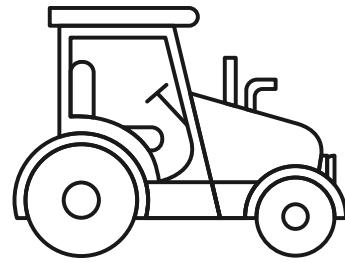
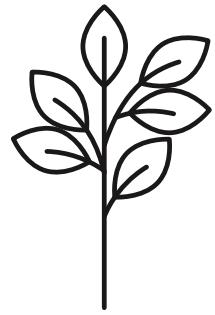


CPA

# Agronegócio no Brasil

Uso de Inteligência Artificial com chatbot do Agronegócio

# Problemas a Resolver



## Sustentabilidade

Extinção da imprevisibilidade de safra, resultando na diminuição de desperdício e aumento da sustentabilidade.

## Produtividade

Utilização da inteligência artificial para sanar quaisquer dúvidas em relação a agricultura, assim o fazendeiro ficará menos suscetível a erros.

## Responsabilidade

Diminuição do uso de agrotóxicos visando a diminuição da degradação do solo,

# Proposta de Solução

Uso de Inteligência Artificial com  
ChatBot voltado para o agronegócio

Nossos objetivos com a solução ao utilizar a IA generativa com uso de um Chat GPT especializado em agronegócio são facilitar e inovar soluções para os fazendeiros, visando a diminuição de erros, tratativa de dúvidas, previsibilidade de safra, aumento de produção, diminuição do desperdício e cultivo adequado do solo para evitar degradação.

# Análise da concorrência

Entendendo nosso cenário competitivo

## Companhia Nacional de Abastecimento

Acompanhamento mensal de safra de acordo com o clima para grãos.

## CHATGPT

Oferece um chat capaz de responder dúvidas sobre todos os assuntos.

## Agrofy

Realizam apenas gestão de safra, disponibilizando bases em tempo real de informações contábeis e operacionais.



# Modelos de Receita

## Como nosso projeto irá gerar lucro?

### Aumento de Eficiência Agrícola

Com a diminuição dos erros, o agricultor terá mais eficiência com o cultivo, assim poderá evitar a degradação do solo.

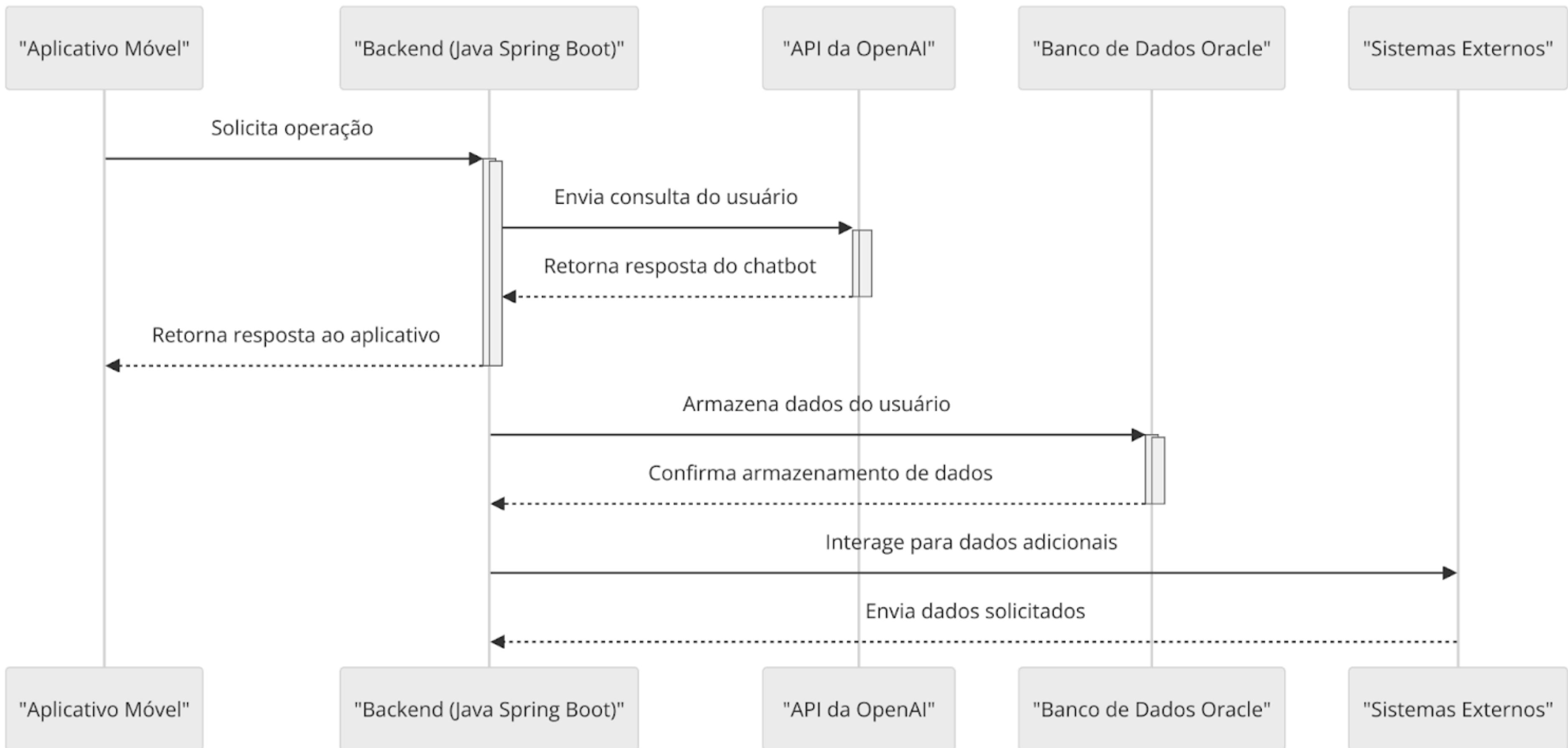
### Redução de Perdas

Ao antecipar possíveis problemas, como pragas, doenças ou condições climáticas adversas, os agricultores podem tomar medidas preventivas para minimizar perdas, o que protege sua renda.

### Melhoria na Tomada de Decisão

Com dados e insights precisos, os agricultores podem tomar decisões mais informadas sobre quais culturas plantar, quando plantar, como gerenciar o solo e como colher, resultando em melhores resultados financeiros.

# ARQUITETURA DA SOLUÇÃO E TECNOLOGIAS UTILIZADAS



# BANCO DE DADOS UTILIZADO NO CRUD - MYSQL

```
1 •    create database dbagro
2      default character set utf8mb4
3      default collate utf8mb4_general_ci;
4
5 •    use dbagro;
6
7 •    create table if not exists agricultura (
8          id int not null auto_increment primary key,
9          cultivo varchar(100) not null,
10         estacaoAno varchar(100) not null,
11         fertilizante varchar(100) not null,
12         clima varchar(100) not null
13     )default char set utf8mb4;
```

# BANCO DE DADOS UTILIZADO NO CRUD - MYSQL

5 • `SELECT * FROM dbagro.agricultura;`

Result Grid | Filter Rows:  | Edit:

	<code>id</code>	<code>cultivo</code>	<code>estacaoAno</code>	<code>fertilizante</code>	<code>dima</code>
▶	1	Trigo	Verao	Aadubo	Clima
	2	Trigo	Verao	Aadubo	Clima
	3	Trigo	Verao	Aadubo	Ensolarado
	4	Tomate	Inverno	Aadubo	Chuvoso
	NULL	NULL	NULL	NULL	NULL

# CONEXÃO COM O BANCO DE DADOS MYSQL

```
✓ using System;
  using System.Collections.Generic;
  using System.Linq;
  using System.Text;
  using System.Threading.Tasks;

✓ namespace SprintAgro
{
  2 referências
  static class ConexaoBanco
  {

    private const string servidor = "Local instance MySQL80";
    private const string bancoDados = "dbagro";
    private const string usuario = "root";
    private const string senha = "nathinha";

    static public string bancoServidor = $"server={servidor}; user id={usuario};database={bancoDados};password={senha}";

  }
}
```

# CRUD - MÉTODO CADASTRAR

```
private void btnCadastrar_Click(object sender, EventArgs e)
{
    try
    {
        if (!txtCultivo.Text.Equals("") && !txtClima.Text.Equals("") && !txtEstacaoAno.Text.Equals("") && !txtFertilizante.Text.Equals(""))
        {
            CadastroAgricultura cadAgro = new CadastroAgricultura();
            cadAgro.Cultivo = txtCultivo.Text;
            cadAgro.Clima = txtClima.Text;
            cadAgro.EstacaoAno = txtEstacaoAno.Text;
            cadAgro.Fertilizante = txtFertilizante.Text;

            if (cadAgro.cadastrarAgro())
            {
                MessageBox.Show($"A agricultura {cadAgro.Cultivo} foi cadastrada com sucesso!");
                txtCultivo.Clear();
                txtClima.Clear();
                txtEstacaoAno.Clear();
                txtFertilizante.Clear();
                txtCultivo.Focus();
            }
            else
            {
                MessageBox.Show("Não foi possível cadastrar agricultura");
            }
        }
        else
        {
            MessageBox.Show("Favor preencher todos os campos corretamente");
            txtCultivo.Clear();
            txtClima.Clear();
            txtEstacaoAno.Clear();
            txtFertilizante.Clear();
            txtCultivo.Focus();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Erro ao cadastrar agricultura" + ex.Message);
    }
}
```

# CRUD - MÉTODO PESQUISAR

```
l referência  
private void btnPesquisar_Click(object sender, EventArgs e)  
{  
    try  
    {  
        if (txtCultivo.Text.Equals(""))  
        {  
            CadastroAgricultura cadAgri = new CadastroAgricultura();  
            cadAgri.Cultivo = txtCultivo.Text;  
  
            MySqlDataReader reader = cadAgri.localizarAgricultura();  
  
            if (reader != null)  
            {  
                if (reader.HasRows)  
                {  
                    reader.Read();  
  
                    lblId.Text = reader["id"].ToString();  
                    lblId.Text = reader["fertilizante"].ToString();  
                    lblId.Text = reader["estacaoAno"].ToString();  
                    lblId.Text = reader["clima"].ToString();  
                }  
                else  
                {  
                    MessageBox.Show("Cultivo não encontrado");  
                    txtCultivo.Clear();  
                    txtClima.Clear();  
                    txtEstacaoAno.Clear();  
                    txtFertilizante.Clear();  
                    txtCultivo.Focus();  
                    lblId.Text = "";  
                }  
            }  
        }  
    }  
}
```

```
private void btnPesquisar_Click(object sender, EventArgs e)  
{  
    try  
    {  
        if (txtCultivo.Text.Equals(""))  
        {  
            CadastroAgricultura cadAgri = new CadastroAgricultura();  
            cadAgri.Cultivo = txtCultivo.Text;  
  
            MySqlDataReader reader = cadAgri.localizarAgricultura();  
  
            if (reader != null)  
            {  
                if (reader.HasRows)  
                {  
                    reader.Read();  
  
                    lblId.Text = reader["id"].ToString();  
                    lblId.Text = reader["fertilizante"].ToString();  
                    lblId.Text = reader["estacaoAno"].ToString();  
                    lblId.Text = reader["clima"].ToString();  
                }  
                else  
                {  
                    MessageBox.Show("Cultivo não encontrado");  
                    txtCultivo.Clear();  
                    txtClima.Clear();  
                    txtEstacaoAno.Clear();  
                    txtFertilizante.Clear();  
                    txtCultivo.Focus();  
                    lblId.Text = "";  
                }  
            }  
        }  
    }  
}
```

# CRUD - MÉTODO ATUALIZAR

```
private void btnAtualizar_Click(object sender, EventArgs e)
{
    try
    {
        if (!txtCultivo.Equals("") && !txtClima.Equals("") && !txtEstacaoAno.Equals("") && !txtFertilizante.Equals(""))
        {
            CadastroAgricultura cadastroAgricultura = new CadastroAgricultura();
            cadastroAgricultura.Id = int.Parse(lblId.Text);
            cadastroAgricultura.Clima = txtClima.Text;
            cadastroAgricultura.Cultivo = txtCultivo.Text;

            if(cadastroAgricultura.atualizarAgricultura())
            {
                MessageBox.Show("Os dados foram atualizados com sucesso");
                txtCultivo.Clear();
                txtClima.Clear();
                txtEstacaoAno.Clear();
                txtFertilizante.Clear();
                txtCultivo.Focus();
            }
            else
            {
                MessageBox.Show("Nao foi possivel atualizar os dados");
                txtCultivo.Clear();
                txtClima.Clear();
                txtEstacaoAno.Clear();
                txtFertilizante.Clear();
                txtCultivo.Focus();
            }
        }
    }
}
```

# CRUD - MÉTODO DELETAR

```
private void btnDeletar_Click(object sender, EventArgs e)
{
    try
    {
        if (!txtCultivo.Equals("") && !txtClima.Equals("") && !txtEstacaoAno.Equals("") && !txtFertilizante.Equals(""))
        {
            CadastroAgricultura cadastroAgricultura = new CadastroAgricultura();
            cadastroAgricultura.Id = int.Parse(lblId.Text);

            if (cadastroAgricultura.deleteAgricultura())
            {
                MessageBox.Show("A agricultura foi excluido com sucesso");
            }
            else
            {
                MessageBox.Show("Nao foi possivel excluir funcionario");
                txtCultivo.Clear();
                txtClima.Clear();
                txtEstacaoAno.Clear();
                txtFertilizante.Clear();
            }
        }
        else
        {
            MessageBox.Show("Favor pesquisar qual agricultura deseja excluir");
        }
    }
    catch (Exception ex)
```

# CRUD - FORMULÁRIO

Cultivo id:

Cultivo:

Trigo

EstacaoAno:

Verao

Fertilizante:

Adubo

Clima:

Ensolarado

[Cadastrar](#)

[Pesquisar](#)

[Atualizar](#)

[Deletar](#)