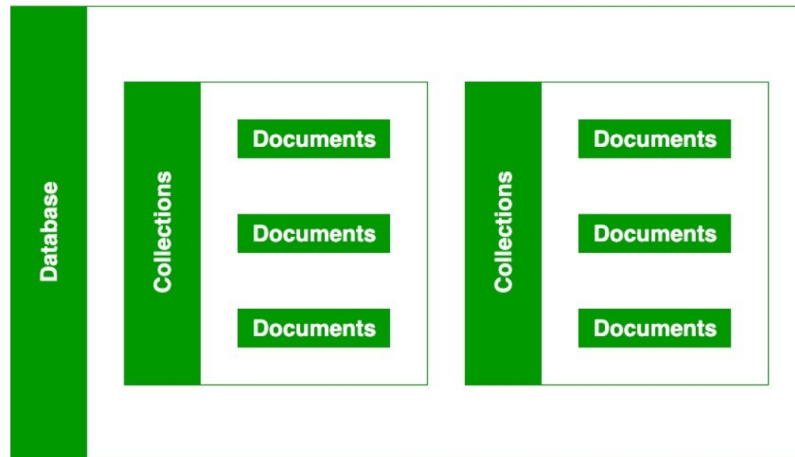


1) Escolha do modelo de BD (relacional, chave-valor, grafos, ...) e do SGBD usado (qual software de banco de dados será usado)

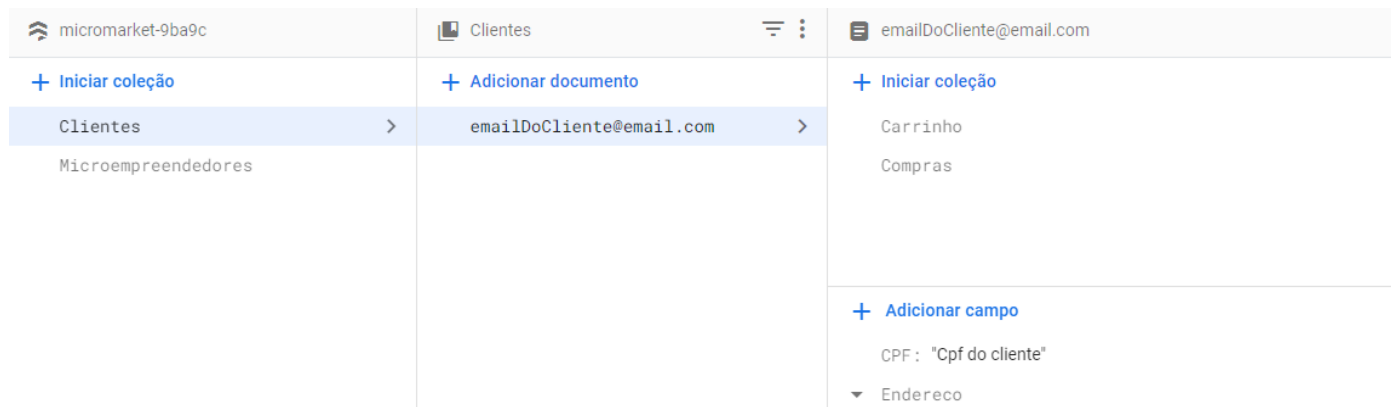
Modelo: Banco de Dados NoSQL, baseado em Documentos e Coleções. SGBD: Firebase Firestore, da Google.



2) Modelo entidade-relacionamento e esquema relacional.

Coleções e Documentos:

No banco de dados do sistema denominado como "MicroMarket", existem duas principais coleções "Clientes" e "Microempreendedores". Pela intuição, essas coleções são usadas para guardar os clientes e os microempreendedores que vão se cadastrar no sistema. Cada coleção possui documentos que são responsáveis por guardar os dados daquele documento. Exemplo: Coleção clientes



Neste exemplo, a Coleção principal "Clientes" possui um Documento chamado emailDoCliente@email.com, sendo o e-mail o "id" de cada cliente (o mesmo vale para os microempreendedores). Esse documento possui dados, sendo esses CPF (String), Endereco (um "map" (similar aos objetos do JavaScript)), Telefone (String), email (String), nome (String) e senha (String). Esse documento em questão também possui suas próprias coleções, sendo essas "Carrinho", "Compras" e "Mensagens". Cada uma dessas coleções possui Documentos que

possuem informações. O mesmo vale para os Microempreendedores, porém, com as coleções "Produtos", "Vendas" e "Mensagens".

3) Scripts de criação do banco de dados.

CRUD com Firebase Firestore e Nodejs:

-Create:

```
const criarProduto = () => {
  const produtoRef = doc(
    firebaseBD,
    'Microempreendedores',
    meiLogado.getEmail(),
    'Produtos',
    novoProduto.getNome()
  );

  const novoProduto = {
    descricao: novoProduto.getDescricao(),
    nome: novoProduto.getNome(),
    preco: novoProduto.getPreco(),
    quantidade: novoProduto.getQuantidade(),
    tags: novoProduto.getTags(),
  };

  addDoc(produtoRef, novoProduto)
    .then(() => {
      console.log('Novo produto criado com sucesso!');
    })
    .catch((error) => {
      console.log('Erro ao inserir a nova mensagem', error);
    });
};
```

-Read:

```
const getProdutos = async () => {
  const collectionRef = collection(db, "Microempreendedores",
    `${meiLogado.getEmail()}`, "Produtos");
  const produtos = []
  try {
    const collectionSnapshot = await getDocs(collectionRef);
    const collectionData = collectionSnapshot.docs.map((doc) =>
      produtos.push(doc.data()));
  }
```

```

    } catch (error) {
      console.log("Erro ao recuperar os produtos:", error);
    } finally {
      setIsLoading(false);
    }
  }
}

```

-Update:

```

const novaDescricao = ''
const novoNome = ''
const novoPreco = ''
const novaQuantidade = ''
const novasTags = []

const updateProduto = () => {
  updateDoc(doc(db, "Microempreendedores",
`${meiLogado.getEmail()}`, "Produtos",
`${produtoSelecionado.getNome()}`), {
    descricao: novaDescricao,
    nome: novoNome,
    preco: novoPreco,
    quantidade: novaQuantidade,
    tags: novasTags
  })
  .then(() => {
    console.log('Novo documento criado com sucesso!');
  })
  .catch((erro) => {
    console.error('Erro ao criar novo documento:', erro);
  });
}

```

-Delete:

```

const excluirDocumento = async() {
  const db = getFirestore();
  const docRef = doc(db, "Microempreendedores",
`${meiLogado.getEmail()}`, "Produtos",
`${produtoSelecionado.getNome()}`);

  try {
    await deleteDoc(docRef);
    console.log("Documento excluído com sucesso");
  }
}

```

```
    } catch (error) {  
      console.error("Erro ao excluir documento:", error);  
    }  
  }  
}
```