

Collegare

- Sprint 02

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

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

Tópicos

01

Funções

- Quais são?
- Execução (1)
- Execução (2)

02

Banco de Dados

- Modelo Conceitual
- Modelo Relacional
- Modelo de Implementação

03

Scripts SQL

- Esquema / Prestador / Cliente
- Serviço / Presta
- Contato
- Agendamento

04

Implementação

- Front-end → Página principal
- Front-end → Login e Cadastro

02

01

Funções

- Quais são?
- Execução (1)
- Execução (2)

Funções do Sistema

-  **Mapa integrado**
-  **Cadastros e Login**
-  **Browser**

-  **Edição de perfil**
-  **Fale conosco e avaliação**



Execução

- ❏ **Mapa integrado → sugestão de endereço / API Google Maps e Places**
- ❏ **Login → testes de autocomplete e usuário e senha corretos**
- ❏ **Cadastro → formulário para usuário preencher**
→ salvar as informações do usuário no banco de dados
- ❏ **Cadastro de serviços → salvar ou procurar um dado no localStorage**

Execução



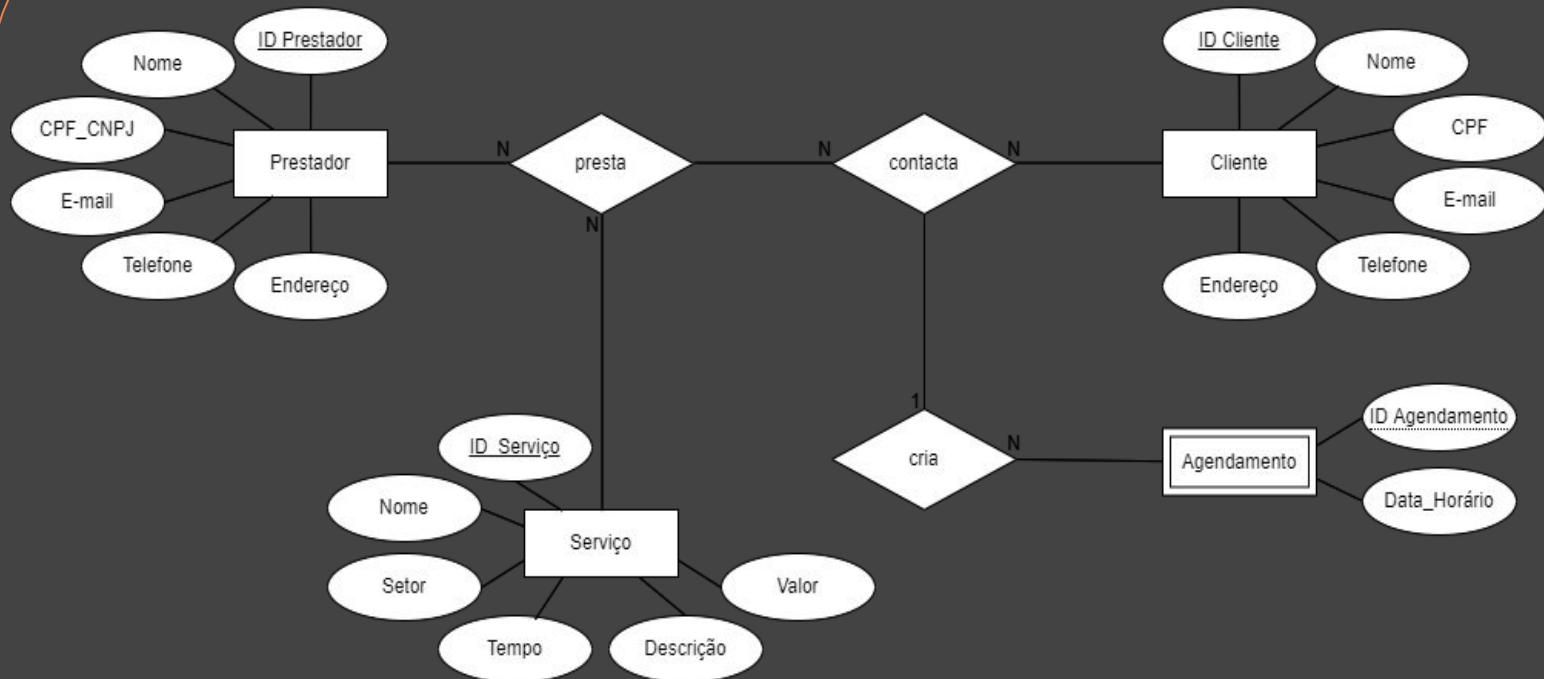
- ❏ **Edição de perfil** → permite ao usuário editar suas informações
- ❏ **Fale conosco** → envio de mensagens do usuário para suporte
- ❏ **Avaliação dos Prestadores** → armazenar a avaliação no localStorage
 - alerta de sucesso de operação
 - edição de avaliação
- ❏ **Browser** → pesquisa
 - sugestão de pesquisa

02

Banco de Dados

- Modelo Conceitual
- Modelo Relacional
- Modelo de Implementação

Modelo Conceitual



Modelo Relacional

PRESTADOR

<u>ID_Prestador</u>	Nome	CPF_CNPJ	E-mail	Telefone	Endereço
---------------------	------	----------	--------	----------	----------

SERVIÇO

<u>ID_Serviço</u>	Nome	Setor	Valor	Descrição	Tempo
-------------------	------	-------	-------	-----------	-------

PRESTA

<u>ID_Prestador</u>	<u>ID_Serviço</u>
---------------------	-------------------

CONTATO

<u>ID_Serviço</u>	<u>ID_Prestador</u>	<u>ID_Cliente</u>
-------------------	---------------------	-------------------

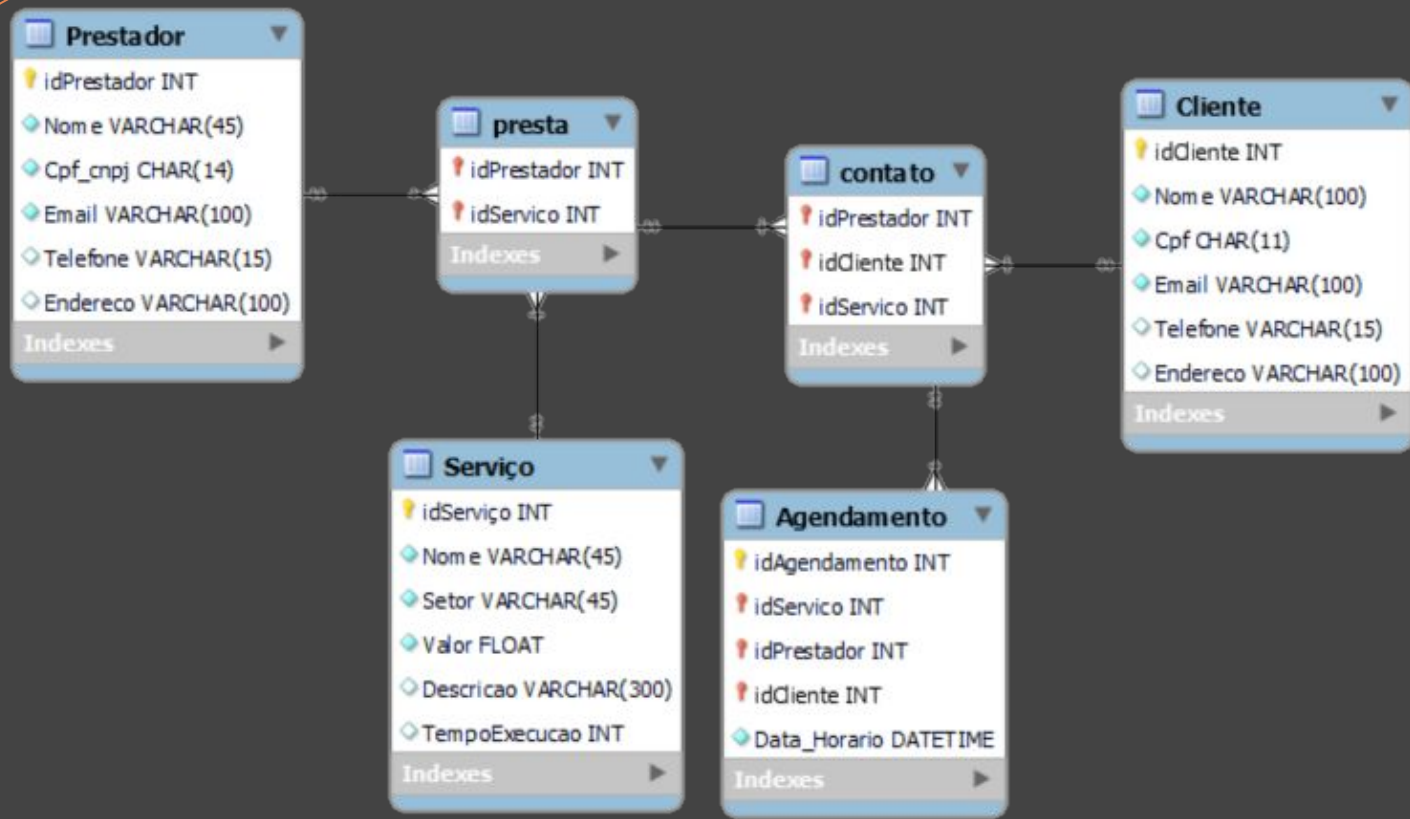
AGENDAMENTO

<u>ID_Agendamento</u>	<u>ID_Serviço</u>	<u>ID_Prestador</u>	<u>ID_Cliente</u>	Data_Horário
-----------------------	-------------------	---------------------	-------------------	--------------

CLIENTE

<u>ID_Cliente</u>	Nome	CPF	E-mail	Telefone	Endereço
-------------------	------	-----	--------	----------	----------

Modelo de Implementação



03

Scripts do SQL

- Esquema / Prestador / Cliente
- Serviço / Presta
- Contato
- Agendamento



Scripts do SQL

```
CREATE SCHEMA IF NOT EXISTS  
`mydb` DEFAULT CHARACTER SET  
utf8 ;  
USE `mydb` ;
```

```
CREATE TABLE IF NOT EXISTS  
`mydb`.`Prestador` (  
  `idPrestador` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Cpf_cnpj` CHAR(14) NOT NULL,  
  `Email` VARCHAR(100) NOT NULL,  
  `Telefone` VARCHAR(15) NULL,  
  `Endereco` VARCHAR(100) NULL,  
  PRIMARY KEY (`idPrestador`))  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS  
`mydb`.`Cliente` (  
  `idCliente` INT NOT NULL,  
  `Nome` VARCHAR(100) NOT NULL,  
  `Cpf` CHAR(11) NOT NULL,  
  `Email` VARCHAR(100) NOT NULL,  
  `Telefone` VARCHAR(15) NULL,  
  `Endereco` VARCHAR(100) NULL,  
  PRIMARY KEY (`idCliente`))  
ENGINE = InnoDB;
```



Scripts do SQL

```
CREATE TABLE IF NOT EXISTS
`mydb`.`Serviço` (
  `idServiço` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Setor` VARCHAR(45) NOT NULL,
  `Valor` FLOAT NOT NULL DEFAULT 0,
  `Descricao` VARCHAR(300) NULL,
  `TempoExecucao` INT NULL DEFAULT 0,
  PRIMARY KEY (`idServiço`))
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`presta` (
  `idPrestador` INT NOT NULL,
  `idServiço` INT NOT NULL,
  PRIMARY KEY (`idPrestador`, `idServiço`),
  CONSTRAINT `idPrestador`
    FOREIGN KEY (`idPrestador`)
      REFERENCES `mydb`.`Prestador` (`idPrestador`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `idServiço`
    FOREIGN KEY (`idServiço`)
      REFERENCES `mydb`.`Serviço` (`idServiço`)
    ON DELETE CASCADE ON UPDATE CASCADE)
ENGINE = InnoDB;
```



Scripts do SQL

```
CREATE TABLE IF NOT EXISTS `mydb`.`contato` (  
  `idPrestador` INT NOT NULL,  
  `idCliente` INT NOT NULL,  
  `idServico` INT NOT NULL,  
  PRIMARY KEY (`idPrestador`, `idCliente`, `idServico`),  
  CONSTRAINT `idPrestador`  
    FOREIGN KEY (`idPrestador`)  
      REFERENCES `mydb`.`presta` (`idPrestador`)  
      ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `idServico`  
    FOREIGN KEY (`idServico`)  
      REFERENCES `mydb`.`presta` (`idServico`)  
      ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `idCliente`  
    FOREIGN KEY (`idCliente`)  
      REFERENCES `mydb`.`Cliente` (`idCliente`)  
      ON DELETE CASCADE ON UPDATE CASCADE)  
ENGINE = InnoDB;
```



Scripts do SQL

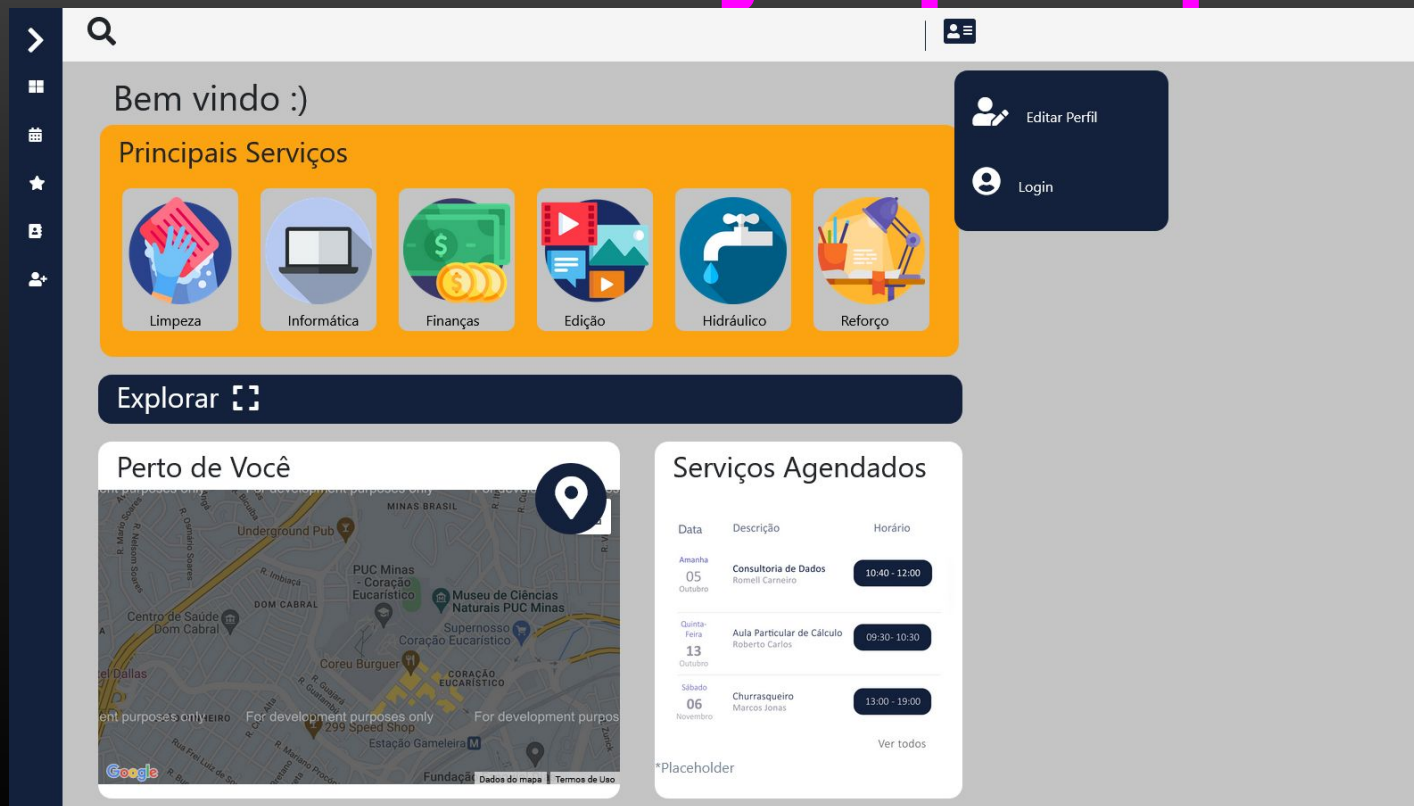
```
CREATE TABLE IF NOT EXISTS `mydb`.`Agendamento` (  
  `idAgendamento` INT NOT NULL,  
  `idServico` INT NOT NULL,  
  `idPrestador` INT NOT NULL,  
  `idCliente` INT NOT NULL,  
  `Data_Horario` DATETIME NOT NULL,  
  PRIMARY KEY (`idAgendamento`, `idServico`, `idPrestador`, `idCliente`),  
  CONSTRAINT `idServico`  
    FOREIGN KEY (`idServico`)  
    REFERENCES `mydb`.`contato` (`idServico`)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `idPrestador`  
    FOREIGN KEY (`idPrestador`)  
    REFERENCES `mydb`.`contato` (`idPrestador`)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `idCliente`  
    FOREIGN KEY (`idCliente`)  
    REFERENCES `mydb`.`contato` (`idCliente`)  
    ON DELETE NO CASCADE ON UPDATE CASCADE )  
ENGINE = InnoDB;
```

04


Implementação

- Front-end → Página principal
- Front-end → Login e Cadastro

Front-end → Página principal



Front-end → Login e Cadastro



Usuário

Senha

Cadastre-se

Login

Cadastre-se conosco! ×

Novo usuário

Usuário

Nome completo

E-mail

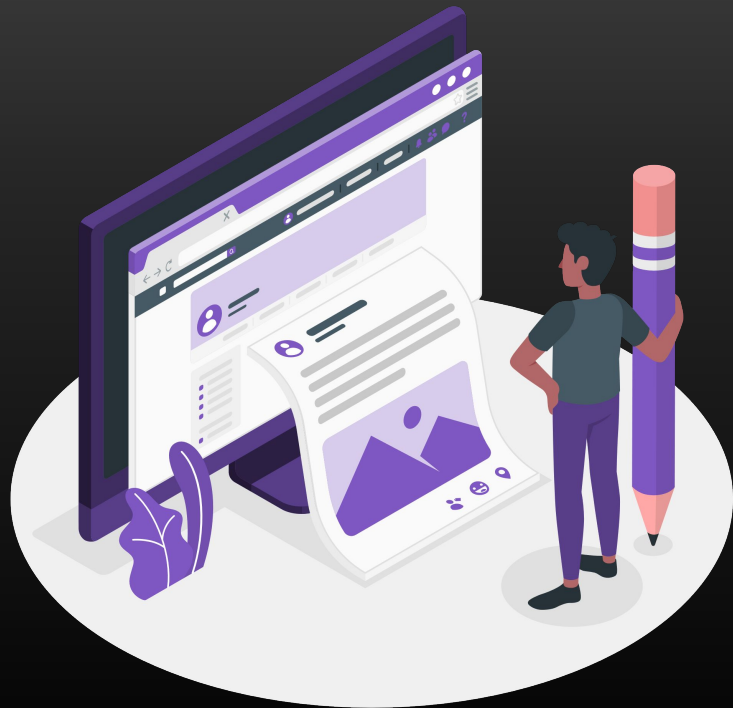
Senha

Confirmação de senha

Cancelar

Salvar

Collegare



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



Obrigado :)

Alguma
pergunta?

