

**INSTITUTO POLIVALENTE EDIK-RAMON**

**ENSINO PARTICULAR**

**CURSO DE INFORMÁTICA**

**Selmar Emanuel de Castro António**

**Sistema de gestão de membros de uma congregação religiosa**

**Luanda**

**2023**

**Selmar Emanuel de Castro António**

**Sistema de gestão de membros de uma congregação religiosa**

**Trabalho de conclusão de curso apresentado ao**

**Colégio Pitruca, como parte das exigências para**

**Finalização da 13ª classe do curso de informática**

**Orientador: Prof. Tomás António**

**Luanda**

**2023**

**Selmar Emanuel de Castro António**

**Sistema de gestão de membros de uma congregação religiosa**

**Trabalho de conclusão de curso apresentado ao**

**Colégio Pitruca, como parte das exigências para**

**Finalização da 13ª classe do curso de informática**

**Orientado: Prof. Tomás António**

**Luanda, aos \_\_\_\_\_ de\_\_\_\_\_de\_\_\_\_\_\_.**

**BANCA EXAMINADORA**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prof. Tomás António**

“Software e catedrais são muito parecidos; primeiro nós os construímos, depois oramos”.

(Por Anônimo)

“Que diremos, pois, a estas coisas? Se Deus é por nós, quem será contra nós?”

(Paulo de tarso)

“Só sei que nada sei”

(Sócrates)

## **“Eu vos dou um novo mandamento: amai-vos uns aos outros”**

(Jesus Cristo)

## **“Um pequeno passo para um homem, um salto gigantesco para a humanidade”**

(Neil Armstrong)

# DEDICATÓRIA

Dedico este trabalho primeiramente ao meu Deus e à sua obra, pois é para sua honra e glória. Dedico também este feito aos meus pais, nomeadamente Sérgio Edgar Lopes António e Elsa Maria Van-Dúnem de Castro António, aos meus companheiros Ismael Mucuenda, Félix Evaristo, Figueiredo Narciso, Daniela Trindade, Hélvio Santana.

À meus familiares, Edna Gomes, Emanuela Gomes, Neusa António, Abel António, Augusto Castro, Juliana Castro, Gilson Van-Dúnem, meus irmãos, Estefânio António, Hélder António, Juliana Almeida, Sérgia António, pois de vós vem a minha inspiração e motivação e permitem que o meu simples devaneio se torne em realidade.

# AGRADECIMENTOS

Aproveito a oportunidade para agradecer primeiramente a Deus, por sempre me ajudar quando preciso, por ser a minha rocha, o meu refúgio e a minha fortaleza, por ter emprestado um pouco de sua sabedoria a mim para que pudesse concluir este projeto, não esquecendo de familiares e amigos que sempre demonstraram o seu apoio a mim.

Um gratífico especial aos meus pais, pois sempre fizeram de tudo por mim e meus irmãos, a vocês digo “muito obrigado”.

# RESUMO

Este projeto, vem para solucionar o problema que o Centro Deus Forte tem, no que tange ao controle de sua membresia. Ele vem para proporcionar rapidez, eficiência e eficácia na resolução deste problema tornando o processo de controle de membros num processo automatizado e rápido.

O programa visa guardar dados de **membros baptizados** e **membros recém-convertidos,** bem como, permitindo aos usuários a visualização rápida dos dados destes, saberão também a média de conversões mensal, pois o programa é feito para que os usuários recebam emails todo final do mês, com a média de conversões.

Palavras-chave: **Membros recém-convertidos e Membros baptizados**

# ABSTARCT

This project comes to solve the problem that the Centro Deus Forte has, with regard to the control of its membership. It comes to provide speed, efficiency and effectiveness in solving this problem making the process of member control an automated and fast process.

The program aims to save data of baptized members and newly converted members, as well as, allowing the users to quickly view their data, they will also know the average of monthly conversions, because the program is made so that users receive emails every end of the month, with the average of conversions.

Keywords: **newly converted members and Baptized Members**

# SIGLÁRIO

HTML – *hyper text markup language*

JS - *javascript*

CSS – *cascading style sheet*

IP – *internet protocol*

CRUD – *creat read update and delete*

SQL – *structured query language*

API – *aplication programing interface*

TCP – *transmission control protocol*

XML- *extended markup language*

SDK – *software development kit*

CDN – *contente delivery network*

JSON – *javascript object notation*

# ÍNDICE DE FIGURAS

[*Figura 1 - Logotipo da linguagem HTML* 23](#_Toc134698681)

[Figura 2- logotipo do css 24](file:///C:\Users\emcastro\Desktop\TRABALHO%20FINAL%20selmar%20(updated.docx#_Toc134698682)

[Figura 3 - logotipo da linguagem Js 24](#_Toc134698683)

[Figura 4 - logotipo do firebase 28](#_Toc134698684)

[Figura 5 - logotipo do serviço Authentication 29](#_Toc134698685)

[Figura 6 - Tela de login 37](#_Toc134698686)

[Figura 7 - Página inicial 40](#_Toc134698687)

[Figura 8 - Formulário de registro de membros baptizados 53](#_Toc134698688)

[Figura 9 - Base de dados de novos membros 56](#_Toc134698689)

[Figura 10 - Formulário de registro de novos membros 59](#_Toc134698690)

[Figura 11 - Tabela de membros baptizados 62](#_Toc134698691)

[Figura 12 - Tabela de novos membros 69](#_Toc134698692)

# SUMÁRIO

[DEDICATÓRIA V](#_Toc135075823)

[AGRADECIMENTOS VI](#_Toc135075824)

[RESUMO VII](#_Toc135075825)

[ABSTARCT VIII](#_Toc135075826)

[SIGLÁRIO IX](#_Toc135075827)

[ÍNDICE DE FIGURAS X](#_Toc135075828)

[SUMÁRIO IV](#_Toc135075829)

[1 INTRODUÇÃO 7](#_Toc135075830)

[1.2 problemática 8](#_Toc135075831)

[1.3 Justificativa 9](#_Toc135075832)

[1.4 Hipótese 10](#_Toc135075833)

[1.5 Objetivos 11](#_Toc135075834)

[1.5.1 Geral 11](#_Toc135075835)

[1.5.2 específicos 11](#_Toc135075836)

[1.6 Diagrama de caso de uso 12](#_Toc135075837)

[1.7 Metodologia 13](#_Toc135075838)

[2 EMBASAMENTO TEÓRICO 14](#_Toc135075839)

[2.1 SISTEMA 14](#_Toc135075840)

[2.1.1 SISTEMA INFORMÁTICO 14](#_Toc135075841)

[2.2 SISTEMA DE GESTÃO 14](#_Toc135075842)

[2.3 SOFTWARE 15](#_Toc135075843)

[2.4 INTERNET 15](#_Toc135075844)

[2.5 HTML 16](#_Toc135075845)

[2.6 CSS 16](#_Toc135075846)

[2.7 JavaScript 17](#_Toc135075847)

[2.8 APIs 17](#_Toc135075848)

[2.9 Firebase 18](#_Toc135075849)

[2.9.1 Integração do *Firebase* 19](#_Toc135075850)

[*2.9.1 Authentication* 21](#_Toc135075851)

[2.9.2 Base de dados 22](#_Toc135075852)

[2.9.3 Key-value (chave-valor) 24](#_Toc135075853)

[*2.9.3 Firebase Realtime database* 24](#_Toc135075854)

[2.9.4 JSON 24](#_Toc135075855)

[2.9.5 Node.js 25](#_Toc135075856)

[2.9.6 Sendinblue 25](#_Toc135075857)

[3 Metodologia 26](#_Toc135075858)

[3.1 Metodologia de investigação científica 27](#_Toc135075859)

[3.1.2. Revisão bibliográfico 27](#_Toc135075860)

[3..1.3. Coleta de dados 27](#_Toc135075861)

[3.1.4. Análise dos dados 27](#_Toc135075862)

[3.1.5. Identificação da melhor via para solução do problema 27](#_Toc135075863)

[3.1.6. Implementação da solução 27](#_Toc135075864)

[3.1.7. Avaliação dos resultados 28](#_Toc135075865)

[4 Resultado e implementação 29](#_Toc135075866)

[4.1 O início (Login) 30](#_Toc135075867)

[4.2 Homepage 33](#_Toc135075868)

[4.3 A base de dados 43](#_Toc135075869)

[4.3.1 CDF MEMBROS BAPTIZADOS 43](#_Toc135075870)

[4.3.2 CDF NOVOS CONVERTIDOS. 49](#_Toc135075871)

[4.4 Tabela de membros baptizados 55](#_Toc135075872)

[4.5 Tabela de novos membros 61](#_Toc135075873)

[Considerações finais 66](#_Toc135075874)

[Bibliografia 67](#_Toc135075875)

# 1 INTRODUÇÃO

**Igreja**, do grego ekklesia, que significa “chamados para fora”, é um templo cristã**o**, é o local da pregação dos ensinamentos de Cristo, obedecendo os princípios da ética cristã. Igreja é o conjunto de fiéis unidos pela mesma fé e que celebram as mesmas doutrinas religiosas.

Em sociologia Igreja é um grupo religioso organizado e institucionalizado. É uma sociedade cujos membros representam da mesma maneira o mundo sagrado e suas relações com o mundo profano. Como grupo, uma igreja abrange uma comunidade dos que crêem e, geralmente um corpo de sacerdotes, hierarquizado ou não. Como instituição, a igreja representa um sistema de preceitos dogmáticos, ritos e crenças.

A igreja é uma instituição como qualquer outra e em vista da sua natureza jurídica, precisa cuidar adequadamente da documentação dos membros da igreja. Para se ter uma ideia, cerca de 2,18 bilhões de pessoas das 8,2 bilhões que ocupam ou habitam no planeta Terra dizem professar a fé cristã.

Em contexto, a igreja a quem este projeto é direcionado, alberga mais de três mil membros, logo encontra-se a necessidade de manter os registros de todos os seus membros.

## 1.2 problemática

(Sganzeria, 2021), independentemente do porte da igreja, ter um cadastro dos membros permite conhecer as pessoas que compõem a comunidade. Sem um cadastro, a igreja dificilmente saberá o tamanho da membresia, e muito menos poderá ter visão geral do perfil da congregação.

Perante os grandes desafios que a igreja tem, nas suas obrigações, não só espirituais, mas também jurídicas e sociais, um grande revés saliente na igreja é justamente a gestão de seus membros, sendo que toda gestão de membros na igreja alvo é feito de forma manual, ou seja, em papel e com isso identificam-se alguns perigos pois o papel é material mui frágil, e acarreta em si vários perigos como: Acesso não autorizado, Perda de dados, desorganização, aumento de custos, etc.

## 1.3 Justificativa

Ao observar a grande dificuldade na gestão de membros que as igrejas têm, notou-se a viabilidade de elaborar e implementar um projeto que resolvesse estes problemas.

O projeto tem então como foco, a criação de um sistema de gestão de membros, tanto **recém-convertidos** quanto **membros baptizados**, dinamizando o processo de implementação, remoção e controle dos dados de seus membros, tornando o processo de gestão de membros de uma tarefa dolorosa, manual e árdua, em um trabalho automatizado, simples e rápido.

A falta de reflexão sobre este tema fará com que a instituição continue a trabalhar de maneira antiquada, perdendo consigo a produtividade, eficácia e eficiência de trabalho.

## 1.4 Hipótese

Tendo analisado toda o ambiente em que será implementado este projeto, e as tecnologias que serão usadas na resolução do problema vigente, a hipótese levantada por esta análise é que: A gestão de membros se torne num processo automatizado.

# 1.5 Objetivos

### 1.5.1 Geral

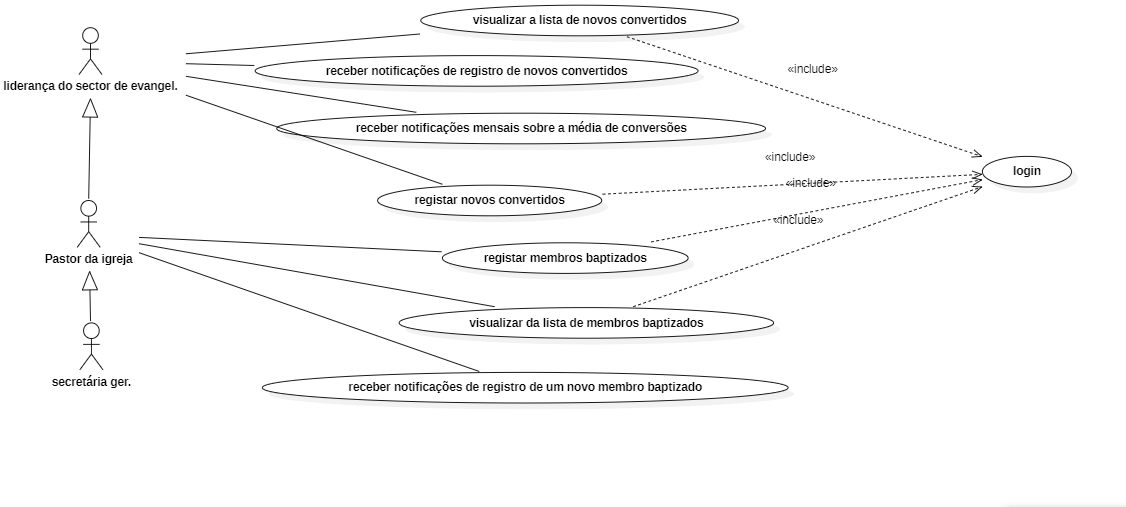
* Desenvolver um sistema de gestão de membros para uma congregação religiosa.

### 1.5.2 específicos

* Estabelecer a média de convertidos e baptizados.
* Empregar um sistema de mensagens que dará notificações automáticas sobre as médias encontradas.
* Especificar a data e hora de cada conversão ou adição na base de dados.

Registrar os nomes, data de nascimento, contacto e lugar de congregação de cada membro.

## 1.6 Diagrama de caso de uso

****

## 1.7 Metodologia

O tipo de pesquisa utilizada no presente artigo foi exploratório em relação aos objetivos, visto que, segundo Gil (1996), proporciona uma proximidade com a questão.

Neste sentido, a metodologia envolve entrevistas com pessoas que tiveram experiências ligadas diretamente com o problema pesquisado.

Os procedimentos de coleta dos dados supracitados, foi através de pesquisa bibliográfica e documental, com abordagem qualitativa, com o intuito de se ganhar uma visão mais ampla sobre o problema pesquisado.

# 2 EMBASAMENTO TEÓRICO

## 2.1 SISTEMA

Um sistema é um conjunto de elementos interdependentes de modo a formar um todo organizado. É uma definição que acontece em várias disciplinas, como [biologia](https://pt.wikipedia.org/wiki/Sistema_(biologia)), medicina, [informática](https://pt.wikipedia.org/wiki/Engenharia_de_software), [administração](https://pt.wikipedia.org/wiki/Administra%C3%A7%C3%A3o), [direito](https://pt.wikipedia.org/wiki/Direito). Vindo do grego, o termo "sistema" significa "combinar", "ajustar", "formar um conjunto".

A boa integração dos elementos componentes do *sistema* é chamada [sinergia](https://pt.wikipedia.org/wiki/Sinergia), determinando que as transformações ocorridas em uma das partes influenciará todas as outras. A alta sinergia de um sistema faz com que seja possível a este cumprir sua finalidade e atingir seu objetivo geral com eficiência; por outro lado se houver falta de sinergia, pode implicar mau funcionamento do sistema, vindo a causar inclusive falha completa, morte, falência, pane, queda do sistema, etc.

### 2.1.1 SISTEMA INFORMÁTICO

Assim sendo, é importante salientar a noção de sistema informático, este sistema é um conjunto de *hardware*, *software* bem como o suporte humano conhecido como *peopleware* que pertencem à uma organização, empresa ou instituição. Incluem computadores, os programas necessários para o processamento de dados bem como as pessoas responsáveis pela sua gestão. (PAULO, 2015, p. 49)

O sistema informático também pode ser definido como o conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informação com a finalidade de facilitar o planeamento, controlo, a coordenação, a análise e o processo decisório em empresas e outras organizações. (*Ibidem*)

## 2.2 SISTEMA DE GESTÃO

Gestão significa **gerenciamento**, **administração**, onde existe uma instituição, uma empresa, uma entidade social de pessoas, a ser gerida ou administrada.

O sistema de gestão é um *software* que permite gerir diferentes tarefas, processos e informações que são operacionalizadas dentro de uma empresa. Dessa forma, ele é uma plataforma inteligente que automatiza vários processos operacionais e conecta as áreas da empresa.

## 2.3 SOFTWARE

A palavra *software* é usada para designar a parte intangível do computador, o *software* pode ser divido em dois tipos:

1. *Software* de sistema é o sistema operativo, a primeira camada de software que consiste num aglomerado de instruções que tornam o hardware num sistema com o qual o usuário pode interagir.
2. *Software* de aplicação são os restantes tipos de aplicativos com o qual o usuário pode realizar tarefas diversas, tais programas incluem folhas de cálculo, sistema de gestão de base de dados, jogos, etc.

## 2.4 INTERNET

A internet é um conjunto de redes de computadores interligadas, cujo funcionamento é regido por um conjunto de protocolos e serviços que permitem a interação dos usuários conectados local ou mundialmente.

També conhecida como “rede das redes” a internet do ponto de vista técnico é um conjunto de enlaces físicos e lógicos que interligam sistemas computacionais por todo mundo.

O projeto da internet é de origem estadunidense, tendo sido formulada nos anos sessenta, fruto do conflito histórico conhecido como a guerra fria, foi desenhada para ser uma rede de comunicação que permitisse a partilha de informação entre vários computadores espalhados pelos Estados unidos. E era imprescindível que na eventualidade da erupção de uma guerra nuclear, mesmo que um dos seus hosts fosse destruído, a rede continuasse funciona.

Representa então o desenvolvimento do projeto de *U.S Department of Defense – Advanced Research Projects Agency* (atualmente conhecida como *DARPA*) por volta dos anos 70, designado por *Arpanet,* que surgiu com o objetivo de partilhar informação com instituições científicas e universidades envolvida no projeto.

Em 1973, foi criado um protocolo de comunicações *standard* designado por TCP/IP, que permitiu a comunicação entre redes públicas e privadas, independentemente dos sistemas operativos, do hardware e dos próprios canais de comunicação (linha telefónica, fibra óptica, linhas dedicadas). (António, 2015, p. 191)

## 2.5 HTML

HTML (Linguagem de Marcação de Hipertexto) é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web. Outras tecnologias além do HTML geralmente são usadas para descrever a aparência/apresentação ([CSS](https://developer.mozilla.org/pt-BR/docs/Web/CSS)) ou a funcionalidade/comportamento ([JavaScript](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript)) de uma página da web.

"Hipertexto" refere-se aos links que conectam páginas da *Web* entre si, seja dentro de um único site ou entre sites. Links são um aspeto fundamental da *web*. Ao carregar conteúdo na Internet e vinculá-lo a páginas criadas por outras pessoas, você se torna um participante ativo na *world wide web.*



*Figura 1 - Logotipo da linguagem HTML*

## 2.6 CSS

CSS  (*Cascading Style Sheets* ou Folhas de Estilo em Cascata)  é uma linguagem de [estilo](https://developer.mozilla.org/en-US/docs/Web/API/StyleSheet) usada para descrever a apresentação de um documento escrito em [HTML](https://developer.mozilla.org/pt-BR/docs/Web/HTML) ou em [XML](https://developer.mozilla.org/pt-BR/docs/Glossary/XML)  (incluindo várias linguagens em XML como [SVG](https://developer.mozilla.org/pt-BR/docs/Web/SVG), [MathML](https://developer.mozilla.org/pt-BR/docs/Web/MathML) ou [XHTML](https://developer.mozilla.org/pt-BR/XHTML)). O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias.

CSS é uma das principais linguagens da open web e é padronizada em navegadores web de acordo com as [especificação da W3C](https://w3.org/Style/CSS/#specs). Desenvolvido em níveis, o CSS1 está atualmente obsoleto, o CSS2.1 é uma recomendação e o [CSS3](https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS3), agora dividido em pequenos módulos, está progredindo para a sua padronização.



**Figura 2- logotipo do css**

## 2.7 JavaScript

*JavaScript*(às vezes abreviado para **JS**) é uma linguagem leve, interpretada e baseada em objetos com [funções de primeira classe](http://en.wikipedia.org/wiki/First-class_function), mais conhecida como a linguagem de script para páginas Web, mas usada também em [vários outros ambientes sem browser](http://en.wikipedia.org/wiki/JavaScript#Uses_outside_web_pages), tais como [node.js](https://nodejs.org/), [Apache CouchDB](https://couchdb.apache.org/) e Adobe Acrobat. O JavaScript é uma linguagem [baseada em protótipos](https://en.wikipedia.org/wiki/Prototype-based), [multi-paradigma](https://en.wikipedia.org/wiki/Programming_paradigm) e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos (como por exemplo a programação funcional).

****

Figura 3 - logotipo da linguagem Js

## 2.8 APIs

A sigla API deriva da expressão inglesa *Application Programming Interface* que, traduzida para o português, pode ser compreendida como uma interface de programação de aplicação. Ou seja, API é um conjunto de normas que possibilita a comunicação entre plataformas por meio de uma série de padrões e protocolos.

Por meio de APIs, desenvolvedores podem criar novos *softwares* e aplicativos capazes de se comunicar com outras plataformas. Por exemplo: caso um desenvolvedor queira criar um aplicativo de fotos para [Android](https://www.techtudo.com.br/tudo-sobre/android/), ele poderá ter acesso à câmera do celular através da API do sistema operacional, sem ter a necessidade de criar uma nova interface de câmera do zero.

A função de uma API é, basicamente, facilitar e simplificar o trabalho de desenvolvedores, além de oferecer um padrão para a criação de novas plataformas. Com o uso das APIs, não é necessário criar códigos personalizados para cada função que um programa for executar, o que simplifica a criação de novos aplicativos, *softwares* e plataformas em geral.

Além disso, as APIs também possuem papel fundamental quando o assunto é segurança, já que também são capazes de bloquear acesso e permissões a dados de *software* e hardware que algumas aplicações não podem usar.

As APIs estão presentes na maioria dos aplicativos que utilizamos no nosso dia a dia. No [WhatsApp](https://www.techtudo.com.br/tudo-sobre/whatsapp-messenger/), por exemplo, podemos perceber a integração da lista de contactos salva no dispositivo com os contactos do aplicativo. No [Facebook](https://www.techtudo.com.br/tudo-sobre/facebook/), temos a integração com o [Instagram](https://www.techtudo.com.br/tudo-sobre/instagram/), que [permite que fotos postadas no aplicativo também sejam postadas automaticamente no Facebook](https://www.techtudo.com.br/dicas-e-tutoriais/2018/03/como-compartilhar-fotos-do-instagram-no-seu-facebook.ghtml).

Também podemos observar o uso das APIs quando realizamos compras online, já que a plataforma utilizada para pagamentos — em que colocamos as informações de cartões de crédito — deve ser integrada à operadora do cartão, que pode ou não autorizar a compra. Esses são alguns exemplos de APIs mais comuns, que podemos perceber no nosso cotidiano, e que facilitam a comunicação entre plataformas.

Neste projeto utilizaremos o *api* “*Sendinblue”*

## 2.9 Firebase

*Firebase* é um conjunto de serviços disponibilizados pela Google, para gestão de back-end, e front-end, isto em nuvem. Os serviços do *Firebase* tornam o trabalho do desenvolvedor num passeio no parque, tornando os aplicativos tanto web como nativos(mobile), em obras muito mais maleáveis. Alguns dos serviços que o *Firebase* oferece são: *Authentication, Firebase firestore, Firebase Realtime database, Google analytics, cloud messaging, etc.* Para este projeto apenas iremos usar o *Firebase Authentication* e o *Firebase Realtime database.*

## 2.9.1 Integração do *Firebase*

Para começar o projeto *Firebase*, eis os passos a tomar:

1. **Cria um projeto do *Firebase***

Antes de adicionar o *Firebase* ao app em JavaScript, é necessário que criar e registrar o app nesse projeto. Ao registrar o app no Firebase, você receberá um objeto de configuração do Firebase que será usado para conectar o app aos recursos do projeto do *Firebase*.

1. **Criar um projeto do *Firebase***

No console do Firebase, clique em adicionar projeto.

* Para adicionar recursos do Firebase a um projeto do Google cloud existente, digite o nome dele ou selecione-o no menu superior
* Para criar um novo projeto, digite o nome dele. Também é possível editar o ID do projeto exibido abaixo do nome dele.

OBS: O Firebase gera um ID exclusivo para o projeto do Firebase com base no nome que você atribuir a ele. Se você quiser editar esse ID, faça isso agora, já que não é possível alterá-lo depois que o Firebase provisionar recursos para o projeto. Consulte [Noções básicas sobre projetos do Firebase](https://firebase.google.com/docs/projects/learn-more?hl=pt#project-id) para saber mais sobre como o Firebase usa o ID do projeto.

* Leia, e aceite os termos do *Firebase.*
* Clique em continuar
* (opcional) configure o Google Analytics para o projeto e tenha uma experiência ideal quando usar qualquer um destes produtos do *Firebase*

1. **Registrar o app**

Depois de criar o projeto do *Firebase*, é possível registrar o app da web nele.

* No centro da página de visão geral do projeto no console do *Firebase*, clique no ícone da web (</>)  para iniciar o fluxo de trabalho de configuração. Se você já adicionou um app ao projeto do Firebase, clique em adicionar app para exibir as opções da plataforma
* Digite o apelido do app. Esse apelido é um identificador interno de conveniência e só é visível no console do *Firebase*.
* Clique em registrar app.
* Siga as instruções exibidas na tela para adicionar e inicializar o SDK do Firebase no app.

1. **Inicializar o Firebase**

Há duas maneiras de integrar o *Firebase* à um projeto *web*, isto é, instalando o *sdk* pelo npm, ou adicionando o cdn. Neste projeto inicializamos o Firebase usando o cdn, nos permitindo assim importar apenas os recursos que iriamos precisar

Para incluir produtos ou serviços específicos (como, Authentication e cloud firestore), vamos adicionar o seguinte código na tag <body> do nosso documento HTML, e isso antes de usar qualquer produto do *Firebase*:

<body>  
  <!—inserir isto no fim do nosso documento e antes de usar qualquer produto ou serviço do Firebase -->  
  <script type="module">  
    import { initializeApp } from 'https://www.gstatic.com/firebasejs/9.21.0/firebase-app.js'

    // adicione os serviços que pretende utilizar  
    import { getAuth } from 'https://www.gstatic.com/firebasejs/9.21.0/firebase-auth.js'  
    import { getFirestore } from 'https://www.gstatic.com/firebasejs/9.21.0/firebase-firestore.js'  
  </script>  
</body>

* adicione o objeto de configuração, e depois inicialize o Firebase no seu app:

<body>  
  <script type="module">  
    // ...  
  
    **// OBS: substituir com o objeto que foi atribuído ao seu projeto  
    const firebaseConfig = {  
      // ...  
    };  
  
    // Inicializar Firebase  
    const app = initializeApp(firebaseConfig);**  
  </script>  
</body>

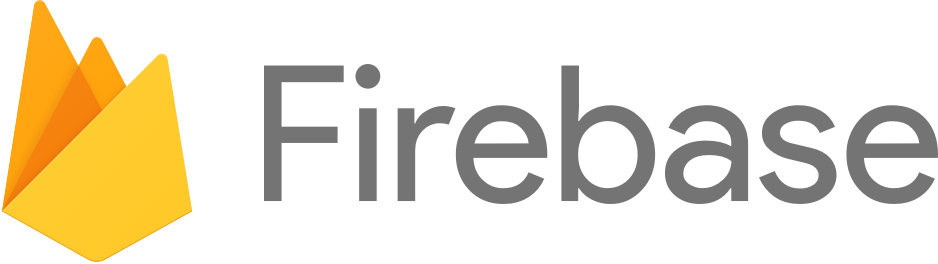
****

Figura 4 - logotipo do firebase

### *2.9.1 Authentication*

A maioria dos *apps* precisa reconhecer a identidade do usuário. Ter essa informação permite que um *app* salve os dados do usuário na nuvem com segurança e forneça a mesma experiência personalizada em todos os dispositivos do usuário.

O Firebase Authentication fornece serviços de *back-end*, *SDKs* fáceis de usar e bibliotecas de IU prontas para autenticar usuários no seu aplicativo. Ele oferece suporte à autenticação usando senhas, números de telefone, provedores de identidade federados conhecidos, como *Google*, *Facebook* e *Twitter*, entre outros.

O Firebase Authentication tem integração com outros serviços do Firebase e aproveita os padrões do setor, como OAuth 2.0 e *OpenID* *Connect*, para que possa ser facilmente integrado ao seu back-end personalizado.

Para adicionar e inicializar o SDK Authentication, é necessário importar pelo CDN e após isso, inicializar o Firebase Authentication e ter uma referência do serviço, como visto a seguir:

import { initializeApp } from "firebase/app";  
**import { getAuth } from "firebase/auth";**  
  
// **OBS: substituir com o objeto que foi atribuído ao seu projeto**  
const firebaseConfig = {  
  // ...  
};  
  
// Inicializar o Firebase  
const app = initializeApp(firebaseConfig);  
  
 **//** inicializar o Firebase Authentication e ter uma referência do serviço

**const auth = getAuth(app);**

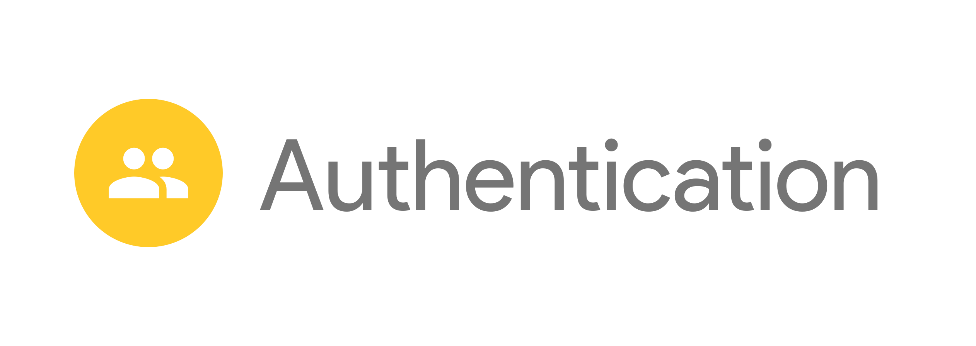
****

Figura 5 - logotipo do serviço Authentication

### 2.9.2 Base de dados

(Ivan de Sousa, 2020), o banco de dados é a organização e armazenagem de informações sobre um domínio específico. De forma mais simples, é o agrupamento de dados que tratam do mesmo assunto, e que precisam ser armazenados para segurança ou conferência futura.

É comum que empresas tenham diversas informações que precisam ser organizadas e disponibilizadas dentro do negócio para que sejam consultadas posteriormente pela equipe e pela gerência.

Por isso, é interessante ter um sistema de gerenciamento de banco de dados, SGBD, para conseguir manipular as informações e tornar a rotina da empresa muito mais simples.

#### 2.9.2.1 Tipos de base de dados

Existem vários tipos de base de dados disponíveis no mercado. Mas, primeiro é necessário entendermos a diferença entre a base de dados relacionais e não relacionais

##### 2.9.2.1.1. Base de dados relacional

As bases de dados relacionais são criadas no paradigma da orientação a conjuntos. Dessa forma, os dados que ali estão disponíveis serão armazenados em tabelas. Cada tabela terá atributos e linhas ou registros responsáveis por organizar essas informações.

São comumente utilizados para dados tabulares, que possuem sua inserção muito mais simples e permite, também, a recuperação de forma mais prática no dia a dia. A linguagem utilizada nesse formato é de SQL. Portanto, se você quer utilizar um banco de dados relacional, é preciso se atentar a esse detalhe. (Ivan de Sousa, 2020)

##### 2.9.2.1.2. Base de dados não relacional

A base de dados não relacional é feita para suprir as demandas que a base de dados relacional não consegue suprir. Um exemplo de demandas são aqueles dados mistos, onde se misturam tabelas, imagens e mapas, por exemplo, que não poderão ser tabulados em colunas e linhas de tabela. (ibidem)

Esta tipo de base de dados, não possuem chaves, tabelas nem colunas. Ao invés disso, o Banco de Dados não Relacional usa um modelo de armazenamento otimizado para o tipo de dado que está sendo armazenado.

Há quatro tipos de modelos na base de dados não relacional, nomeadamente:

* **Key – Value (chave - valor):** como o nome indica ele simplesmente possui uma coleção de pares chave-valor em um objeto.
* **Document:** este tipo, utiliza uma string para denominar uma base de dados no formato json (que também pode ser encontrada no formato json). Os documentos não precisam manter a estrutura idêntica, gerando um ganho em flexibilidade.
* **Wide-column:** Este tipo é semelhante ao banco de dados relacional. Ele armazena seus dados em colunas flexíveis que podem ser compartilhadas em múltiplos servidores e bancos de dados.
* **Graph:** Este tipo é o mais complexo dentro dos quatro citados. Ele é projetado para armazenar de forma eficiente as relações entre as entidades.

Para fazer consultas em um banco de dados NoSQL do tipo graph, você pode usar um conceito chamado OGM, que pode ser implementado usando diversas linguagens de programação, como JavaScript, .NET, PHP e JAVA. (João Ferreira)

### 2.9.3 Key-value (chave-valor)

Para este projeto decidiu-se que será usado uma base de dados não relacional, baseado no modelo chave-valor.

### *2.9.3 Firebase Realtime database*

O *realtime database (*base de dados em tempo real*),* do *firebase* é uma base de dados hospedada na nuvem. Os seus dados são guardados como json e sincronizados para cada cliente.

### 2.9.4 JSON

JSON significa “*javascript object notation*”, que traduzindo para o português significa “notação de objetos do javascript”, é um formato leve de troca de dados entre servidores e clientes. Em outras palavras, é uma sintaxe de serialização de objetos, matrizes, números, strings, booleanos, e null, o json se baseia na sintaxe do javascript, mas são distintas, nem todo javascript é json, e nem todo json é javascript.

A especificação JSON surgiu por volta do ano 2000, mas só passou a fazer parte da linguagem javascript após o lançamento do EcmaScript. Atualmente esse formato é suportado por diversos tipos de linguagem de programação, além de ser uma alternativa mais leve que o modelo XML. (Ivan de Sousa)

Os dados contidos em um arquivo no formato JSON devem ser estruturados por meio de uma coleção de pares com nome e valor ou ser uma lista ordenada de valores. Seus elementos devem conter:

**chave**: corresponde ao identificador do conteúdo. Por isso, deve ser uma string delimitada por aspas;

**valor**: representa o conteúdo correspondente e pode conter os seguintes tipos de dados: string, array, object, number, boolean ou null.

### 2.9.5 Node.js

Node.js é um ambiente multiplataforma em javascript que permite aos desenvolvedores produzirem aplicações para rede e *server-Side* usando o javascript.

#### 2.9.5.1 Nodemailer

O [Nodemailer](https://nodemailer.com/about/) é um módulo do Node.js que permite enviar e-mails para o seu servidor com facilidade. Quando você quiser se comunicar com seus usuários ou apenas ser notificado quando algo der errado, uma das opções para fazer isso seria usando o e-mail. (Ben Rachel)

Neste projeto, utilizamos o Nodemailer, em comunhão com o api “Sendinblue” para enviar e-mails automaticamente.

### 2.9.6 Sendinblue

A Sendinblue é uma das ferramentas mais populares para automatizar uma estratégia de email de forma prática e eficiente.

# 3 Metodologia

O capítulo de metodologia é um dos pilares fundamentais de qualquer trabalho acadêmico. Nesta seção, descrevemos as etapas metodológicas adotadas para alcançar os objetivos propostos e responder às perguntas de pesquisa formuladas no início do trabalho. A metodologia descreve o caminho percorrido, as técnicas utilizadas, as análises e os resultados obtidos, o que contribui para a validade e a confiabilidade do estudo.

O capítulo de metodologia tem como objetivo principal descrever detalhadamente como o trabalho foi realizado, a fim de permitir a replicação e a validação dos resultados. Para tanto, esta seção é dividida em quatro partes principais: a descrição do objeto de estudo, a abordagem metodológica, a coleta de dados e a análise dos dados.

## 3.1 Metodologia de investigação científica

Este projeto de pesquisa utiliza uma abordagem qualitativa, tendo em vista que se buscou compreender os fenômenos a partir de suas explicações e motivações, tendo em vista a necessidade de compreender o problema do Centro Deus Forte, no que se refere ao controle de sua membresia. A seguir, serão apresentadas as etapas da metodologia utilizada neste trabalho.

### 3.1.2. Revisão bibliográfico

Foi realizada uma revisão bibliográfica a partir de fontes relevantes para a compreensão do tema, como livros, artigos científicos, com o objetivo de embasar a pesquisa e permitir a compreensão teórica do problema em questão.

### 3..1.3. Coleta de dados

Foram realizadas entrevistas com pessoas que tiveram ligação direta com o problema pesquisado, isto é, liderança do Centro Deus Forte.

### Análise dos dados

Os dados coletados foram analisados qualitativamente, com o objetivo de identificar as principais características do problema e as possíveis soluções.

### 3.1.5. Identificação da melhor via para solução do problema

Com base nos dados coletados e analisados, foi identificada a melhor via para solucionar o problema do controle de membresia do Centro Deus Forte. Foram considerados aspetos como a eficiência, eficácia, rapidez e custo-benefício da solução proposta.

### Implementação da solução

A solução proposta foi implementada, utilizando-se um sistema de gestão de membros, que permite o controle e a atualização dos dados dos membros de forma automatizada e rápida. Foi realizada uma capacitação com os responsáveis pelo controle de membresia do Centro Deus Forte para utilização do sistema.

### Avaliação dos resultados

Após a implementação da solução, foi realizada uma avaliação dos resultados, verificando se a solução proposta atendeu às expectativas e solucionou o problema do Centro Deus Forte. Foram considerados aspetos como a eficiência, eficácia e satisfação dos usuários.

Com base na revisão bibliográfica, na coleta e análise de dados, foi possível identificar a melhor via para solucionar o problema e implementar a solução proposta, permitindo o controle de membresia de forma automatizada e eficiente. A metodologia utilizada foi essencial para garantir a autenticidade, confiabilidade e valor científico deste trabalho.

# 4 Resultado e implementação

Neste capítulo, apresentamos os resultados práticos obtidos a partir da aplicação da metodologia descrita anteriormente. Aqui, descrevemos como implementamos o conhecimento adquirido a fim de solucionar o problema vigente.

Os resultados são apresentados em dois níveis: o primeiro nível descreve os dados coletados e as análises realizadas, enquanto o segundo nível apresenta as ações práticas tomadas a partir dos resultados obtidos.

No primeiro nível, apresentamos os dados coletados, incluindo a amostra, as técnicas de coleta e as principais variáveis. Descrevemos também as análises realizadas, como a estatística descritiva e as correlações, que permitiram entender as relações entre as variáveis e as características da amostra.

No segundo nível, descrevemos as ações práticas tomadas a partir dos resultados obtidos. Isso incluiu a implementação de soluções para o problema vigente, bem como a adoção de novas práticas e políticas que visam melhorar o desempenho organizacional.

Além disso, detalhamos como as ações foram implementadas, incluindo os recursos necessários, as etapas do processo e as equipes envolvidas. Por fim, destacamos os resultados das ações práticas, avaliando o impacto das soluções propostas e as melhorias alcançadas.

Em resumo, este capítulo descreve as ações práticas tomadas para solucionar o problema vigente, incluindo as etapas do processo e os resultados obtidos. As soluções propostas foram baseadas nos resultados da análise de dados e visam melhorar o desempenho organizacional.

## 4.1 O início (Login)

Ao entrar no site, o usuário primeiramente depara-se com uma tela de login, pois lhe é requerido fornecer suas credenciais e só assim poderá ele Acessar os recursos do site, salvo se suas credenciais forem validadas, caso contrário, o acesso lhe é negado. A tela de login é apresentada na figura abaixo:

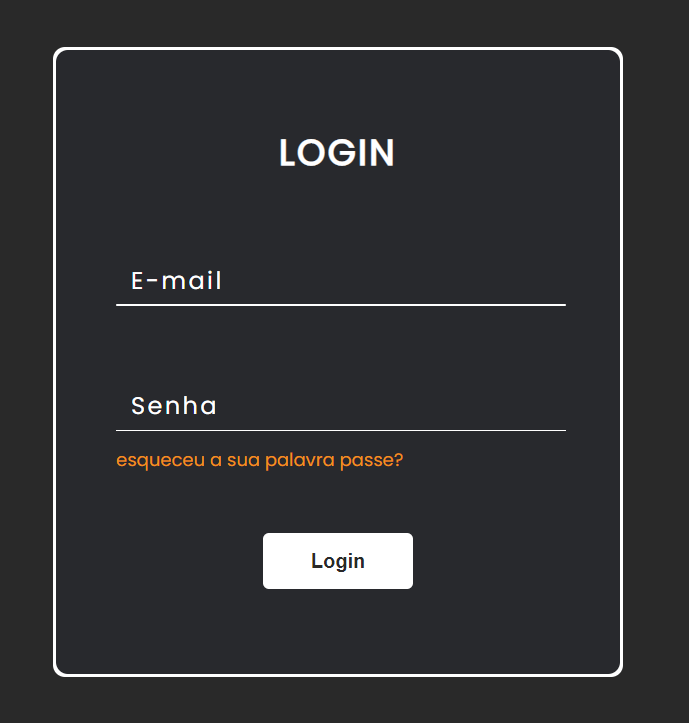


Figura 6 - Tela de login

O código fonte (HTML), é mostrado abaixo:

<! DOCTYPE HTML>

<HTML Lang="en">

<head>

    <link rel="stylesheet" href="index.css">

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login CDF</title>

</head>

<body>

    <div class="box">

        <div class="form">

            <h2>LOGIN</h2>

                <div class="inputBox">

                    <input type="email" required="required" id="userEmail">

                    <span>E-mail</span>

                    <i></i>

                </div>

            <div class="inputBox">

                <input type="password" required="required" id="userPassword">

                <span>Senha</span>

                <i></i>

            </div>

            <div class="links">

                <a href="">esqueceu a sua palavra passe?</a>

            </div>

            <button type="submit" class="btn" id="signin">Login</button>

        </div>

    </div>

<script type="module" src="index.js"></script>

</body>

</html>

Logo a seguir é o script, que nos permite importar os recursos do firebase, e utilizar os recursos de autenticação e login.

  // Import the functions you need from the SDKs you need

  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-app.js";

  import { getAuth, signInWithEmailAndPassword, onAuthStateChanged, signOut } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-auth.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 = {

// estes dados foram apagados para salvaguardar a integridade deste projeto

  };

  // Initialize Firebase

  const app = initializeApp(firebaseConfig);

  const auth = getAuth(app);

  const userEmail = document.getElementById("userEmail");

  const userPassword = document.getElementById("userPassword");

  const signin= document.getElementById("signin");

  // sign in

  const userSignin = async() => {

      const signinEmail = userEmail.value

      const signinPassword = userPassword.value

     signInWithEmailAndPassword(auth, signinEmail, signinPassword )

      .then((userCredential) => {

          const user = userCredential.user;

          console.log(user);

          window.location.assign("/home/home.html");

      })

      .catch((error)=> {

          const errorCode = error.code

          const errorMesage = error.message

          console.log(errorCode + errorMesage)

          alert(error);

      })

  }

      // checking auth atate, verifica o estado de auth

      const checkAuthState = async() =>{

          onAuthStateChanged(auth, user =>{

              if(user){

                  //se o user estiver logado, n faz nada, permite apenas a entrada do mesmo.

              }else{

                 //se não, redireccione-o para a página de login.

              }

          })

      }

      const userSignOut = async() =>{

          await signOut(auth)

      }

      checkAuthState();

      console.log(auth)

  userPassword.addEventListener('keypress', function(event) {

      if(event.key === 'Enter'){

          userSignin();

      }

  })

  signin.addEventListener('click', userSignin)

## 4.2 Homepage

Conceda que o usuário é legítimo para fazer uso do sistema, após o login, ele é redirecionado à página inicial, que é apresentada a seguir:



Figura 7 - Página inicial

Eis o código fonte(HTML) para esta página:

<!DOCTYPE html>

<html lang="en">

<head>

    <script src="https://kit.fontawesome.com/a44fb1f0d1.js" crossorigin="anonymous"></script>

    <!-- CSS only -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx" crossorigin="anonymous">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@500&family=Montaga&family=Poppins&display=swap" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@500&family=Montaga&display=swap" rel="stylesheet">

    <link rel="stylesheet"href="https://fonts.googleapis.com/css?family=Font+Name">

    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet">

    <link rel="stylesheet" href="home.css">

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge"><meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Centro DEUS FORTE</title>

</head>

<body>

<div class="container"><div class="wrapper">

    <div class="loader">

        <div class="inner one"></div>

        <div class="inner two"></div>

        <div class="inner three"></div>

      </div>

</div>

    <i class="fa-solid fa-right-from-bracket" id="signout"></i>

        <img src="logo\_ADPM-removebg.png" alt="">

        <p class="intro">CENTRO <span class="df">DEUS FORTE</span></p>

        <P class="intro"> Quem você deseja <span class="registrar">registrar?</span> </P>

        <div class="FLOW">

            <div class="float">

                <i class="fa-solid fa-user"></i>

                <p>Membro Baptizado?</p>

                <button id="registrarMB-btn" class="reg-btn" onclick="MB()"><a href="/MB/MembrosBaptizados.html">Registar</a></button>

            </div>

            <div class="float2">

                <i class="fa-solid fa-user"></i>

                <p>Recém Convertido?</p>

                <buttton id="registrarNBM-btn" class="reg-btn"><a href="/NMB/NovosMb.html">Registar</a></BUTton>

            </div>

        </div>

        <div class="verLista">

            <h1>ou visualize as suas listas</h1>

            <div class="listaDeMembros">

                <i class="fa-solid fa-list"></i>

                <p>Recém convertidos</p>

                <a href="/listaNm/listapageNm.html"><button id="registrarMB-btn" class="reg-btn" onclick="MB()">VER</button></a>

            </div>

            <div class="listaDeNovos">

                <i class="fa-solid fa-list"></i>

                <p>Membros Baptizados</p>

                <a href="/ListaMb/listapageMb.html"><button id="registrarMB-btn" class="reg-btn" onclick="MB()">VER</button></a>

            </div>

        </div>

</div>

<!-- JavaScript Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDKKtQBHUuLZ9AsSv4jD4Xa" crossorigin="anonymous"></script>

<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

<!-- <script> -->

<script type="module" src="home.js"></script>

</body>

</html>

Esta página foi feita com ajuda do *bootstrap, fonteawesome e Google fonts.*

O código css é o que se segue:

body::-webkit-scrollbar{

  display: none;

   /\* for chrome and safari \*/

}

body{

    -ms-overflow-style: none;  /\* Internet Explorer and Edge \*/

    scrollbar-width: none;  /\* Firefox \*/

    margin: 0;

    padding: 0;

    background: url(hands-holding-cross-while-praying.jpg);

    background-size: cover;

    background-repeat: no-repeat;

    display: flex;

    text-align: center;

    overflow: visible;

}

.wrapper{

  display: flex;

  justify-content: center;

  align-items: center;

  position: absolute;

  top: 0;

  left: 0;

  width: 100%;

  height: 100%;

  background:#292929 ;

}

.loader {

  position: absolute;

  top: calc(50% - 32px);

  left: calc(50% - 32px);

  width: 70px;

  height: 70px;

  border-radius: 50%;

  perspective: 800px;

}

.inner {

  position: absolute;

  box-sizing: border-box;

  width: 100%;

  height: 100%;

  border-radius: 50%;

}

.inner.one {

  left: 0%;

  top: 0%;

  animation: rotate-one 1s linear infinite;

  border-bottom: 3px solid #EFEFFA;

}

.inner.two {

  right: 0%;

  top: 0%;

  animation: rotate-two 1s linear infinite;

  border-right: 3px solid #EFEFFA;

}

.inner.three {

  right: 0%;

  bottom: 0%;

  animation: rotate-three 1s linear infinite;

  border-top: 3px solid #EFEFFA;

}

@keyframes rotate-one {

  0% {

    transform: rotateX(35deg) rotateY(-45deg) rotateZ(0deg);

  }

  100% {

    transform: rotateX(35deg) rotateY(-45deg) rotateZ(360deg);

  }

}

@keyframes rotate-two {

  0% {

    transform: rotateX(50deg) rotateY(10deg) rotateZ(0deg);

  }

  100% {

    transform: rotateX(50deg) rotateY(10deg) rotateZ(360deg);

  }

}

@keyframes rotate-three {

  0% {

    transform: rotateX(35deg) rotateY(55deg) rotateZ(0deg);

  }

  100% {

    transform: rotateX(35deg) rotateY(55deg) rotateZ(360deg);

  }

}

#signout{

  margin-top: 2em;

  float: left;

  transition: 0.6s;

}

#signout:hover{

  cursor: pointer;

  transform: scale(1.9);

}

.container{

    color:#fff;

}

.light{

  float: right;

  display: flex;

   width: 30px;

   height: 30px;

   cursor: pointer;

   margin-top: 5px;

}

img{

    width: 200px;

    height: 200px;

}

.registrar{

    background: url(//s2.svgbox.net/pen-brushes.svg?ic=brush-1&color=ffff43);

    margin: -2px -6px;

    padding:  2px  6px;

    color: black;

}

.df{

    margin: -2px -6px;

    padding:  2px  6px;

    background: url();

}

.intro{

    font-family: 'Josefin Sans', sans-serif;

    font-family: 'Montaga', serif;

    font-family: 'Poppins', sans-serif;

    font-weight: bolder;

    font-size: 50px;

    margin-bottom: 50px;

}

.FLOW{

  display: flex;

}

.float2{

    width: 50%;

    float: right;

    padding: 20px;

    text-align: center;

    font-family: 'Josefin Sans', sans-serif;

    font-family: 'Montaga', serif;

    font-family: 'Poppins', sans-serif;

    font-weight: bold;

    font-size: 30px;

}

.float{

    width: 50%;

    float: left;

    padding: 20px;

    border-right: 1px solid yellow;

    text-align: center;

    font-family: 'Josefin Sans', sans-serif;

    font-family: 'Montaga', serif;

    font-family: 'Poppins', sans-serif;

    font-weight: bold;

    font-size: 30px;

}

/\* CSS \*/

.reg-btn{

  align-items: center;

  background-color: #FFFFFF;

  border: 1px solid rgba(0, 0, 0, 0.1);

  border-radius: .25rem;

  box-shadow: rgba(0, 0, 0, 0.02) 0 1px 3px 0;

  box-sizing: border-box;

  color: rgba(0, 0, 0, 0.85);

  cursor: pointer;

  display: inline-flex;

  font-family: system-ui,-apple-system,system-ui,"Helvetica Neue",Helvetica,Arial,sans-serif;

  font-size: 16px;

  font-weight: 600;

  justify-content: center;

  line-height: 1.25;

  margin: 0;

  min-height: 3rem;

  padding: calc(.875rem - 1px) calc(1.5rem - 1px);

  position: relative;

  text-decoration: none;

  transition: all 250ms;

  user-select: none;

  -webkit-user-select: none;

  touch-action: manipulation;

  vertical-align: baseline;

  width: auto;

}

.reg-btn:hover,

.reg-btn:focus {

  border-color: rgba(0, 0, 0, 0.15);

  box-shadow: rgba(0, 0, 0, 0.1) 0 4px 12px;

  color: rgba(0, 0, 0, 0.65);

}

.reg-btn:hover {

  transform: translateY(-2px);

  transition: 0.1s;

  background-color: yellow;

  color: #292929;

}

.reg-btn:active {

  background-color: #F0F0F1;

  border-color: rgba(0, 0, 0, 0.15);

  box-shadow: rgba(0, 0, 0, 0.06) 0 2px 4px;

  color: rgba(0, 0, 0, 0.65);

  transform: translateY(0);

}

a{

  text-decoration: none;

  color: black;

}

a:hover{

  color: black;

}

.verLista{

  height: 80vh;

  text-align: center;

}

.verLista h1{

  margin-top: 50px;

  margin-bottom: 90px;

}

#arrow{

 width: 80px;

 height: 100px;

 margin-right: -700px;

}

.listaDeMembros{

  width: 50%;

  float: right;

  padding: 20px;

  text-align: center;

  font-family: 'Josefin Sans', sans-serif;

  font-family: 'Montaga', serif;

  font-family: 'Poppins', sans-serif;

  font-weight: bold;

  font-size: 30px;

}

.listaDeNovos{

  width: 50%;

  float: left;

  padding: 20px;

  border-right: 1px solid yellow;

  text-align: center;

  font-family: 'Josefin Sans', sans-serif;

  font-family: 'Montaga', serif;

  font-family: 'Poppins', sans-serif;

  font-weight: bold;

  font-size: 30px;

}

A seguir, é apresentado o script que nos permite importar recursos do firebase, como autenticação, e nos traz interatividade na página:

    // Import the functions you need from the SDKs you need

    import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-app.js";

    import { getAuth, signInWithEmailAndPassword, onAuthStateChanged, signOut } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-auth.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 = {

     // estes dados foram apagados para salvaguardar a integridade do projeto

    };

    // Initialize Firebase

    const app = initializeApp(firebaseConfig);

    const auth = getAuth(app);

    // end

    const userEmail = document.getElementById("userEmail");

    const userPassword = document.getElementById("userPassword");

    const signout= document.getElementById("signout");

        // o código a seguir, tem a função de verificar o estado de auth, se o usuário estiver logado, apenas permite o acesso, senão, redireciona-o para a tela de login

        const checkAuthState = async() =>{

            onAuthStateChanged(auth, user =>{

                if(user){}

                else{

                 window.location.assign("/login/index.html")

                }

            })

        }

        // este código é uma função assíncrona, ela espera um evento de signout, que é ativado quando o usuário prima no botão signout, e termina a sua sessão.

        const userSignOut = async() =>{

            await signOut(auth)

        }

        checkAuthState();

        signout.addEventListener('click', userSignOut)

       window.onload(

        document.querySelector('.wrapper').style.display='none')

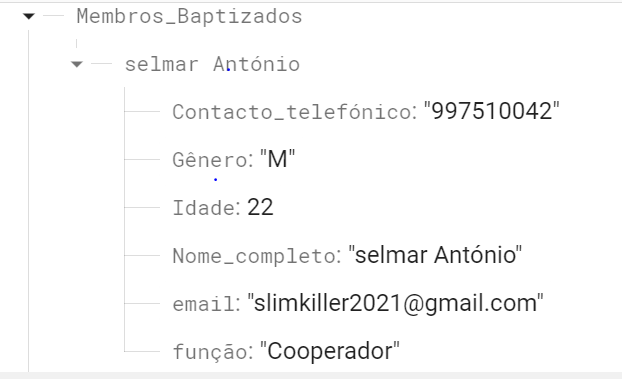
## 4.3 A base de dados

Como já supracitado, decidimos criar uma base de dados NoSQL, ou seja, não relacional, que nos é disponibilizada pelo firebase, é necessário ressaltar que, foram criadas duas bases de dados separadas, cujos nomes são: **CDF MEMBROS BAPTIZADOS** e **CDF NOVOS CONVERTIDOS.**

### 4.3.1 CDF MEMBROS BAPTIZADOS

É nesta base de dados que guardamos os dados dos membros baptizados, os dados que são guardados são: Nome, idade, gênero, email, contacto telefónico e a função do membro.

Eis a seguir uma imagem que ilustra como a base de dados está organizada.



**Obs: é relevante ressaltar que os dados mostrados neste documento, são dados experimentais, e não são dados reais da instituição a quem este projeto é alvo.**

É realmente imprescindível que se guarde os dados dos membros baptizados, poias são eles que trabalham, cooperam, e mantêm o funcionamento da igreja.

O código abaixo é o código fonte do formulário que faz o registro dos membros baptizados

<!DOCTYPE html>

<html lang="en">

<head>

    <script src="https://kit.fontawesome.com/a44fb1f0d1.js" crossorigin="anonymous"></script>

    <!-- CSS only -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx" crossorigin="anonymous">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@500&family=Montaga&family=Poppins&display=swap" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@500&family=Montaga&display=swap" rel="stylesheet">

    <!-- <link rel="stylesheet"href="https://fonts.googleapis.com/css?family=Font+Name"> -->

    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet">

    <link rel="stylesheet" href="MembrosBaptizados.css">

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

        <div class="wrapper">

            <div class="title">Formulário de registro de membros baptizados</div>

            <div class="form">

            <div class="input\_field">

                <label for="">NOME COMPLETO</label>

                   <input id="nome" type="text" class="input" name="nome">

                </div>

            <div class="input\_field">

                <label for="">Número de telefone</label>

                   <input id="fone" type="text" class="input" name="number">

            </div>

            <div class="input\_field">

                <label for="">Data de nascimento</label>

                   <input type="date" class="input" name="birth" id="data">

                   <p id="result"></p>

            </div>

            <div class="input\_field">

                <label for="">E-mail</label>

                   <input type="text" class="input" name="email" id="email">

            </div>

            <div class="input\_field">

                <label for="">Gênero</label>

                   <select name="" id="sexo">

                    <option value="M" name="gender">Masculino</option>

                    <option value="F" name="gender" >Feminino</option>

                   </select>

            </div>

            <div class="input\_field">

                <label for="">Função</label>

                   <select  id="Funcao">

                    <optionvalue="Cooperadorname="função">Cooperador</option>

                    <option value="Pastor" name="função">Pastor</option>

                    <option value="Diácono" name="função">Diácono</option>

                    <option value="Ancião" name="função">Ancião</option>

                   </select>

            </div>

            <div class="input\_field">

                <label for="">Morada</label>

                   <textarea id="morada" class="textarea" name="morada"></textarea>

            </div>

        </div>

        <CEnter><button class="reg-btn" id="registrar">REGISTRAR</button></CEnter>

        <CEnter><button class="reg-btn" id="backbtn" >Voltar</button></CEnter>

        </div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDKKtQBHUuLZ9AsSv4jD4Xa" crossorigin="anonymous"></script>

<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

<script type="module" src="MembrosBaptizados.js"></script>

</body>

</html>

e o resultado deste código é mostrado na figura abaixo:

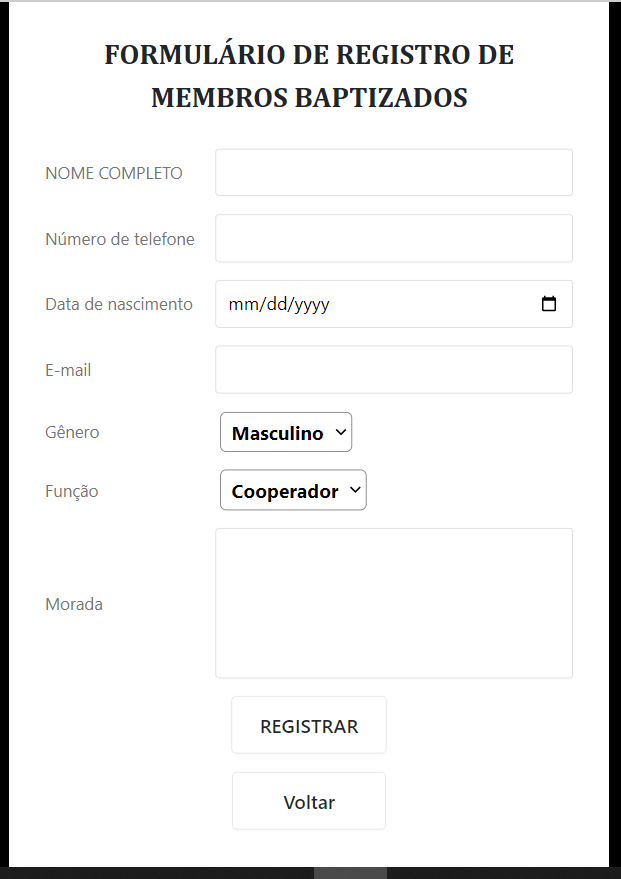


Figura 8 - Formulário de registro de membros baptizados

O código a seguir é o código fonte (JS), que nos permite importar os serviços do firebase, fazer a integração com a base de dados e gravar os dados na base de dados, confira abaixo:

// Import the functions you need from the SDKs you need

import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-app.js";

import { getDatabase, ref, set } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-database.js";

import { getAuth, signInWithEmailAndPassword, onAuthStateChanged, signOut } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-auth.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: // estes dados foram escondidos para manter a segurança do projeto

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

const db = getDatabase(app);

const auth = getAuth(app);

const checkAuthState = async() =>{

        onAuthStateChanged(auth, user =>{

            if(user){

            }else{

                // location.assign('my\_project/login/index.html')

            }

        })

    }

    const userSignOut = async() =>{

        await signOut(auth)

    }

    checkAuthState();

    console.log(auth)

//   calcula a idade

    function calculateAge(data\_) {

        const dob = new Date(data\_); //uma nova data é guarg«dada nessa variável, a data é que foi o input no cadastro

        const ageDiffMs = Date.now() - dob.getTime(); // acha a diferença em meses

        const ageDate = new Date(ageDiffMs);

        const idade = Math.abs(ageDate.getUTCFullYear() - 1970);

        return idade;

      }

// fim da função de calcular idade

    let entradas = 0;

export function writeUserData(nome, fone, sexo, local, facebook, data, funcao, email) {

   let Memblist = [];

    entradas++ // incrementa o valor de entradas por um sempre que a função é chamada

    const db = getDatabase(app);

    let registrar = document.getElementById('registrar');

    let nomes = document.getElementById('nome').value;

    let fone\_ = document.getElementById('fone').value;

    let sexo\_ = document.getElementById('sexo').value;

    let email\_= document.getElementById('email').value;

    let data\_ = document.getElementById("data").value;

    let funcao\_ = document.getElementById('Funcao').value;

    let morada = document.getElementById('morada').value;

 Memblist.push([nome, fone, sexo, local, facebook, data, funcao, email]);

const idade = calculateAge(data\_); // call the calculateAge function to get the age

    // Increment the 'entradas' counter in the database

    set(ref(db, "entradas"), entradas);

    set(ref(db, "Membros\_Baptizados/" +" " + nomes), {

            Nome\_completo: nomes,

            Contacto\_telefónico: fone\_,

            Idade : idade,

            Gênero : sexo\_,

            função : funcao\_,

            email : email\_,

    })

    .then(() => {

        alert("salvo com sucesso")

    })

    .catch((error) =>{

        alert(error)

        console.log(error)

    })

    }

    registrar.addEventListener('click', writeUserData);

    document.getElementById("backbtn").addEventListener('click', ()=>{

        window.history.back();

})

### 4.3.2 CDF NOVOS CONVERTIDOS.

É nesta base de dados que guardamos os dados dos membros baptizados, os dados que são guardados são: Nome, idade, gênero, email, contacto telefónico e lugar de conversão

Eis a seguir uma imagem que ilustra como a base de dados está organizada.

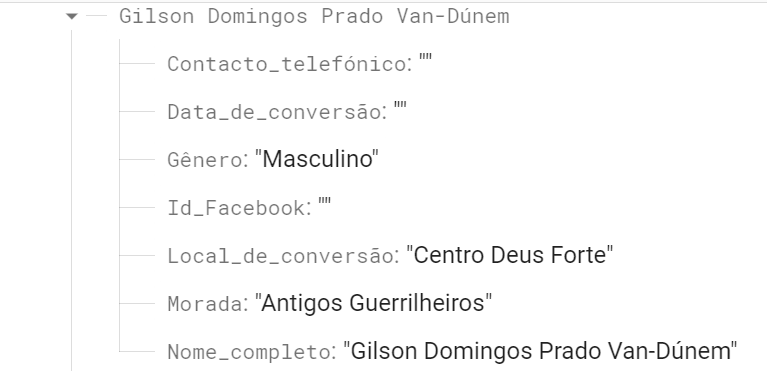


Figura 9 - Base de dados de novos membros

Abaixo, eis o código fonte do formulário para registro de novos membros:

<!DOCTYPE html>

<html lang="en">

<head>

    <script src="https://kit.fontawesome.com/a44fb1f0d1.js" crossorigin="anonymous"></script>

    <!-- CSS only -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx" crossorigin="anonymous">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@500&family=Montaga&family=Poppins&display=swap" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@500&family=Montaga&display=swap" rel="stylesheet">

    <!-- <link rel="stylesheet"href="https://fonts.googleapis.com/css?family=Font+Name"> -->

    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet">

    <link rel="stylesheet" href="NovosMb.css">

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>4

<div class="wrapper">

    <div class="title">Formulário de registro de novos membros</div>

    <div class="form">

        <div class="input\_field">

        <label for="">NOME COMPLETO</label>

        <input type="text" class="input" id="nome-el">

    </div>

    <div class="input\_field">

        <label for="">Número de telefone</label>

        <input type="text" class="input" id="fone-el">

    </div>

    <div class="input\_field">

        <label for="">Gênero</label>

        <select name="" id="sexo">

            <option value="Masculino">Masculino</option>

            <option value="Feminino">Feminino</option>

        </select>

    </div>

    <div class="input\_field">

        <label for="">Lugar de conversão</label>

        <select id="local">

            <option value="Centro Deus Forte">Centro Deus Forte</option>

            <option value="Tabernáculo da glória">Tabernáculo da glória</option>

            <option value="Tenda de oração">Tenda de oração</option>

            <option value="Deus é Fiel">Deus é Fiel</option>

            <option value="Deus é amor">Deus é amor</option>

            <option value="Coluna de fogo">Coluna de fogo</option>

        </select>

        </div>

        <div class="input\_field">

            <label for="">Data de conversão</label>

            <input type="date" class="input" id="data-el">

        </div>

        <div class="input\_field">

            <label for="">Facebook ID </label>

            <input type="text" class="input" id="fbId">

        </div>

        <div class="input\_field">

            <label for="">Morada</label>

            <textarea class="textarea" id="morada"></textarea>

        </div>

          <CEnter><button class="reg-btn" id="registrar">REGISTRAR</button></CEnter>

          <CEnter><button class="reg-btn" id="backbtn" >Voltar</button></CEnter>

</div>

    <!-- JavaScript Bundle with Popper -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDKKtQBHUuLZ9AsSv4jD4Xa" crossorigin="anonymous"></script>

    <script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

    <script type="module" src="NovosMb.js"></script>

    <script type="module" src="calc\_med.js"></script>

</body>

</html>

E tendo este código, eis o resultado:

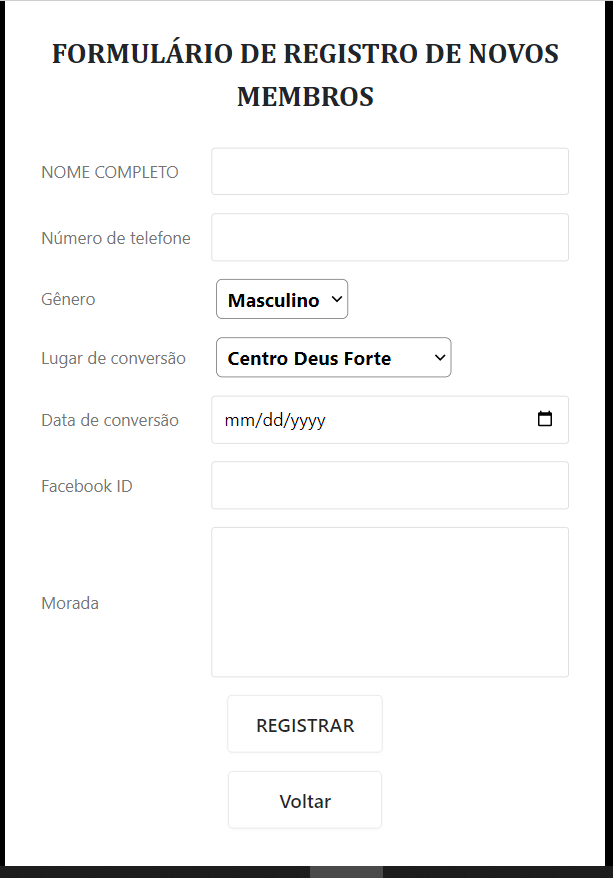


Figura 10 - Formulário de registro de novos membros

O código a seguir é o código fonte (JS), que nos permite importar os serviços do firebase, faer a integração com a base de dados e gravar os dados na base de dados, confira abaixo:

 // Import the functions you need from the SDKs you need

   import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-app.js";

   import { getDatabase, ref, set, child, update, remove } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-database.js";

   import { getAuth, signInWithEmailAndPassword, onAuthStateChanged, signOut } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-auth.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: // estes dados foram apaados, para salvaguargar a integridade

deste projeto.

   // Initialize Firebase

   const app = initializeApp(firebaseConfig);

   const db = getDatabase(app);

   const auth = getAuth(app);

        // checking auth atate

        const checkAuthState = async() =>{

            onAuthStateChanged(auth, user =>{

                if(user){}

                else{

                // window.location.assign("my\_project/login/index.html")

                }

            })

        }

        checkAuthState();

        console.log(auth)

   export function writeUserData(nome, fone, sexo, local, facebook, data) {

       const db = getDatabase(app);

       let registrar = document.getElementById('registrar');

       let nomes = document.getElementById('nome-el').value;

       let fone\_ = document.getElementById('fone-el').value;

       let sexo\_ = document.getElementById('sexo').value;

       let local\_ = document.getElementById('local').value;

       let facebook\_= document.getElementById('fbId').value;

       let data\_ = document.getElementById("data-el").value;

       let morada = document.getElementById("morada").value;

            // Increment the 'entradas' counter in the database

        set(ref(db, "entradas"), entradas);

        set(ref(db, "Novos\_Convertidos/" + nomes), {

               Nome\_completo: nomes,

               Contacto\_telefónico: fone\_,

               Data\_de\_conversão : data\_,

               Gênero : sexo\_,

               Local\_de\_conversão : local\_,

               Id\_Facebook : facebook\_,

               Morada: morada

       })

       .then(() => {

           alert("salvo com sucesso")

        //    location.reload();

       })

       .catch((error) =>{

           alert(error)

           console.log(error)

       })

       }

const registrar = document.getElementById("registrar")

registrar.addEventListener('click', writeUserData);

document.getElementById("backbtn").addEventListener('click', ()=>{

        window.history.back();

})

## 4.4 Tabela de membros baptizados

Não basta apenas guardar os dados, é necessário também que os usuários possam visualizar os dados guardados, e os dados são guardados em forma de tabelas.

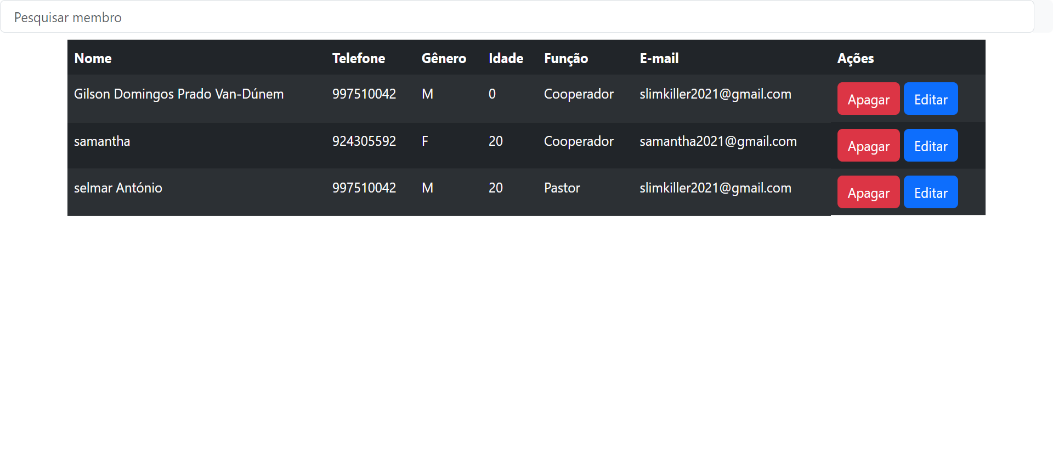


Figura 11 - Tabela de membros baptizados

Eis o código fonte(HTML) desta página:

<!DOCTYPE html>

<html lang="en">

<head>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">

  <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="stylelistMb.css">

    <title>Document</title>

</head>

<body>

    <div class="input-group rounded">

        <input  type="search" id="search-inp" class="form-control rounded" placeholder=" Pesquisar membro" aria-label="Search" aria-describedby="search-addon" />

        <span class="input-group-text border-0" id="search-addon">

          <i class="fas fa-search"></i>

        </span>

      </div>

    <h3 id="findName"></h3>

    <h3 id="findfone"></h3>

    <h3 id="finddata"></h3>

    <h3 id="findgenero"></h3>

    <h3 id="findfuncao"></h3>

    <h3 id="findemail"></h3>

<div class="container">

        <table class="table table-striped table-dark">

            <thead>

                <th>Nome</th>

                <th>Telefone</th>

                <th>Gênero</th>

                <th>Idade</th>

                <th>Função</th>

                <th>E-mail</th>

                <th>Ações</th>

            </thead>

            <tbody id="tbody1">

            </tbody>

        </table>

<!-- Button trigger modal -->

<!-- <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#staticBackdrop">

Launch static backdrop modal

</button> -->

<!-- Modal for delete -->

<div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">

  <div class="modal-dialog">

    <div class="modal-content">

      <div class="modal-header">

        <center><h1 class="modal-title fs-5" id="staticBackdropLabel">Tem a certeza que pretende apagar este registro?</h1></center>

        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

      </div>

      <div class="modal-body">

        <p>Após concluir esta ação, os dados deste membro não poderão ser recuperados</p>

      </div>

      <div class="modal-footer">

        <button type="button" class="btn btn-primary"  data-bs-dismiss="modal">Cancelar</button>

        <button type="button" id="apaga" class="btn btn-danger" >apagar</button>

      </div>

    </div>

  </div>

</div>

       </div>

<!-- end of modl  -->

<!-- modal for edit -->

<!-- Modal -->

<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

  <div class="modal-dialog">

    <div class="modal-content">

      <div class="modal-header">

        <h1 class="modal-title fs-5" id="exampleModalLabel">Editar </h1>

        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

      </div>

      <div class="modal-body">

        <div class="wrapper">

          <div class="form">

          <div class="input\_field">

              <label for="">NOME COMPLETO</label>

                 <input id="nome" type="text" class="input" name="nome">

              </div>

          <div class="input\_field">

              <label for="">Número de telefone</label>

                 <input id="fone" type="text" class="input" name="number">

          </div>

          <div class="input\_field">

              <label for="">Data de nascimento</label>

                 <input type="date" class="input" name="birth" id="data">

                 <p id="result"></p>

          </div>

          <div class="input\_field">

              <label for="">E-mail</label>

                 <input type="text" class="input" name="email" id="email">

          </div>

          <div class="input\_field">

              <label for="">Gênero</label>

                 <select name="" id="sexo">

                  <option value="M" name="gender">Masculino</option>

                  <option value="F" name="gender" >Feminino</option>

                 </select>

          </div>

          <div class="input\_field">

              <label for="">Função</label>

                 <select  id="Funcao">

                  <option value="Cooperador" name="função">Cooperador</option>

                  <option value="Pastor" name="função">Pastor</option>

                  <option value="Diácono" name="função">Diácono</option>

                  <option value="Ancião" name="função">Ancião</option>

                 </select>

          </div>

          <div class="input\_field">

              <label for="">Morada</label>

                 <textarea id="morada" class="textarea" name="morada"></textarea>

          </div>

      </div>

      <div class="modal-footer">

        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

        <button type="button" class="btn btn-primary">Salvar</button>

      </div>

    </div>

  </div>

</div>

<!-- end of modal -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/js/bootstrap.min.js" integrity="sha384-heAjqF+bCxXpCWLa6Zhcp4fu20XoNIA98ecBC1YkdXhszjoejr5y9Q77hIrv8R9i" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>

<script type="module" src="scriptmb.js"></script>

</body>

</html>

Em seguida, eis o script que nos permite importar os recursos do firebase, buscar os dados na base de dados e mostrar eles em tabelas:

    // Import the functions you need from the SDKs you need

    import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-app.js";

    import { getDatabase, ref, set, child, update, remove, get, onValue,push} from "https://www.gstatic.com/firebasejs/9.17.2/firebase-database.js";

    import { getAuth, signInWithEmailAndPassword, onAuthStateChanged, signOut } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-auth.js";

    // import { writeUserData} from "../MB/MembrosBaptizados.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: "AIzaSyArcPHlC5iCokwdwl8Vgwx4Og2edN7T6-c",

        authDomain: "first-firebase-test-31e09.firebaseapp.com",

        databaseURL: "https://first-firebase-test-31e09-default-rtdb.firebaseio.com",

        projectId: "first-firebase-test-31e09",

        storageBucket: "first-firebase-test-31e09.appspot.com",

        messagingSenderId: "114644799892",

        appId: "1:114644799892:web:0cbc816d5858a14c54ff81"

    };

    // Initialize Firebase

    const app = initializeApp(firebaseConfig);

    const db = getDatabase(app);

    const auth = getAuth(app);

    const dataRef = ref(db, "Membros\_Baptizados/");

    const checkAuthState = async() =>{

            onAuthStateChanged(auth, user =>{

                if(user){

                }else{

                    // window.location.assign('/login/index.html')

                }

            })

        }

        const userSignOut = async() =>{

            await signOut(auth)

        }

        checkAuthState();

        console.log(auth)

    const dataTableBody = document.getElementById("tbody1");

    // ----------------------- GET ALL DATA FROM DATABASE---------------------

    onValue(dataRef, (snapshot)=>{

        const data = snapshot.val();

        tbody1.innerHtml = "";

        snapshot.forEach((childSnapshot)=>{

        const key = childSnapshot.key;

        const childData = childSnapshot.val();

        const row = document.createElement("tr", key, childData);

        row.setAttribute("data-key", key); // Add data-key attribute

        const btns = document.createElement('button')

        row.innerHTML =`

        <td>${childData.Nome\_completo}</td>

        <td>${childData.Contacto\_telefónico}</td>

        <td>${childData.Gênero}</td>

        <td>${childData.Idade}</td>

        <td>${childData.função}</td>

        <td>${childData.email}</td>

        ` ;

    console.log(childData)

    dataTableBody.appendChild(row);

    let ctrlDiv = document.createElement("div")

    ctrlDiv.innerHTML = '<button type="button" class="btn btn-danger" id="apagar" onclick="getID(this.id)">Apagar</button>' + ' '

    ctrlDiv.innerHTML += '<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">Editar</button>';

    row.appendChild(ctrlDiv);

});

});

    // -------------- END OF FETCH----------------------

    // ---------------DELETE FUNCTION-----------------

    const apagar = document.getElementById('apagar')

    function deleteRow(row) {

        const key = row.getAttribute("data-key");

        const rowRef = child(dataRef, key);

        remove(rowRef)

          .then(() => {

            console.log("Row deleted successfully");

          })

          .catch((error) => {

            console.error("Error deleting row: ", error);

          });

        row.remove();

      }

    apagar.addEventListener('click', deleteRow);

## 4.5 Tabela de novos membros



Figura 12 - Tabela de novos membros

Eis o código fonte(HTML) desta página:

<!DOCTYPE html>

<html lang="en">

<head>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="stylelistNm.css">

    <title>Document</title>

</head>

<body>

    <div class="input-group rounded">

        <span class="input-group-text border-0" id="search-addon">

          <i class="fas fa-search"></i>

        </span>

      </div>

    <h3 id="findName"></h3>

    <h3 id="findfone"></h3>

    <h3 id="finddata"></h3>

    <h3 id="findgenero"></h3>

    <h3 id="findfuncao"></h3>

    <h3 id="findemail"></h3>

<div class="container">

        <table class="table table-striped table-dark">

            <thead>

                <th >Nome</th>

                <th>Telefone</th>

                <th>Gênero</th>

                <th>Lugar de conversão</th>

                <th>Data de conversão</th>

                <th>Id Facebook</th>

                <th>Morada</th>

                <th>Ações</th>

            </thead>

            <tbody id="tbody1">

            </tbody>

        </table>

<!-- Modal -->

<div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">

  <div class="modal-dialog">

    <div class="modal-content">

      <div class="modal-header">

        <center><h1 class="modal-title fs-5" id="staticBackdropLabel">Tem a certeza que pretende apagar este registro?</h1></center>

        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

      </div>

      <div class="modal-body">

        <p>Após concluir esta ação, os dados deste membro não poderão ser recuperados</p>

      </div>

      <div class="modal-footer">

        <button type="button" class="btn btn-primary"  data-bs-dismiss="modal">Cancelar</button>

        <button type="button" class="btn btn-danger" onclick="apagar()">APAGAR</button>

      </div>

    </div>

  </div>

</div>

       </div>

       </div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/js/bootstrap.min.js" integrity="sha384-heAjqF+bCxXpCWLa6Zhcp4fu20XoNIA98ecBC1YkdXhszjoejr5y9Q77hIrv8R9i" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>

<script type="module" src="script.js">

 </script>

</body>

</html>

O script que nos permite ligar com a base de dados e buscar os dados lá guardados, e mostrá-los em tabela:

 // Import the functions you need from the SDKs you need

 import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-app.js";

 import { getDatabase, ref, set, child, update, remove, get, onValue } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-database.js";

 import { getAuth, signInWithEmailAndPassword, onAuthStateChanged, signOut } from "https://www.gstatic.com/firebasejs/9.17.2/firebase-auth.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: "AIzaSyDZZZVZyVf2kBFlsraWzgb-frcL2Im8qxk",

     authDomain: "novo-test-c620e.firebaseapp.com",

     databaseURL: "https://novo-test-c620e-default-rtdb.firebaseio.com",

     projectId: "novo-test-c620e",

     storageBucket: "novo-test-c620e.appspot.com",

     messagingSenderId: "165155012476",

     appId: "1:165155012476:web:94207d8b96cc0f2785aef9",

 };

 // Initialize Firebase

 const app = initializeApp(firebaseConfig);

 const db = getDatabase(app);

 const auth = getAuth(app);

 const dataRef = ref(db, "Novos\_Convertidos/");

 const checkAuthState = async() =>{

         onAuthStateChanged(auth, user =>{

             if(user){

             }else{

                window.location.assign('13ª project/login/index.html')

             }

         })

     }

     const userSignOut = async() =>{

         await signOut(auth)

     }

     checkAuthState();

     console.log(auth)

// ----------------------- GET ALL DATA FROM DATABASE---------------------

const dataTableBody = document.getElementById("tbody1");

 onValue(dataRef, (snapshot)=>{

         const data = snapshot.val();

         tbody1.innerHtml = "";

         snapshot.forEach((childSnapshot)=>{

             const childData = childSnapshot.val();

             const row = document.createElement("tr");

             row.innerHTML =  `

         <td>${childData.Nome\_completo}</td>

         <td>${childData.Contacto\_telefónico}</td>

         <td>${childData.Gênero}</td>

         <td>${childData.Local\_de\_conversão}</td>

         <td>${childData.Data\_de\_conversão}</td>

         <td>${childData.Id\_Facebook}</td>

         <td>${childData.Morada}</td>

         `;

         dataTableBody.appendChild(row);

            let ctrlDiv = document.createElement("div")

            ctrlDiv.innerHTML = '<button type="button" class="btn btn-danger" data-bs-toggle="modal" data-bs-target="#staticBackdrop">Apagar</button>' + ' '

            ctrlDiv.innerHTML += '<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#staticBackdrop">Editar</button>'

            row.appendChild(ctrlDiv)

     });

});

// -------------- END OF FETCH----------------------

# Considerações finais

Ao longo deste trabalho, foi possível constatar a importância de um sistema de gestão de membros para uma congregação religiosa. O Centro Deus Forte enfrentava dificuldades em controlar sua membresia de forma eficiente e eficaz, o que motivou o desenvolvimento de um sistema que automatizasse esse processo.

O sistema desenvolvido permitiu que os usuários do centro tivessem acesso rápido e fácil aos dados dos membros batizados e recém-convertidos, além de receberem relatórios mensais com a média de conversões. Isso permitiu que a congregação religiosa pudesse acompanhar de perto o desempenho de suas atividades e tomar decisões estratégicas com base em dados precisos e atualizados.

Ao final deste trabalho, pode-se concluir que a implementação de um sistema de gestão de membros é uma estratégia importante para o sucesso de uma congregação religiosa. Além de permitir um controle mais eficiente dos dados dos membros, um sistema automatizado também pode trazer benefícios em termos de redução de custos e aumento da eficiência operacional.

Portanto, é fundamental que as congregações religiosas invistam em tecnologias que possam ajudá-las a gerenciar seus dados e informações de forma mais eficiente, visando sempre melhorar a qualidade dos serviços prestados aos seus membros e o alcance de seus objetivos institucionais.

# Bibliografia

(s.d.).

António, P. F. (2015). *Informática e tecnologias de informação.* Sílabo.

Coelho, B. (24 de fevereiro de 2020). *Quer aprender a delimitar a metodologia TCC?* Obtido de blog.mettzer.com: https://blog.mettzer.com/metodologia-tcc/

*developer.mozilla.* (s.d.). Obtido de Express/Node introduction: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\_Nodejs/Introduction

*Firebase documentation - realtime database.* (25 de 03 de 2023). Obtido de firebase: https://firebase.google.com/docs/database?authuser=0&hl=pt

*Json.* (s.d.). Obtido de Modzilla developer network - MDN web docs: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\_Objects/JSON

musselman, v. (12 de Agosto de 2014). *voltemos ao evangelho*. Obtido em 14 de janeiro de 2023, de https://voltemosaoevangelho.com/blog/2014/08/igrejaeclesia-significa-chamados-para-fora-no-novo-testamento/

*nodejs.docs.* (s.d.). Obtido de nodejs: https://nodejs.org/en/docs

Rachel, T. B. (20 de Março de 2023). *freecodecamp.* Obtido de Como usar o Nodemailer para enviar e-mails do seu servidor do Node.js: https://www.freecodecamp.org/portuguese/news/como-usar-o-nodemailer-para-enviar-emails-do-seu-servidor-do-node-js/

team, G. f. (26 de agosto de 2022). *Documentação do firebase - authentication.* Obtido de Firebase.com: https://firebase.google.com/docs/auth?authuser=0&hl=pt

Tumelero, N. (15 de 02 de 2023). *metodologia-tcc.* Obtido de blog.mettzer: https://blog.mettzer.com/metodologia-tcc/