

# Collegare

- Sprint 03

**Trabalho Interdisciplinar II:  
Banco de Dados e Sistemas Inteligentes**

**Bárbara Luciano  
Frederico Malaquias  
Larissa Valadares  
Ricardo Furiati**

# Tópicos

01

## Back-end

- Conexão com Front-end
- CRUDs

02

## Banco de Dados

- DDL
- Scripts
- Conexão com back-end

03

## Sistemas Inteligentes

- Intelligent System Canvas
- Ferramental
- Entradas e Saídas
- Fornecedor

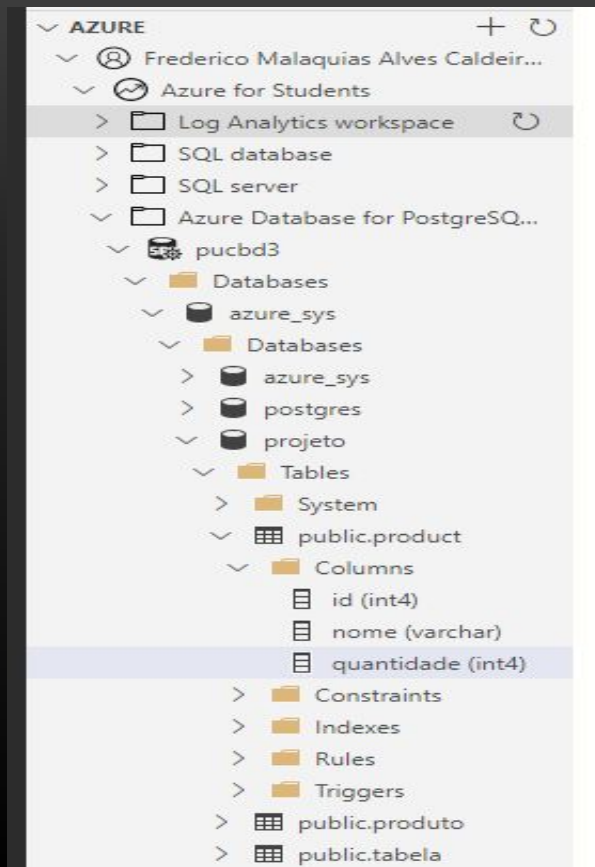
02

**01**

## Back-end

- Conexão com Front-end
- CRUDs

# Conexão com o Front-end



# CRUDs

```
function incluirCadastro() {  
  
  let objDados = leDados();  
  
  let strNome = document.getElementById("inputNome").value;  
  let strEmpresa = document.getElementById("inputEmpresa").value;  
  let strCategoria = document.getElementById("inputCategoria").value;  
  let strEndereco = document.getElementById("inputEndereco").value;  
  let strTelefone = document.getElementById("inputTelefone").value;  
  let strEmail = document.getElementById("inputEmail").value;  
  let strDescricao = document.getElementById("inputDescricao").value;
```

```
function validar(){  
  localStorage.setItem("usuario", document.querySelector("input#user").value);  
  localStorage.setItem("nome", document.querySelector("input#name").value);  
  localStorage.setItem("sobrenome", document.querySelector("input#lastName").value);  
  localStorage.setItem("email", document.querySelector("input#email").value);  
  localStorage.setItem("telefone", document.querySelector("input#telephone").value);  
  localStorage.setItem("endereco", document.querySelector("input#address").value);  
  localStorage.setItem("cidade", document.querySelector("input#city").value);  
  localStorage.setItem("estado", document.querySelector("input#state").value);  
  
  var user = document.getElementById("user");  
  var name = document.getElementById("name");  
  var lastName = document.getElementById("lastName");  
  var email = document.getElementById("email");  
  var telephone = document.getElementById("telephone");  
  var address = document.getElementById("address");  
  var city = document.getElementById("city");  
  var state = document.getElementById("state");
```

Cadastre-se conosco! ×

## Novo usuário

Usuário

Nome completo

E-mail

Senha

Confirmação de senha

Cancelar

Salvar

## Collegare

[Voltar ao menu principal](#)

[Alterar senha](#)

## Editar perfil

Usuário (\*)

Nome (\*)

Sobrenome (\*)

Email (\*)

Telefone para contato (\*)

(DDD)xxxxx-xxxx

Endereço (\*)

Cidade (\*)

Estado (\*)

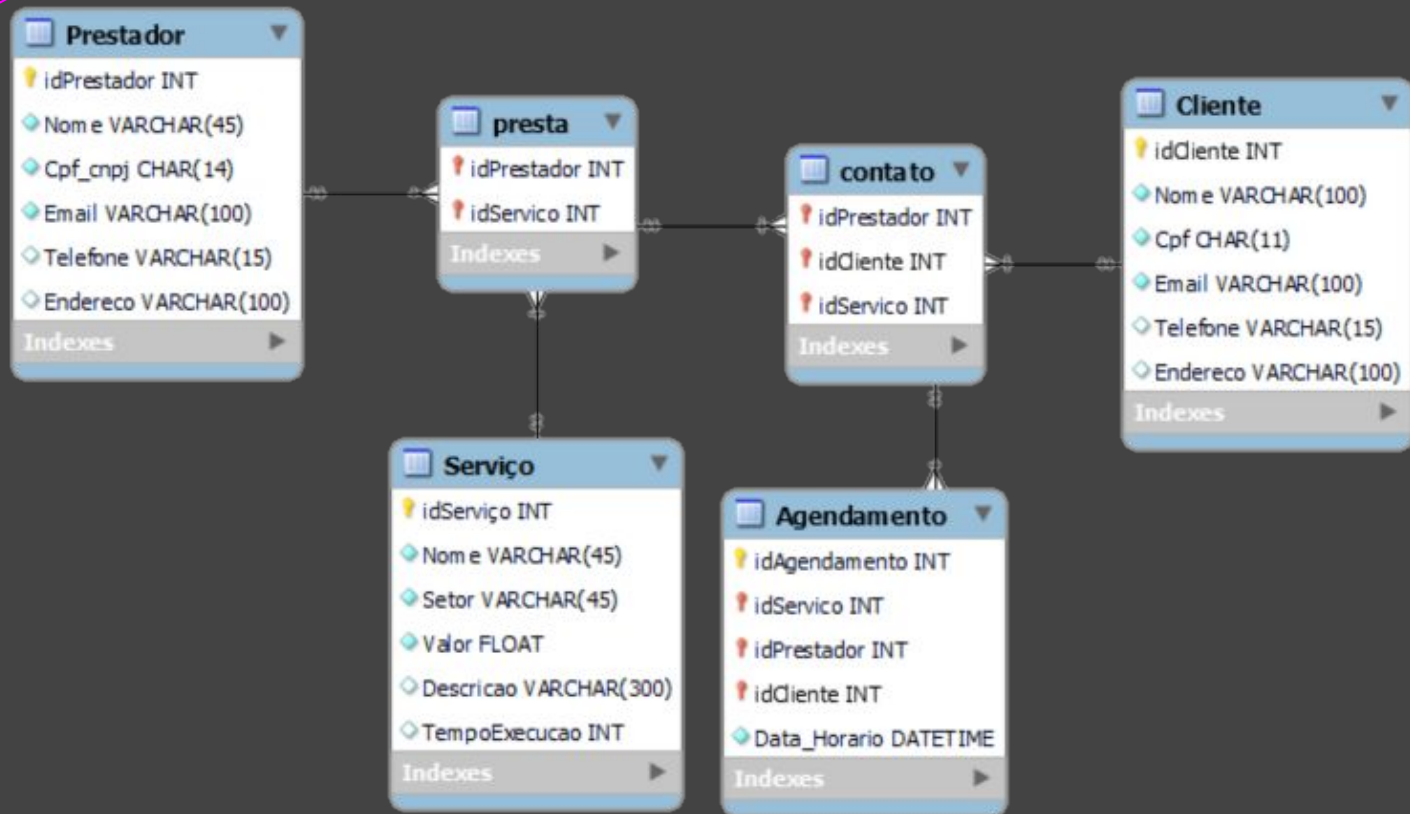
Concluir

## 02

## Banco de Dados

- DDL
- Scripts
- Conexão com back-end

# DDL



# Scripts

C: > Users > fredri > Documentos > PUCMinas > TI2 > AzureBD > TABLEA2.pgsq

```
1  -- CREATE TABLE SERVICE (  
2  --     SERVICIO serial PRIMARY KEY,  
3  --     VALOR VARCHAR(50),  
4  --     DESCRICAO INTEGER  
5  -- );  
6  
7  -- INSERT INTO SERVICE (SERVICIO, VALOR, DESCRICAO) VALUES(1, '', 150);  
8  -- INSERT INTO SERVICE (SERVICIO, VALOR, DESCRICAO) VALUES(2, '', 154);  
9  
10 select * from SERVICE
```

2 rows returned

	servico integer	valor character varying	descricao integer
1	1		150
2	2		154

C: > Users > fredri > Documentos > PUCMinas > TI2 > AzureBD > TABELA\_USUARIO.pgsq

```
1  CREATE TABLE USERS(  
2  |    USUARIO serial PRIMARY KEY,  
3  |    NOME VARCHAR(50),  
4  |    CPF INTEGER  
5  | );  
6  
7  INSERT INTO USERS (USUARIO, NOME, CPF) VALUES(1, '', 150);  
8  INSERT INTO USERS (USUARIO, NOME, CPF) VALUES(2, '', 154);  
9  
10 select * from USERS
```

null rows created

1 row inserted

1 row inserted

2 rows returned

	usuario integer	nome character varying	cpf integer
1	1		150
2	2		154



# Conexão com Back-end

```
package model;

import java.text.SimpleDateFormat;
import java.util.Date;

public class Servico {

    private int idServico;
    private String nome;
    private String setor;
    private float valor;
    private String descricao;
    private Date tempoExecucao;

    public Servico() {
        idServico = 0;
        nome = "";
        setor = "";
        valor = 0;
        descricao = "";
    }

    public Servico(int idServico,
String nome, String setor, float valor,
String descricao, Date tempoExecucao) {
        setIdServico(idServico);
        setNome(nome);
        setSetor(setor);
        setValor(valor);
        setDescricao(descricao);
        setTempoExecucao(tempoExecucao);
    }
}
```

```
public int getIdServico() {
    return idServico;
}

public void setIdServico(int idServico) {
    this.idServico = idServico;
}
```

```
@Override
public String toString() {

    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

    return "Id: " + idServico + " Nome: " + nome + " Setor: " +
        setor + " Valor: " + valor + " Descricao: " + descricao +
        " Tempo de Execucao: " + sdf.format(tempoExecucao);
}




@Override
public boolean equals(Object obj) {
    return (this.getIdServico() == ((Servico) obj).getIdServico());
}
```

## 03

## Sistemas Inteligentes

- Intelligent System Canvas
- Ferramental
- Entradas e Saídas
- Fornecedor

# Intelligent System Canvas

Ferramental de IA	Entradas	Proposição de valor	Equipe	Clientes
<ul style="list-style-type: none"> <li>- Um agente inteligente de software para sanar dúvidas específicas (frequentes) de usuários quanto a utilização da plataforma</li> <li>- Um algoritmo de associação e previsão de série temporal para recomendar a melhor solução</li> </ul> 	<ul style="list-style-type: none"> <li>- Perguntas pré-selecionadas pelos criadores do Collegare que possuem alta possibilidade de serem feitas pelos usuários do sistema</li> </ul>	<ul style="list-style-type: none"> <li>- Modernização do sistema, a fim de atrair potenciais usuários e colaboradores</li> <li>- Aproximação real entre sistema e usuário a partir da interação pelo chatbot</li> </ul> 	<ul style="list-style-type: none"> <li>- Desenvolvedor Web</li> <li>- Cientista de Dados (IA)</li> <li>- Analista/Projetista de Dados (BD)</li> </ul>	<ul style="list-style-type: none"> <li>- Prestadores de serviços diversos que queiram divulgá-los em uma plataforma própria para isso</li> <li>- Usuários que precisam adquirir um determinado serviço e que pesquisam online</li> </ul> 
<b>Custos</b> <ul style="list-style-type: none"> <li>- Equipamentos (computadores)</li> <li>- Custo de armazenamento de dados em nuvem (Microsoft Azure)</li> </ul>			<b>Receitas</b> <ul style="list-style-type: none"> <li>- Opção de destaque (paga) para Prestadores</li> <li>- Recomendação dos melhores serviços e descontos (opção premium) para Usuários</li> </ul>	



## Ferramental de Inteligência Artificial

AI Toolkit Conceptual Framework

### Agentes Inteligentes

- Automação de Processos
- *Chatbots e Assistentes*
- Robôs
- Simulação

### Machine Learning

- Previsão
- Classificação
- Associação
- Agrupamento
- Otimização orientada a dados

### Fala & Texto

- Análise de sentimentos
- Conversões

### Visão Computacional

- Identificação de elementos
- Análise de sentimentos

# Entradas e Saídas

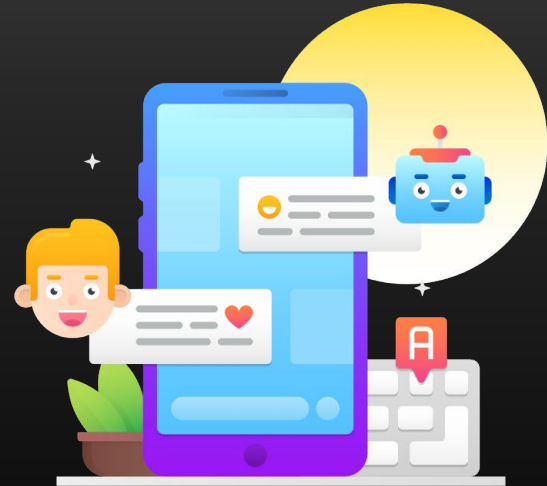
## ❑ Entrada:

-Pergunta inserida pelo usuário.

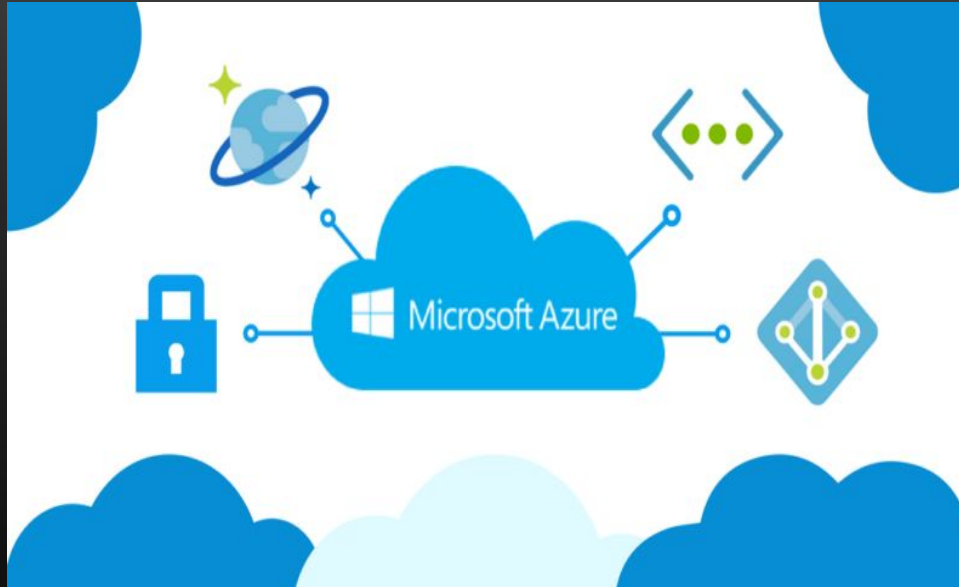
## ❑ Saída:

-Resposta do robô com base um banco de respostas.

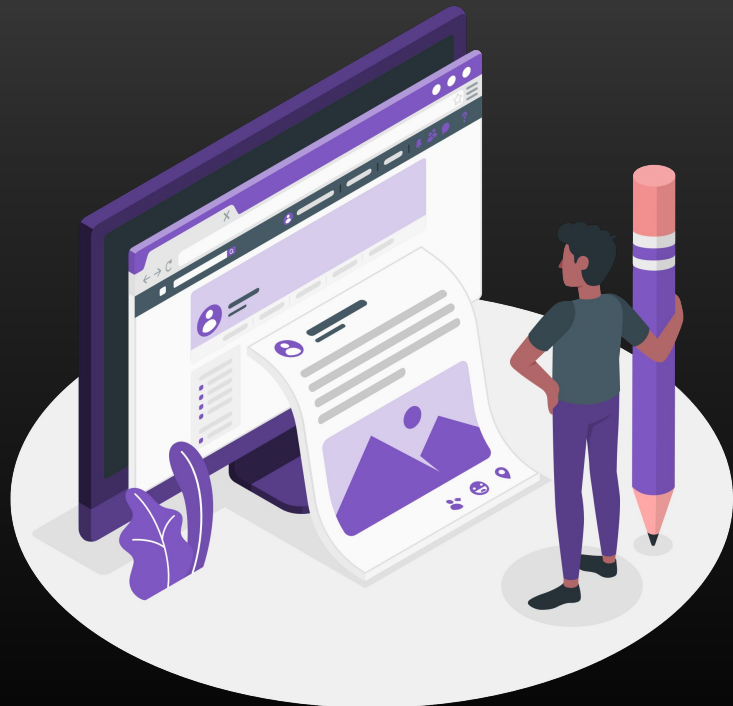
-Através da conexão de palavras-chaves da entrada do usuário.



# Fornecedor



# Collegare



<https://rickfuriati.github.io/Collegare/>



Obrigado :)

Alguma  
pergunta?

