

MG

# Apresentação final MyGift

30 DE NOVEMBRO DE 2020



# EQUIPE MYGIFT

**Marcos Ani Cury V. Silva - Líder da equipe, Back-and e apresentação**

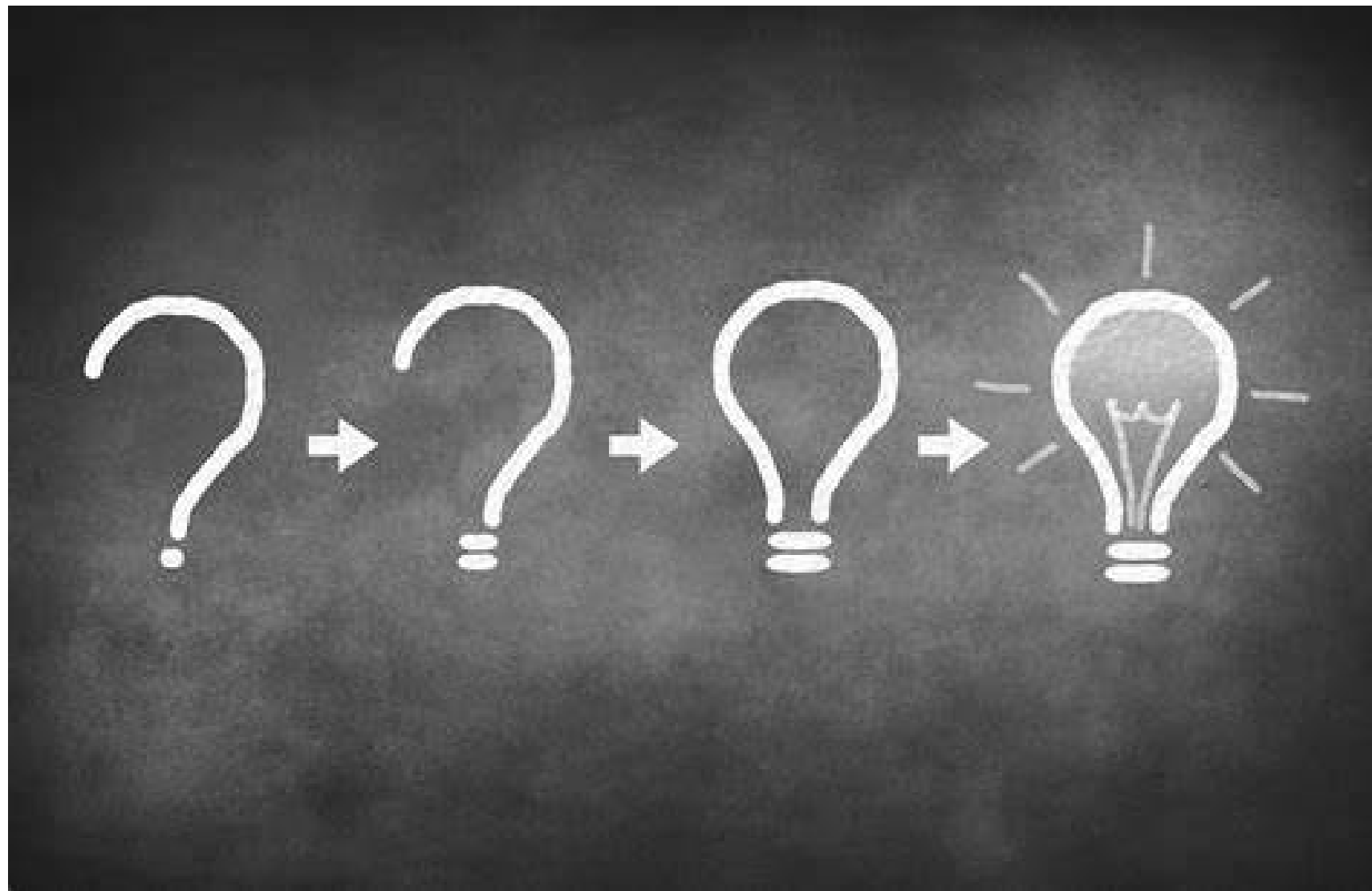
**Letícia A. Lucas - Responsável pelo Front-and**

**Victor L. de Andrade - Responsável pelo Banco de Dados e Chatbot**

**Ludmila B. S. Nascimento - Responsável pela documentação do projeto e Back-and**

# DEFINIÇÃO DO PROBLEMA

Dificuldade da escolha de um presente para presentear alguém

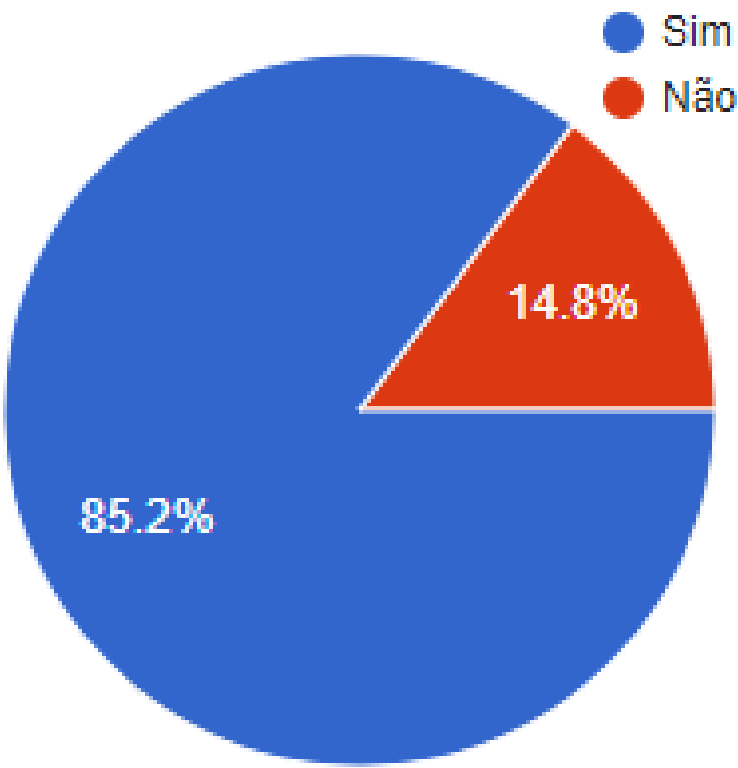



O costume de dar presentes em datas comemorativas é muito valorizado e quando não ocorre é visto com maus olhos.

Dificuldade de agradar um indivíduo ao presenteá-lo, com um longo tempo procurando o presente ideal

# PESQUISA DE MERCADO

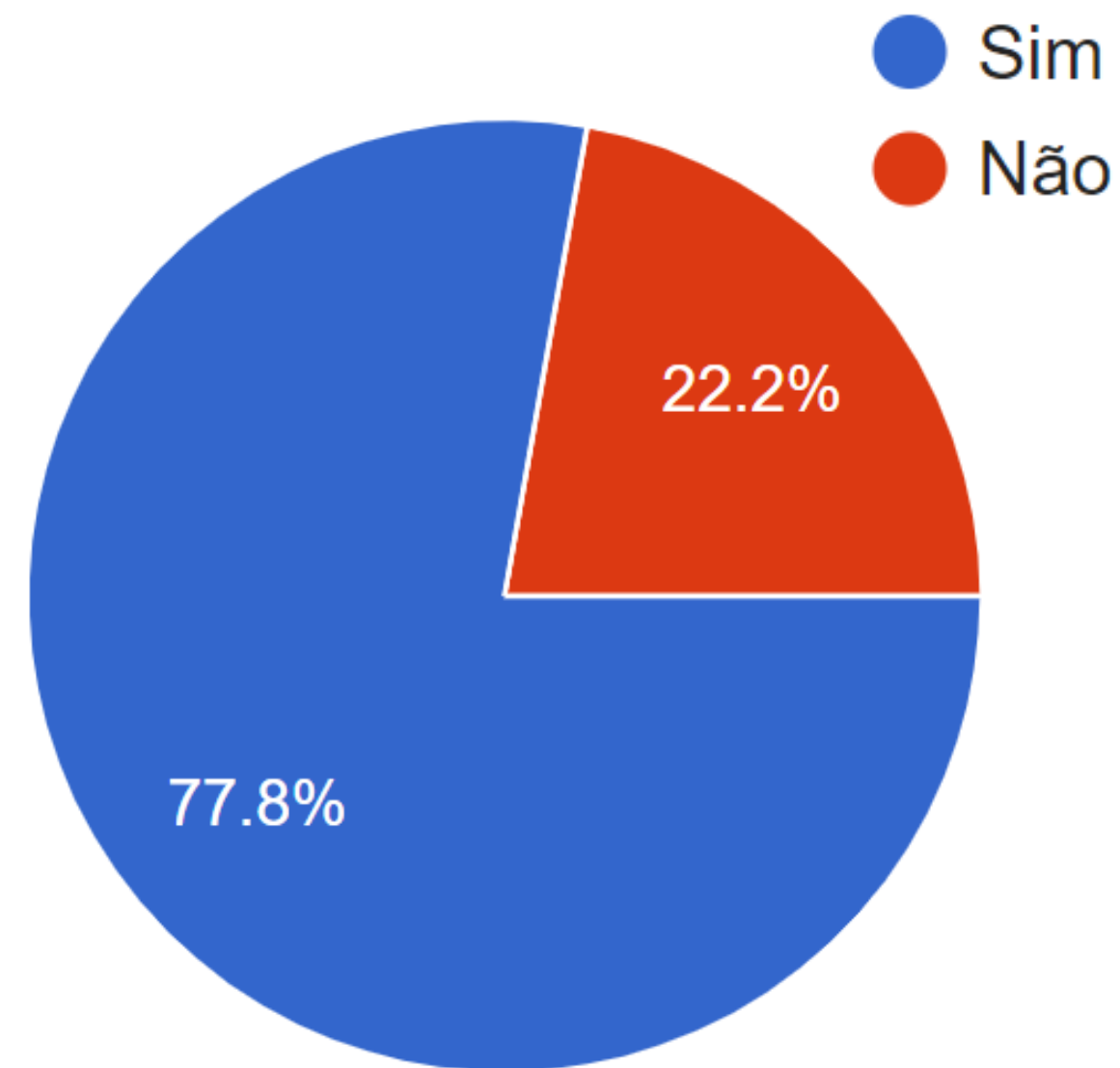
Você costuma presentear outras pessoas em datas comemorativas?



	Olga Camargos	
	Idade: 40 Ocupação: Dentista	Aplicativos: <ul style="list-style-type: none"><li>WhatsApp</li><li>Facebook</li><li>Instagram</li></ul>
Motivações: <ul style="list-style-type: none"><li>Família</li><li>Consultório</li></ul>	Frustrações: <ul style="list-style-type: none"><li>Comprar um produto e vir quebrado e/ou danificado</li><li>Segurança da entrega do produto</li><li>Dificuldades de comprar presentes para família</li></ul>	Hobbies, História: <ul style="list-style-type: none"><li>Brincar com os filhos</li><li>Ler</li><li>Pintura</li></ul>

# PESQUISA DE MERCADO

Você usaria um site que escolhe o melhor presente para presentear alguém?



# OBJETIVOS

## Objetivo Geral

Apresentações são ferramentas de comunicação que podem ser usadas como demonstrações, palestras, discursos, relatórios e mais.

## Objetivo Específico

1. Um portal que ajuda o usuário na escolha dos presentes.
2. Utilizar um chatbot para ajudar o usuário.
3. Uma IA que consiga acertar com uma certa precisão com base no perfil de uma pessoa

# PÚBLICO ALVO

Mulheres entre 20 e 45 anos

Embora todas as pessoas que buscam comprar um presente sejam adequadas para a aplicação



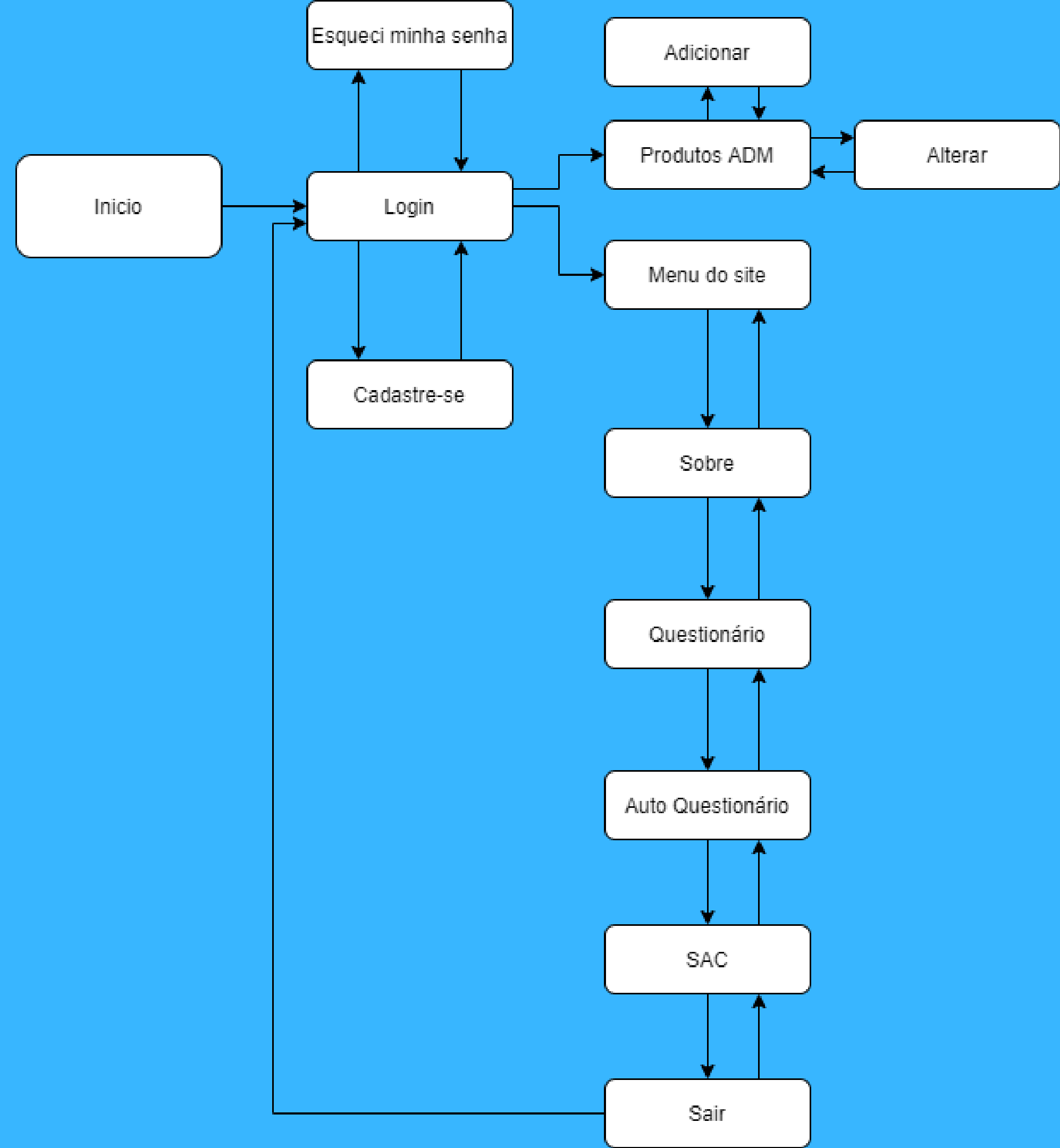
# PROJETO DE INTERFACE

## Características do Fluxograma

1. Agilidade

2. Acessibilidade

3. Usabilidade





# TECNOLOGIAS



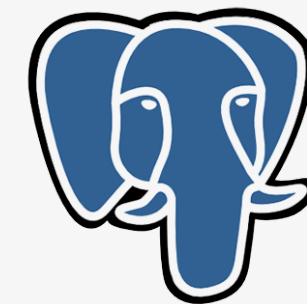
Back-and

Parte responsável pela  
funcionalidade da aplicação



Fron-end

Parte responsável pelo designer  
da aplicação



PostgreSQL

Banco de Dados

Parte responsável pelo  
armazenamento dos dados  
gerados e utilizados pela  
aplicação

# TECNOLOGIAS



## Hospedagem de Código

GitHub é uma plataforma de hospedagem de código-fonte e arquivos



## Organização do projeto

Trello é uma ferramenta de colaboração que organiza seus projetos em quadros.

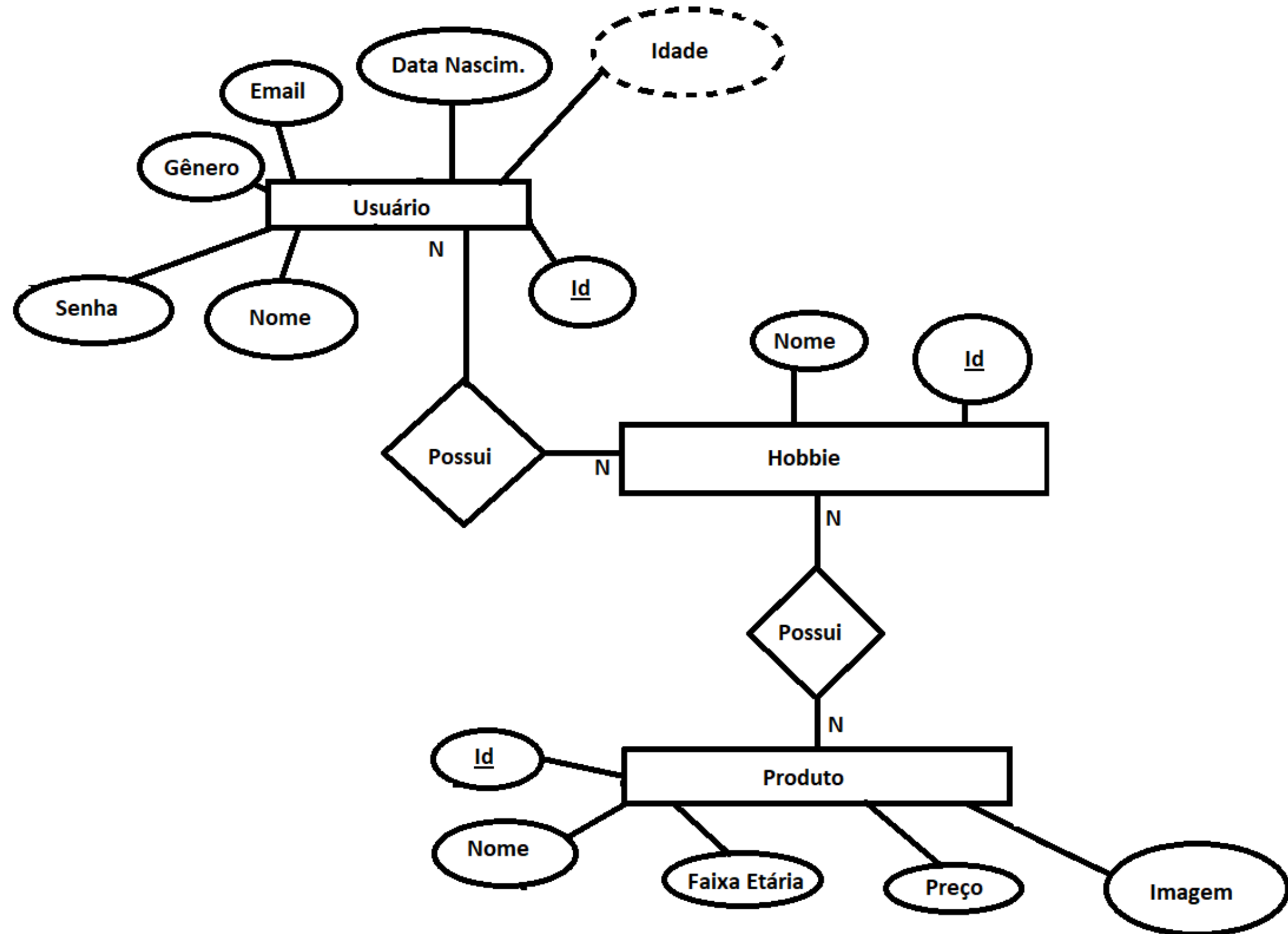


## Compilação

Apache Maven, ou Maven, é uma ferramenta de automação de compilação utilizada primariamente em projetos Java.

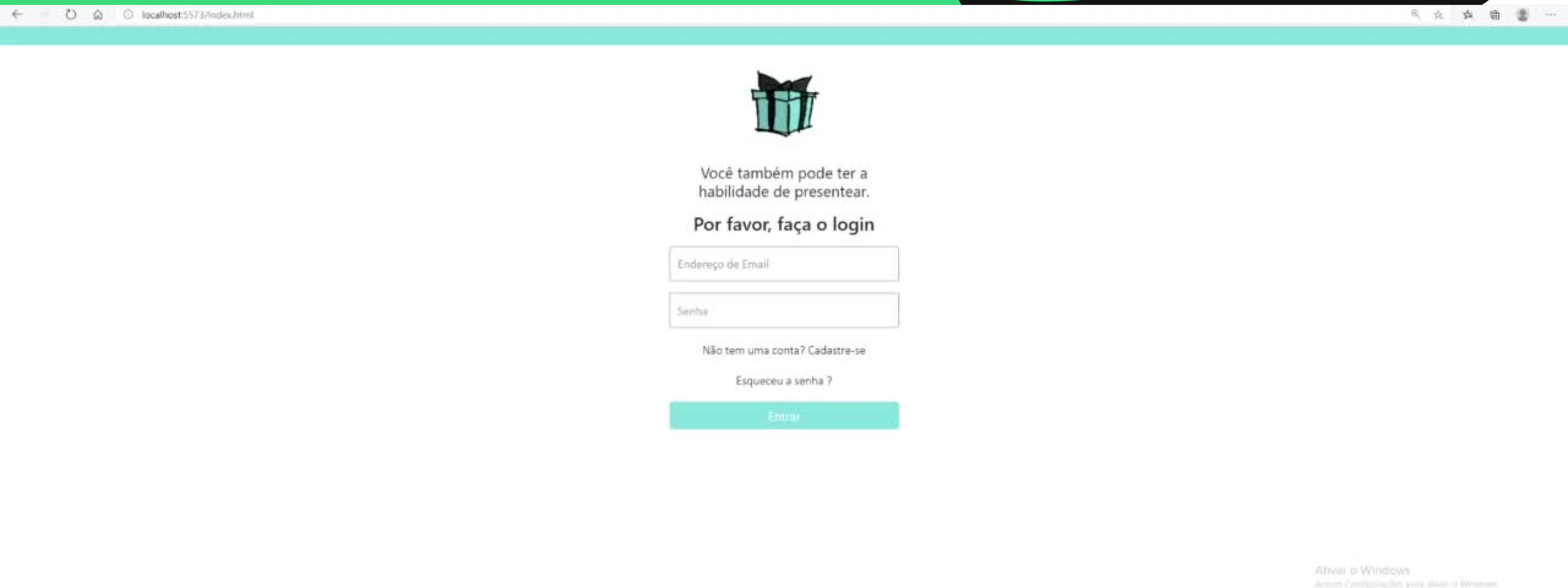
# BANCO DE DADOS

Etapa Conceitual (DER)



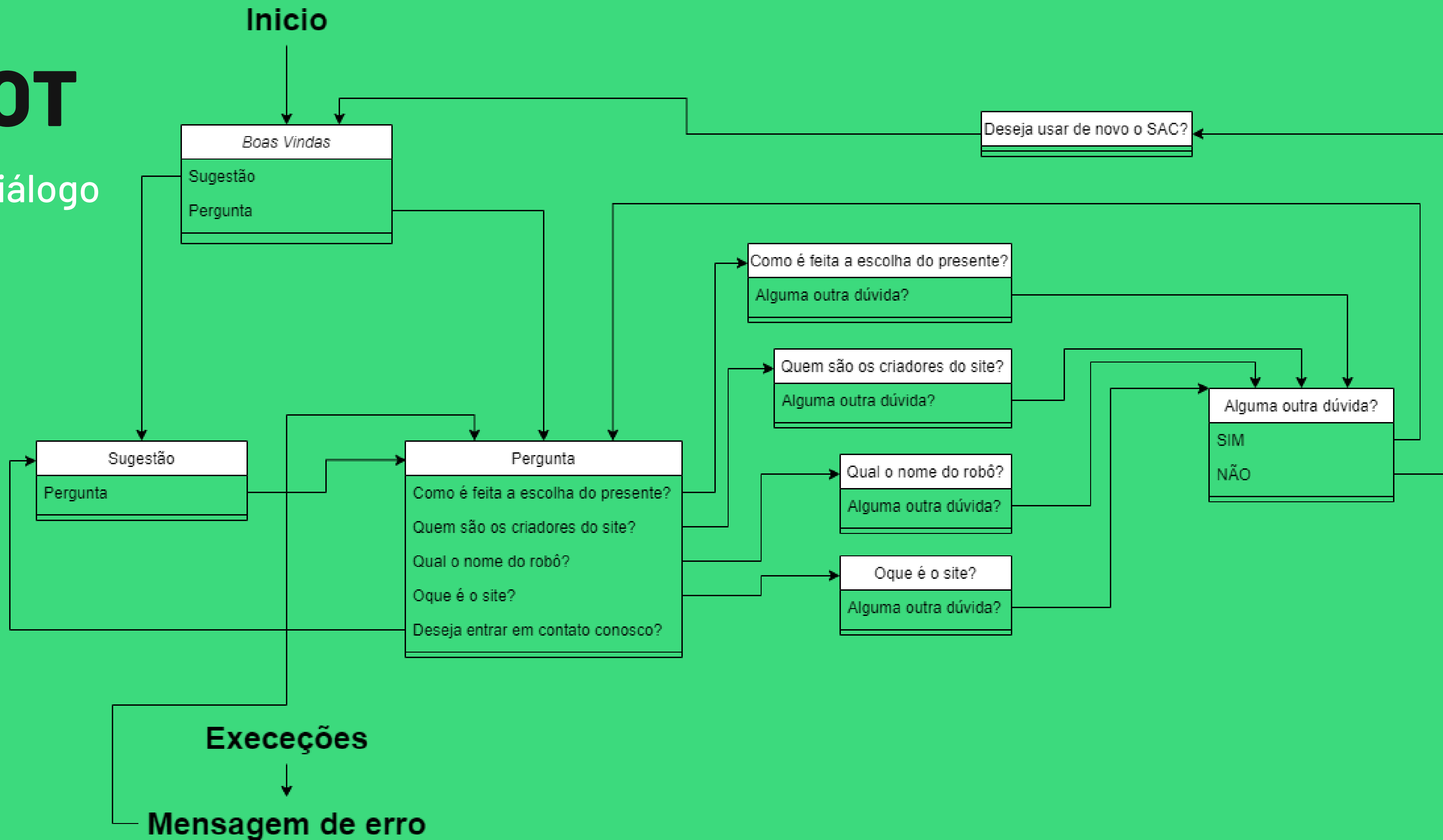
# Um pouco do nosso portal

Nosso portal foi feito com o intuito de ser simples, intuitivo e que cumpra seu propósito



# CHATBOT

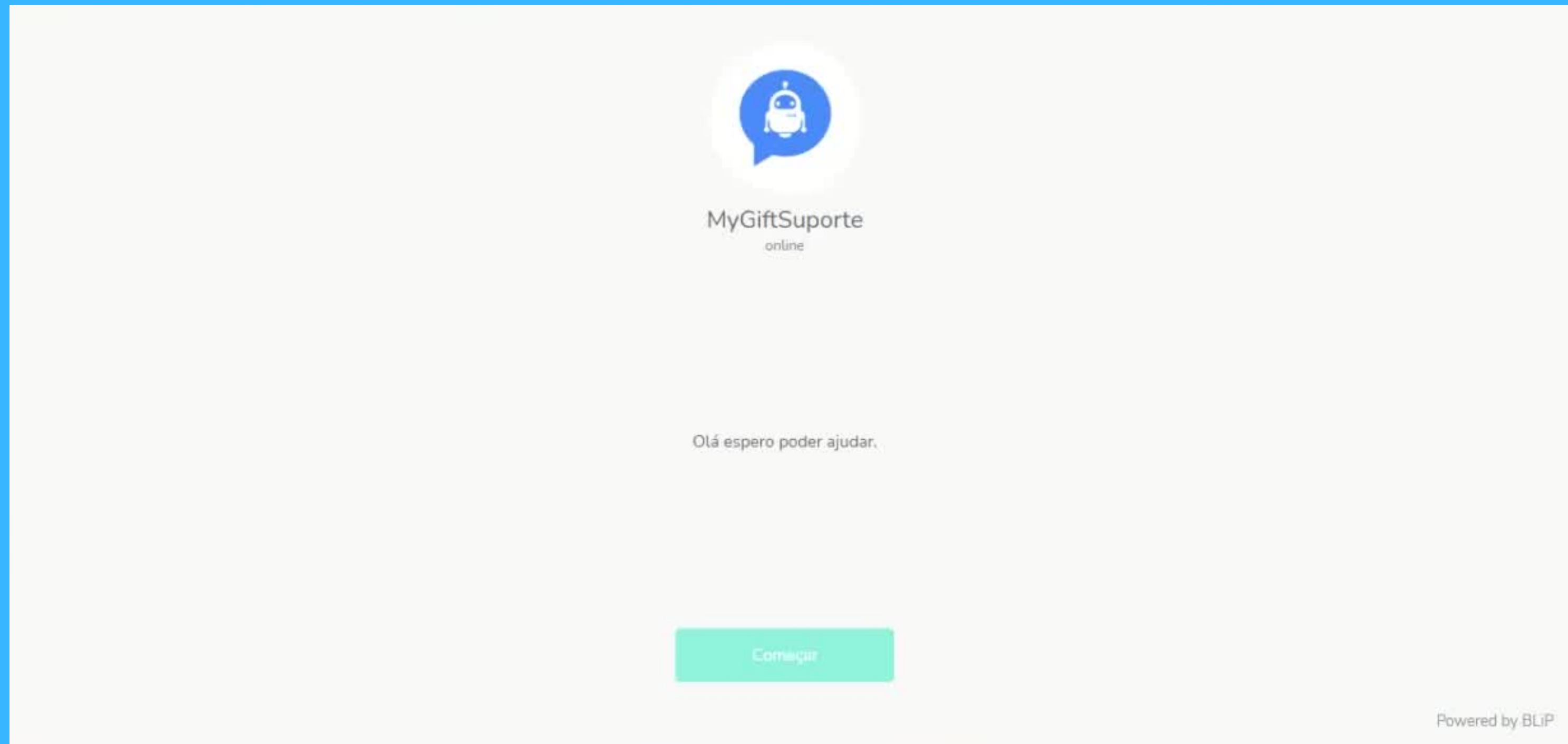
## Fluxograma de diálogo



# CHATBOT

## Um pouco sobre o Chatbot

Nosso chatbot foi feito para ser o SAC do nosso portal, dessa forma ele interage com o cliente para agilizar o atendimento sem a necessidade de um humano



# CÓDIGO

## Algoritmo de recomendação

Algoritmo criado para recomendar a um usuário um produto ideal com base na sua idade, seus hobbies e o valor que ele deseja pagar

```
//pegar produto por produto que existe entre os hobbies
Statement st = conexao.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
ResultSet rs = st.executeQuery("SELECT produto FROM hobbies_produto WHERE hobbie = "+idhobbies[i++]);
while(rs.next())
{
    //verificar faixa etaria do produto e se o valor e compativel com o do usuario
    boolean isProprio = false;
    boolean isPreco = false;
    Statement st2 = conexao.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
    ResultSet rs2 = st2.executeQuery("SELECT etaria,preco FROM produto WHERE id = "+ rs.getInt("produto"));
    if(rs2.next())
    {
        if(rs2.getInt("etaria") <= idade)
            isProprio = true;
        switch (valor)
        {
            case '-':
                if (rs2.getDouble("preco") < 50)
                    isPreco = true;
                break;
            case '1':
                if (rs2.getDouble("preco") < 150)
                    isPreco = true;
                break;
            case '+': isPreco = true;
                break;
            default:
        }
    }
}
st2.close();
if (isProprio && isPreco)
{
    //verificar todos os hobbies que cada produto possui e ver se possui algum que o hobbie do user tem
    st2 = conexao.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
    rs2 = st2.executeQuery("SELECT hobbie FROM hobbies_produto WHERE produto = "+ rs.getInt("produto"));
    int tmpconthobbies = 0;
    while(rs2.next())
    {
        for(int j = 0;j < idhobbies.length;j++)
        {
            if(rs2.getInt("hobbie") == idhobbies[j])
            {
                tmpconthobbies++;
                //forçar parada do for caso ache o hobbie no idhobbies
                j = idhobbies.length;
            }
        }
    }
    st2.close();
    //verificar se o produto possui um numero maior de hobbies compatíveis com o user do que o armazenado atual
    if (tmpconthobbies >= contMelhorProduto)
    {
        contMelhorProduto = tmpconthobbies;
        idMelhorProduto = rs.getInt("produto");
    }
    //verificar se o produto tem o mesmo numero de hobbies que o user
    if (contMelhorProduto == idhobbies.length - 1)
        i = idhobbies.length;
    st.close();
}
}
```

# CÓDIGO

## Requisições

Formas do seu front-and pedir/passar informações para o seu back-and processar e retornar um valor esperado

```
dao.conectar();
port(5573);
staticFiles.location("/public");

//--Usuario--

post("/formulario", (request, response) -> Servicos.formulario(request, response));

post("/autoformulario", (request, response) -> Servicos.autoformulario(request, response));

post("/login", (request, response) -> Servicos.login(request, response));

post("/cadastro", (request, response) -> Servicos.cadastro(request,response));

post("/esquecisenha", (request, response) -> Servicos.esqueciSenha(request,response));

//pegar 1 produto pelo id
get("/getproduto/:id", (request, response) -> Servicos.getProduto(request,response));

//resposta do formulario
get("/presenteado/:id", (request, response) -> Servicos.acharProduto(request, response));

//--Admin--

get("/listarprodutos", (request, response) -> Servicos.listarProdutos(request,response));

post("/inserirproduto", (request, response) -> Servicos.inserirProduto(request,response));

post("/deletarproduto", (request, response) -> Servicos.deletarProduto(request,response));

post("/alterarproduto", (request, response) -> Servicos.alterarProduto(request,response));

dao.close();
```



# OBRIGADO POR PARTICIPAR!

1/17 - Abertura do slide

2/17 - Apresentação dos integrantes

3/17 - Problemas

4/17 - Pesquisa de mercado Part.1

5/17 - Pesquisa de mercado Part.2

6/17 - Objetivos

7/17 - Público Alvo

8/17 - Projeto de Interface

9/17 - Tecnologias Part.1

10/17 - Tecnologias Part.2

11/17 - Banco de Dados

12/17 - Portal

13/17 - Chatbot Part.1

14/17 - Chatbot Part.2

15/17 - Código Part.1

16/17 - Código Part.2

17/17 - Final