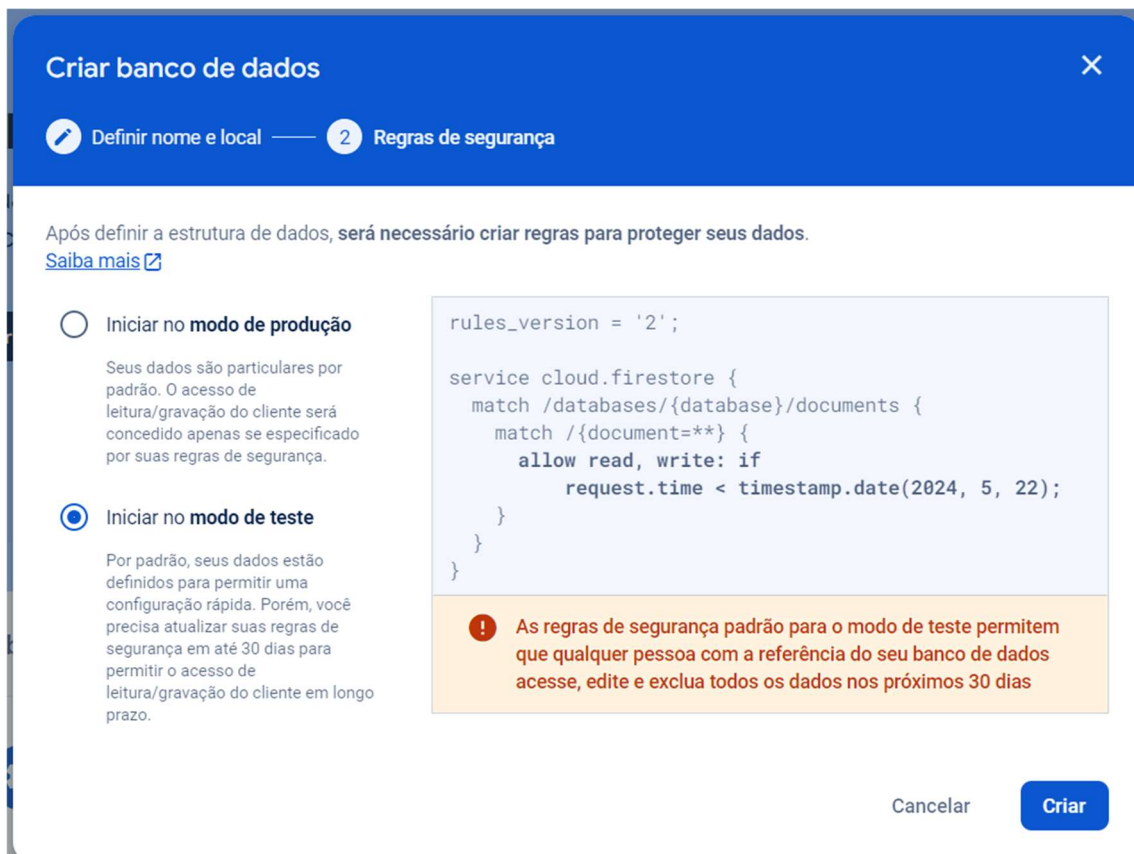
Saiba mais sobre o Firebase para Web: [Vamos começar](#), [Referência da API Web SDK](#), [Amostras](#)

Continuar no console

- 5) No menubar da esquerda procure "Firestore Database"
- 6) Clique em criar banco de dados



7) Coloque banco de dados na forma teste e clique em criar



8) Abra o vscode cire o seguinte padrão de navegação de pastas

```
IntroFirebase
| - index.html
| - script.js
| - style.css
```

9) Conteúdo do index.htm

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Cadastro de Usuários</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="notify"></div>
  <div class="form">
    <input type="text" id="name" placeholder="coloque o nome"
autocomplete="off">
    <input type="email" id="email" placeholder="digite o email"
autocomplete="off">
    <button id="add_Data" class="add_Data">Cadastrar</button>
    <button id="update_data" class="update_data">Atualizar</button>
  </div>

  <div class="table">
    <table>

    </table>
  </div>

  <script type="module" src="script.js"></script>
</body>
</html>

```

10) Conteúdo do script.js

```

// importando SDKs di fures base
import { initializeApp } from
"https://www.gstatic.com/firebasejs/10.14.0/firebase-app.js";
import {
  getFirestore ,
  addDoc,
  collection ,
  getDocs,
  deleteDoc,
  doc,

```

```

    getDoc,
    updateDoc

} from "https://www.gstatic.com/firebasejs/10.14.0/firebase-
firestore.js"; //itens relacionado a base de dados firestone

// adicione aqui o que foi copiado em seu bloco de notas
const firebaseConfig = {

  apiKey: "XXXXXXXXXXXX",
  authDomain: "XXXXXXXXXXXX",
  projectId: "XXXXXXXXXXXX",
  storageBucket: "XXXXXXXXXXXX",
  messagingSenderId: "XXXXXXXXXXXX",
  appId: "XXXXXXXXXXXX"

};

// inicialze firebase
const app = initializeApp(firebaseConfig);

//pegando base de dados firestore do firebase
const db = getFirestore(app)

const notify = document.querySelector('.notify');

// adicionando dados na base de dados firebase

async function addData(){

  const name = document.querySelector('#name').value;
  const email = document.querySelector('#email').value;

  try {
    const DocRef = await addDoc(collection(db,"users"),{
      name:name,
      email:email
    })
    notify.innerHTML = `Data Added `
    document.querySelector('#name').value="";
    document.querySelector('#email').value="";

    setTimeout(()=>{
      notify.innerHTML = ""
    },3000)//atualizar a pagina a cada 3 segundos
  }
}

```

```

        GetData()

        } catch(error){
            console.log(error);
        }
    }

    // constante de estado do botão cadastrar
    const addBtn = document.querySelector('#add_Data')
    // evento de click no botão cadastrar
    addBtn.addEventListener('click', addData)

    // buscar dados no firestore e concatene o conteúdo na tablea do html

    async function GetData(){
        try{
            const getDataQuery = await getDocs(collection(db,"users"))
            let html = "";

            getDataQuery.forEach((doc)=>{
                const data = doc.data()

                html += `
                    <tr>
                        <td>${doc.id}</td>
                        <td>${data.name}</td>
                        <td>${data.email}</td>
                        <td><button class="del_btn"
onclick="deleteData('${doc.id}')">Excluir</button></td>
                        <td><button class="up_btn"
onclick="updateData('${doc.id}')">Atualizar</button></td>

                    </tr>
                `
            })
            document.querySelector('table').innerHTML = html

        }catch(err){
            console.log(err);
        }
    }

    GetData()

```



```

// evento de excluir usuário da base de dados
window.deleteData = async function(id){
  try{
    await deleteDoc(doc(db, "users", id))
    notify.innerHTML = "data Deleted";

    setTimeout(()=>{
      notify.innerHTML = ""
    },3000)

    getDocs()

  }catch(err){
    console.log(err);
  }
}

// atualizar dados
window.updateData = async function(id){
  try{
    const docSnapshot = await getDoc(doc(db , "users", id))
    const currentUser = docSnapshot.data()
    document.querySelector('#name').value= currentUser.name;
    document.querySelector('#email').value=currentUser.email;

    const updateDataBtn = document.querySelector('#update_data')
    updateDataBtn.classList.add('show')
    addBtn.classList.add('hide')
    updateDataBtn.addEventListener("click", async function (){
      const newName = document.querySelector('#name').value;
      const newEmail = document.querySelector('#email').value

      if( newName !== null && newEmail !== null){
        await updateDoc(doc(db,"users",id),{
          name: newName,
          email: newEmail
        })

        notify.innerHTML = "Data Updated"
        GetData()
        updateDataBtn.classList.remove('show')
        addBtn.classList.remove('hide')

        document.querySelector('#name').value="";
        document.querySelector('#email').value="";

        setTimeout(()=>{

```

```

        notify.innerHTML = ""
    },3000)

    }

    })

    }catch(err){
        console.log(err);
    }

}

```

11) Contudo do Style css

```

*{
}
.form{

}

.form input {

}

.form button{

}

.form button:hover{

}

.notify{

}


.table{

}

.table table{

}

```

```
.table table td{

}

.table table td button{

}

.del_btn{
}
.up_btn{
}

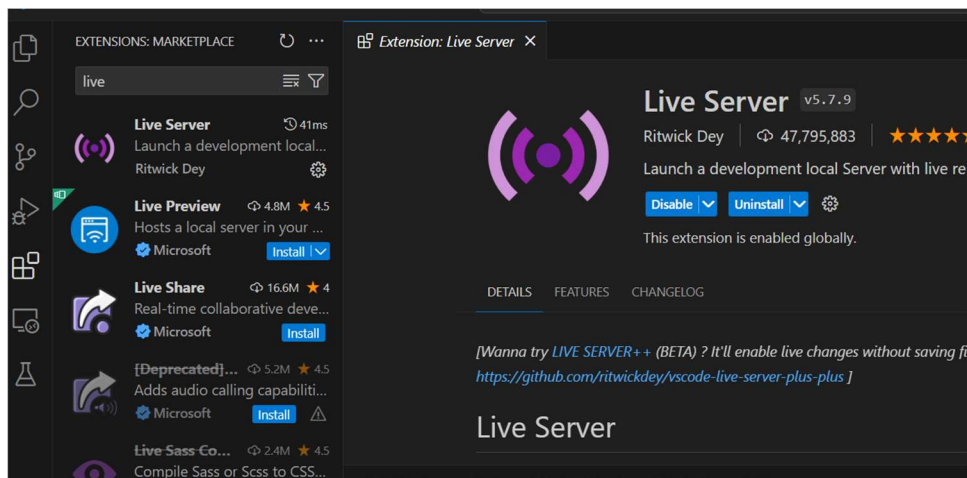
.table table td button:hover{
}

#notify{
}

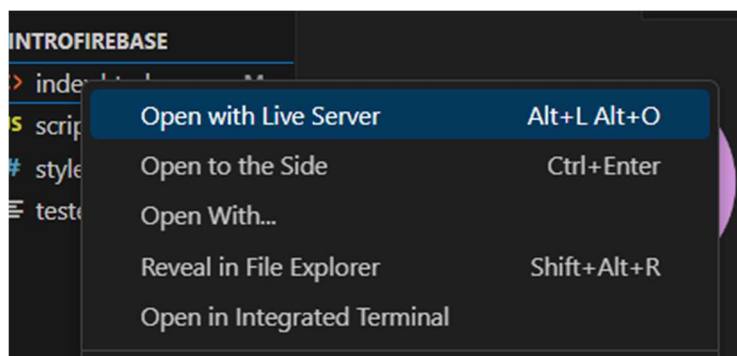
.update_data{
}

.show{
}
.hide{
}
```

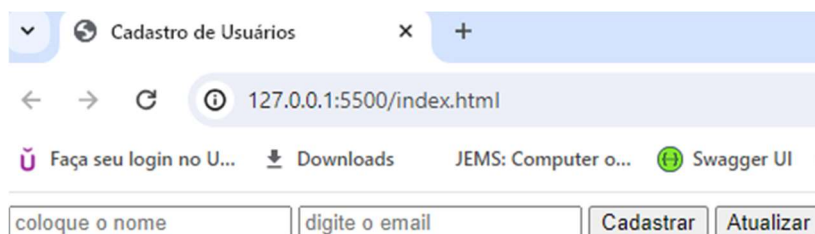
12) Instale o plugin Live Server no VS Code



13) Clique com botão esquerdo no index.html e abra com live server



14) Ao abrir o navegador dever ter uma página a seguir com os itens funcionais.



Tarefas

- I) Estude a implementação do formulário da página tutorial.
- II) Estilize os itens da página do tutorial, bolões, input, table e etc.

- III) Coloque cabeçalhos na tabela sendo id, nome e email.
- IV) Crie um popup de alerta, para que o usuário seja somente removido após pressionar ok no alerta, se caso apertar cancelar nada deve ser feito.
- V) Tendo como base o tutorial anterior. Crie uma página com o formulário a seguir com as operações de CRUD no Firebase. Escolha quatro campos principais para exibir na tabela de usuários. Além disso, ao clicar em atualizar o formulário deve mostrar os campos do usuário que deverá ser atualizado.

Nome Completo:	<input type="text" value="Nome Completo"/>	Data de Nascimento:	<input type="text" value="__/__/__"/>
RG:	<input type="text" value="00.000.000-0"/>	CPF:	<input type="text" value="000.000.000-0"/>
Pai:	<input type="text" value="Nome do Pai"/>	Mãe:	<input type="text" value="Nome da Mãe"/>
Escolaridade:	<input type="text" value="Selecione ▼"/>	Profissão:	<input type="text" value="Profissão"/>
Telefone:	<input type="text" value="0000-0000"/>	Ceular:	<input type="text" value="(00) 0000-0000"/>
E-mail:	<input type="text" value="exemplo@email.com"/>		

Obs: objeto de entrega duas páginas funcionais de Crud no Firebase.