

Projeto ***FinManage***





01

Banco de Dados

02

Front-End

03

Back-End

04

Sistemas Inteligentes



01

Banco de Dados

02

Front-End

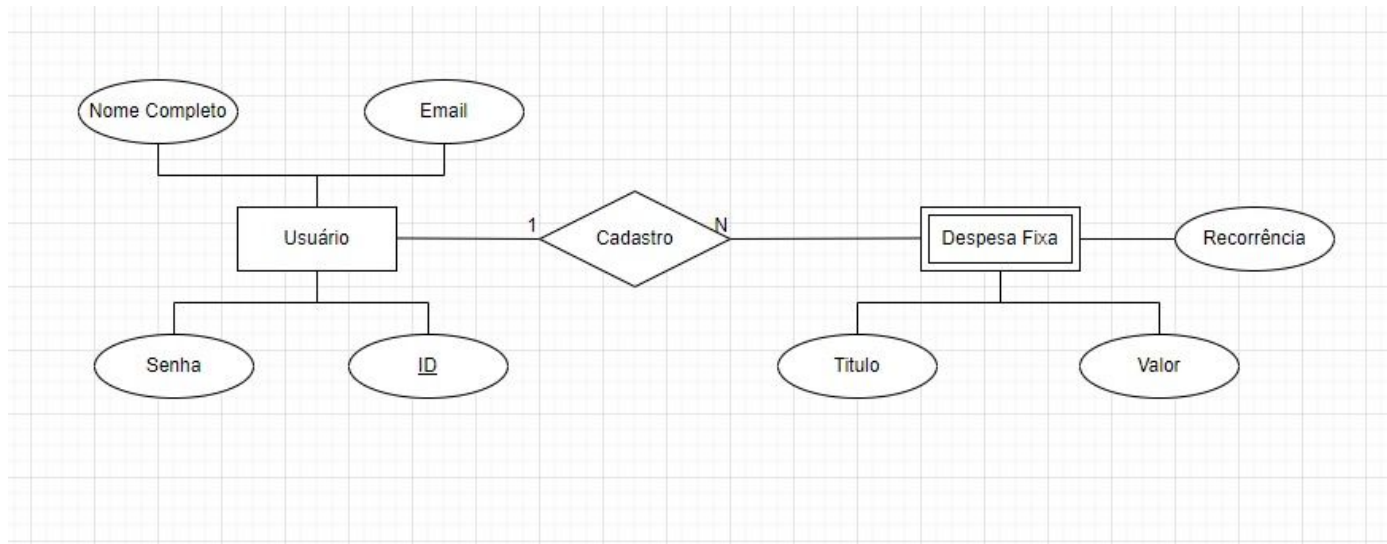
03

Back-End

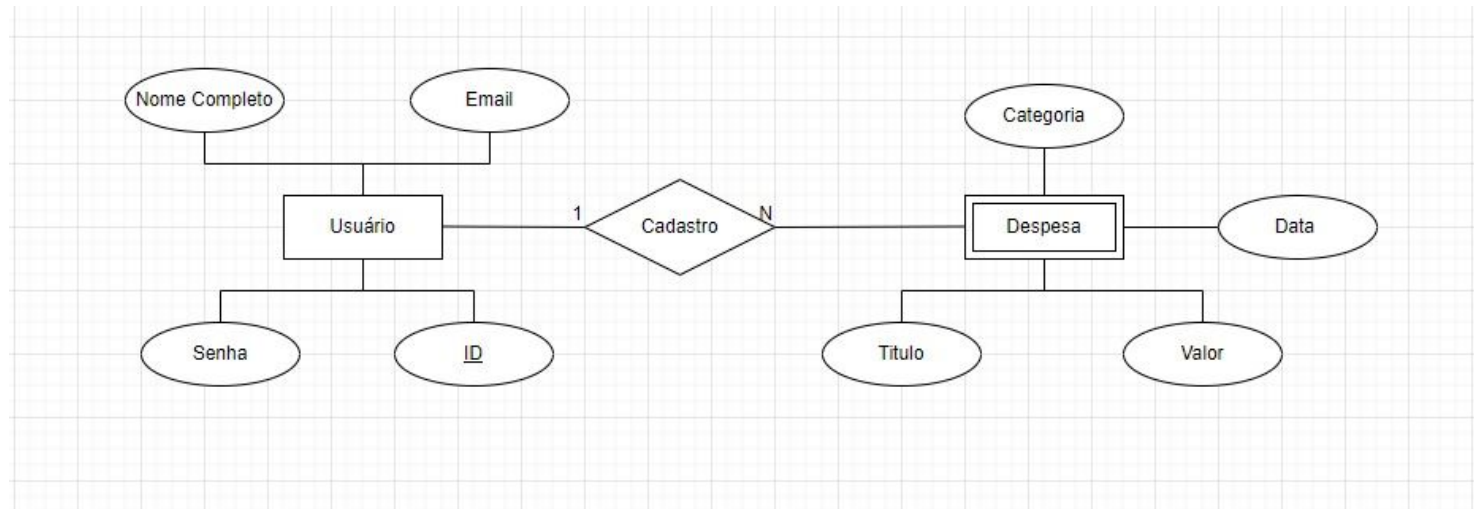
04

Sistemas Inteligentes

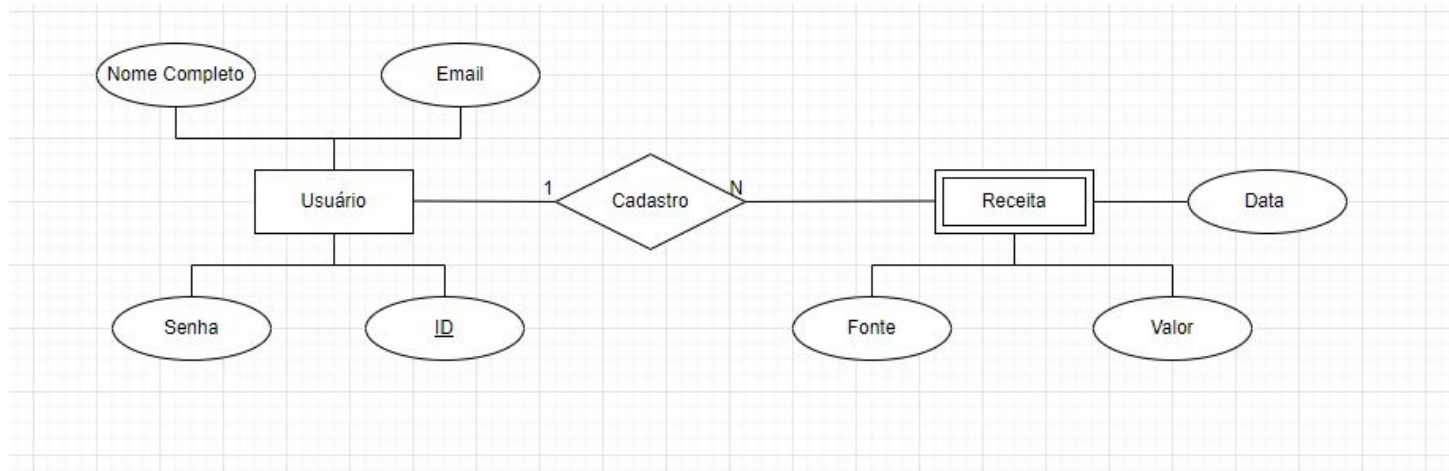
Banco de Dados



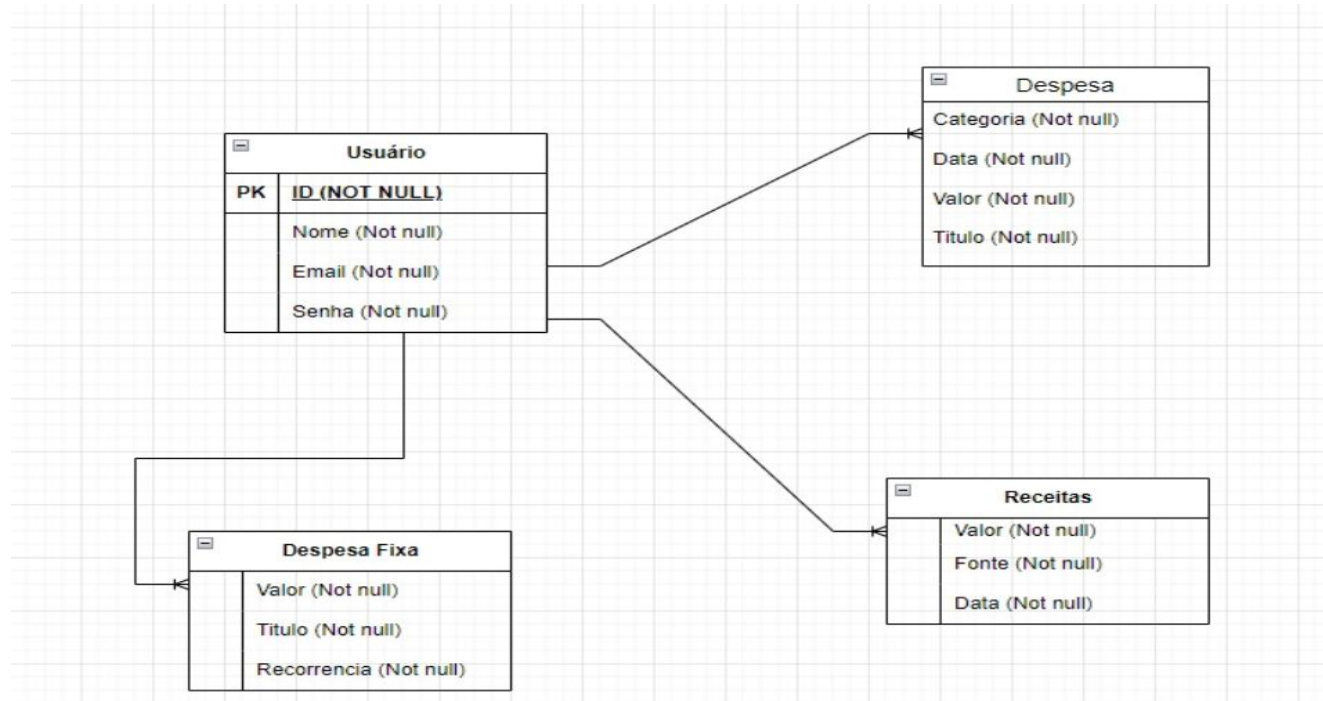
Banco de Dados



Banco de Dados



Banco de Dados





01

Banco de Dados

02

Front-End

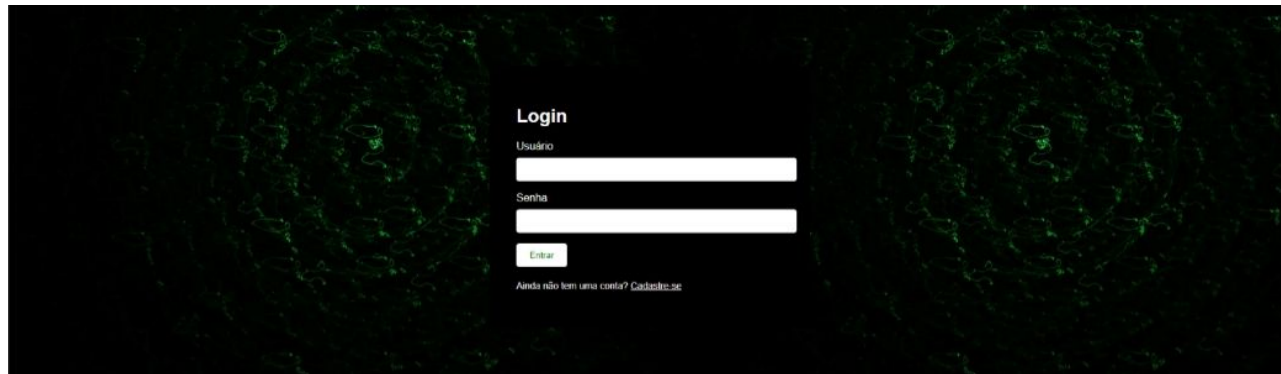
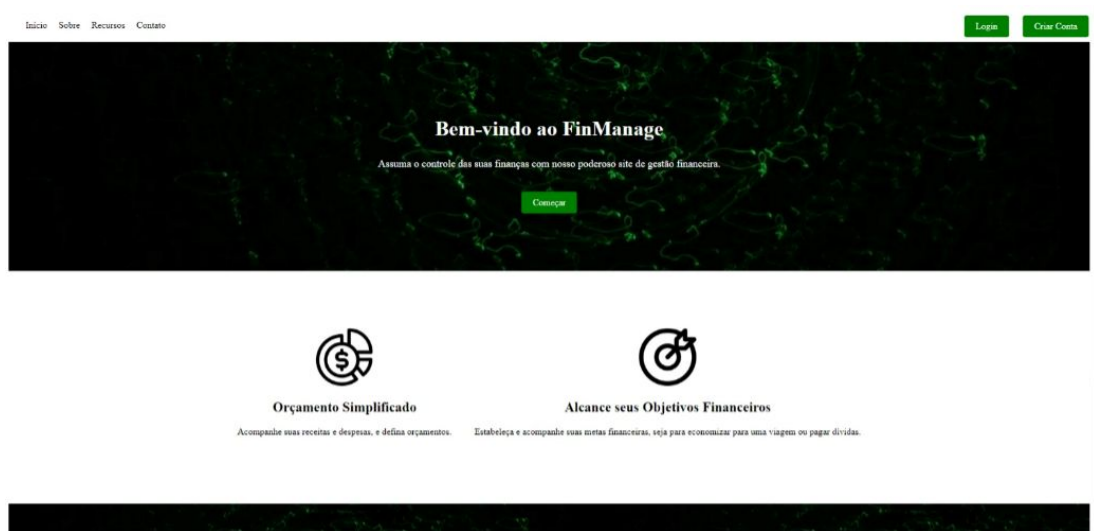
03

Back-End

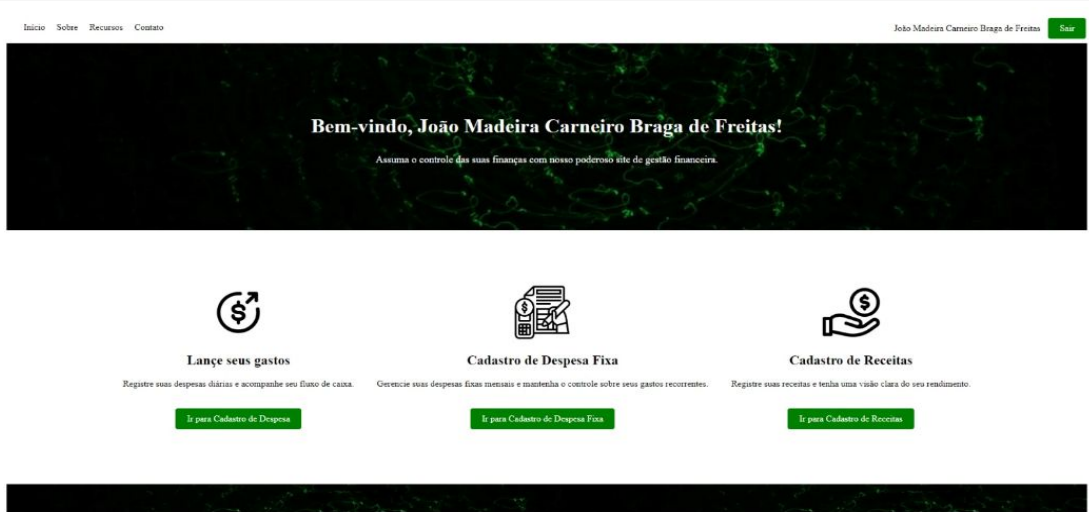
04

Sistemas Inteligentes

Front-End



Front-End



Lance seus gastos

Título:

Valor:

Data:
 ☐

Categoria:

Título	Valor	Data	Categoria	Editar	Excluir
uber	12.9	2023-08-26	transporte	<input type="button" value="Editar"/>	<input type="button" value="Excluir"/>

Lance seus gastos

Título:

Valor:

Data:
 ☐

Categoria:

Título	Valor	Data	Categoria	Editar	Excluir
--------	-------	------	-----------	--------	---------

Front-End

```
1 document.getElementById("formConta").addEventListener("submit", function (event) {
2     event.preventDefault();
3
4     var nome = document.getElementById("nome").value;
5     var email = document.getElementById("email").value;
6     var senha = document.getElementById("senha").value;
7
8     var conta = {
9         nomeCompleto: nome,
10        email: email,
11        senha: senha
12    };
13
14    axios.post("http://localhost:6789/usuario/insert", conta)
15        .then((response) => {
16            console.log(response.data);
17        })
18        .catch((error) => {
19            console.error("Erro ao enviar requisição:", error);
20        });
21
22    // Exibir a mensagem de conta criada com sucesso
23    exibirMensagemContaCriada();
24
25 });
26
27
28 function exibirMensagemContaCriada() {
29     // Obtém o elemento que exibirá a mensagem
30     var mensagemContaCriada = document.getElementById("mensagemContaCriada");
31
32     // Verifica se o elemento existe
33     if (mensagemContaCriada) {
34         // Altera o estilo do elemento para que seja exibido
35         mensagemContaCriada.style.display = "block";
36
37         // Aguarda dois segundos antes de redirecionar para a página de sucesso
38         setTimeout(function () {
39             window.location.href = "conta-criada-sucesso.html";
40         }, 2000); // 2000 milissegundos = 2 segundos
41     } else {
42         console.error("Elemento mensagemContaCriada não encontrado!");
43     }
44 }
45
```

```
<html>
<head>
  <title>Criação de Conta</title>
  <link rel="stylesheet" type="text/css" href="css/criacao-conta.css">
</head>
<body>
  <div class="container">
    <h1>Criação de Conta</h1>
    <form id="formConta">
      <label for="nome">Nome:</label>
      <input type="text" id="nome" required>
      <label for="email">E-mail:</label>
      <input type="email" id="email" required>
      <label for="senha">Senha:</label>
      <input type="password" id="senha" required>
      <button type="submit">Criar Conta</button>
      <button type="button" id="btnCancelar">Cancelar</button>
    </form>
    <div id="mensagemContaCriada" style="display: none;">
      Conta criada com sucesso!
    </div>
    <ul id="listaContas"></ul>
  </div>
  <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
  <script src="scripts/criacao-conta.js"></script>
</body>
</html>
```



01

Banco de Dados

02

Front-End

03

Back-End

04

Sistemas Inteligentes

Back-End

```
Aplicacao.java X
41     }
42
43     return "OK";
44 });
45
46 //USUARIOS
47 post("/usuario/insert", (request, response) -> usuarioService.insert(request, response));
48
49 get("/usuario/get", (request, response) -> usuarioService.get(request, response));
50
51 get("/usuario/getAll", (request, response) -> usuarioService.getAll(request, response));
52
53 //DESPESA
54 post("/despesa/insert", (request, response) -> despesaService.insert(request, response));
55
56 get("/despesa/get/:id", (request, response) -> despesaService.get(request, response));
57
58 put("/despesa/update/:id", (request, response) -> despesaService.update(request, response));
59
60 delete("/despesa/delete/:id", (request, response) -> despesaService.delete(request, response));
61
62 //DESPESA FIXA
63 post("/despesa-fix/insert", (request, response) -> despesaFixService.insert(request, response));
64
65 get("/despesa-fix/get/:id", (request, response) -> despesaFixService.get(request, response));
66
67 put("/despesa-fix/update/:id", (request, response) -> despesaFixService.update(request, response));
68
69 delete("/despesa-fix/delete/:id", (request, response) -> despesaFixService.delete(request, response));
70
71 //RECEITA
72
73 post("/receita/insert", (request, response) -> receitaService.insert(request, response));
74
75 get("/receita/get/:id", (request, response) -> receitaService.get(request, response));
76
77 put("/receita/update/:id", (request, response) -> receitaService.update(request, response));
78
79 delete("/receita/delete/:id", (request, response) -> receitaService.delete(request, response));
80
81 }
82 }
83
84 class CorsFilter {
```

Back-End

```
Aplicacao.java  *DespesaService.java X
11
12 public class DespesaService {
13
14     private DespesaDAO DespesaDAO = new DespesaDAO();
15
16     public DespesaService() {
17
18     }
19
20     public String insert(Request request, Response response) {
21         try{
22             String json = request.body();
23             Gson gson = new Gson();
24
25             Despesa despesa = gson.fromJson(json, Despesa.class);
26
27             if(DespesaDAO.insert(despesa) == true) {
28                 response.status(201);
29                 return "{\"message\": \"Despesa inserida com sucesso\"}";
30             } else {
31                 response.status(404);
32                 return "{\"message\": \"Despesa não foi inserida\"}";
33             }
34         } catch (NullPointerException e){
35             response.status(500);
36             return "{\"message\": \"Ocorreu um erro no servidor\"}";
37         }
38     }
39
40     public Object get(Request request, Response response) {
41         String id = request.params("id");
42
43         ArrayList<Despesa> despesas = DespesaDAO.get(Integer.parseInt(id));
44
45         if (despesas != null) {
46             response.header("Content-Type", "application/json");
47             response.header("Content-Encoding", "UTF-8");
48
49             return new Gson().toJson(despesas);
50         } else {
51             response.status(404);
52             return "{\"message\": \"Despesa não encontrada\"}";
53         }
54     }
}
```



01

Banco de Dados

02

Front-End

03

Back-End

04

Sistemas Inteligentes

Sistemas Inteligentes

ChatBot

Responder a perguntas
Fornecer dicas de finanças
pessoais

Integração com o BD

Respostas específicas para o
usuário
Mecanismo de Feedback

Valores

Melhor experiência do usuário
Personalização
Coleta de Feedback

Sistemas Inteligentes

ChatBot

Offline (base de conhecimento
treinada previamente)

Inputs

Dados sobre as despesas
inseridas
Feedback dos usuários após as
interações

Outputs

Você gastou R\$200 em
alimentação este mês
Solicitações de feedback ou
confirmações, como "Despesa
adicionada com sucesso. Foi
útil?"

Integrantes

Guilherme

João Madeira

João Lucas Curi

Arthur

fin-manage.joaomadeira12.repl.co



PUC MINAS- coreu



(32) 99984-3010

