

# DOCUMENTAÇÃO DA API RECEPTORA DE DADOS DO SISTEMA DE CONTAGENS

Faculdade Impacta — São Paulo, SP, Brasil, 2023

## INTEGRANTES:

Rafael Belmonte Izukawa  
Noemi Cavalcanti Almeida  
Miguel Vinicius Santos Albiach  
Victor Neri Kokado  
Guilherme Valverde Trindade  
Wesley Camilo da Silva Nascimento

Link do Github para acesso ao código:

[https://thecatinthebox@dev.azure.com/thecatinthebox/TCC/\\_git/BusPrime](https://thecatinthebox@dev.azure.com/thecatinthebox/TCC/_git/BusPrime).

## 1. INTRODUÇÃO

A API do projeto foi desenvolvida para receber os dados enviados pelo contador, e salvamento desses dados no banco de dados. O código fonte está disponível no Azure Web Apps, sendo possível acessar na url: <<https://busprime.azurewebsites.net/>>. Os dados recebidos são armazenados no banco de dados também disponibilizado pelo serviço de banco de dados da Azure.

## 2. LINGUAGENS E FERRAMENTAS

A API foi codificada em C# versão 10, e estruturado com o ASP.NET Core versão 6.0, utilizando a IDE Microsoft Visual Studio versão 1.78.2. O banco de dados foi criado com o SQL no gerenciador SQL Server Management Studio (ou SSMS) versão 12.0.2. A Microsoft oferece 12 meses de serviços gratuitos na plataforma Azure, sendo esse um dos motivos da utilização desses serviços.

Essas linguagens e ferramentas foram utilizadas pela facilidade encontrada no desenvolvimento, além da documentação da Microsoft ser extensa, possibilitando a simplicidade na implementação e codificação do projeto.

Existe no projeto, uma parcela de código em linguagens *front end*, como o HTML versão 5.3, CSS versão 4.15 e JavaScript 13ª edição que resultam em páginas *web*, onde o usuário pode consultar os dados salvos da contagem, e cadastrar novos registros de ônibus e motoristas.

## 3. MÉTODOS

A API é conectada na url: <<https://busprime.azurewebsites.net/>>, todos os métodos utilizados pelo registro dos dados estão nessa API. O principal método utilizado, nomeado “SendCount”, é responsável por receber os dados enviados pelo contador, através de uma requisição de protocolo HTTP e método POST. O *endpoint* requisitado deve ser “/Contador”, sendo um método POST. Os parâmetros da url estão no formato query, com as propriedades “contP”, sendo essa a contagem total de passageiros, “idBus”, sendo o código do ônibus e “idMot” equivalente ao código do motorista do ônibus. É validado a legitimidade dos dados do motorista e do ônibus antes do salvamento, depois realizado o cálculo da hora exata da

contagem, sendo considerado o horário de Brasília, e por fim os dados são salvos no banco de dados utilizando a biblioteca Entity Framework Core. O processo retorna um *status* 200 OK, ou um *status* de *Bad Request*, quando os dados inseridos são inválidos, ou um erro *Internal Server Error* (500), para os demais erros que ocorrerem no processo.

Como mostrado na figura 1, observa-se o trecho do código escrito na linguagem C# que contém os detalhes da entrada dos dados até o salvamento no banco de dados e concluindo o retorno do método.

Figura 1 — Trecho do código em C# da entrada dos dados até o salvamento no banco de dados e concluindo o retorno do método.

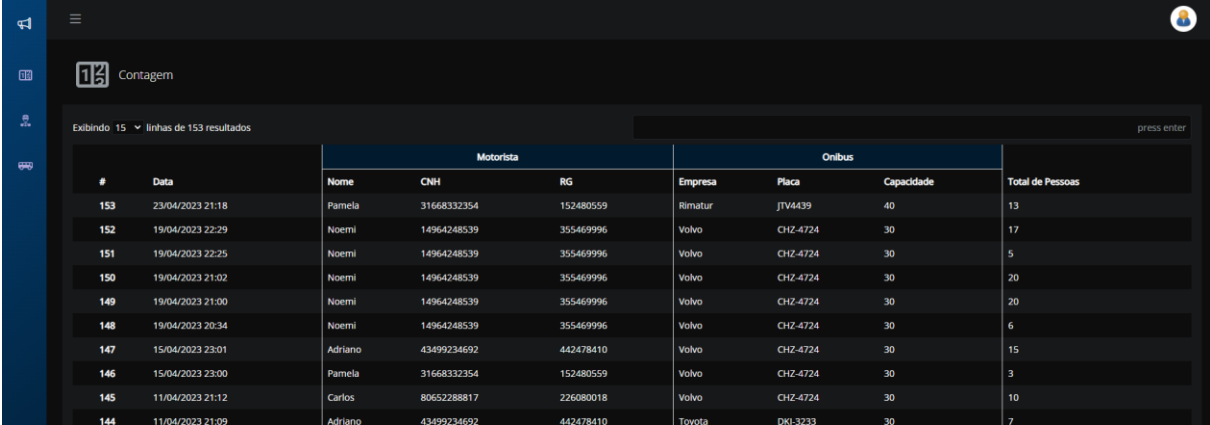
```
81 [HttpPost]
82 public ActionResult SendCount([FromQuery] ContagemRequest model)
83 {
84     try
85     {
86         if (!db.Motorista.Any(x => x.Id == model.idMot))
87             return BadRequest("Id de motorista não encontrado.");
88
89         if (!db.Onibus.Any(x => x.Id == model.idBus))
90             return BadRequest("Id de ônibus não encontrado.");
91
92         var date = DateTime.Now.AddHours(-3); //gtm sp
93
94         Contagem contagem = new Contagem()
95         {
96             Motorista_Id = model.idMot,
97             TotalPessoas = model.contP,
98             Onibus_Id = model.idBus,
99             DataContagem = date,
100         };
101
102         db.Contagem.Add(contagem);
103         db.SaveChanges();
104
105         return Ok();
106     }
107     catch (Exception e)
108     {
109         return BadRequest(e.Message);
110     }
111 }
```

Fonte: Os Autores

Outros métodos são utilizados para o cadastro e listagem dos motoristas e ônibus (CRUD). É possível a exibição desses dados através da listagem que considera filtros por texto.

A figura 2 mostra a exibição da listagem das contagens realizadas, integrando os dados do motorista e do ônibus da contagem, podendo ser acessado em: <<https://busprime.azurewebsites.net/contador>>.

Figura 2 — Exibição da listagem das contagens realizadas, integrando os dados do motorista e do ônibus da contagem

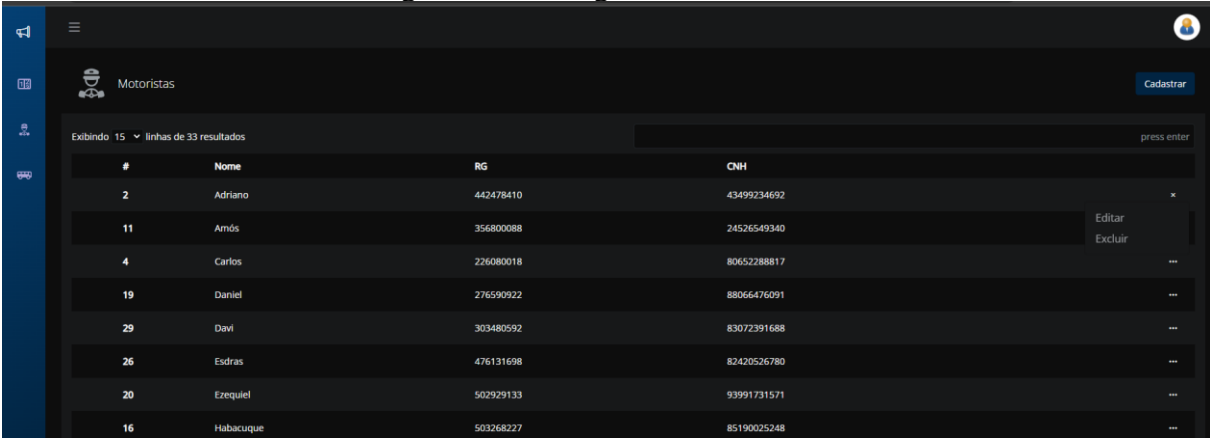


#	Data	Motorista			Onibus			Total de Pessoas
		Nome	CNH	RG	Empresa	Placa	Capacidade	
153	23/04/2023 21:18	Pamela	31668332354	152480559	Rimatur	JTV4439	40	13
152	19/04/2023 22:29	Noemi	14964248539	355469996	Volvo	CHZ-4724	30	17
151	19/04/2023 22:25	Noemi	14964248539	355469996	Volvo	CHZ-4724	30	5
150	19/04/2023 21:02	Noemi	14964248539	355469996	Volvo	CHZ-4724	30	20
149	19/04/2023 21:00	Noemi	14964248539	355469996	Volvo	CHZ-4724	30	20
148	19/04/2023 20:34	Noemi	14964248539	355469996	Volvo	CHZ-4724	30	6
147	15/04/2023 23:01	Adriano	43499234692	442478410	Volvo	CHZ-4724	30	15
146	15/04/2023 23:00	Pamela	31668332354	152480559	Volvo	CHZ-4724	30	3
145	11/04/2023 21:12	Carlos	80652288817	226080018	Volvo	CHZ-4724	30	10
144	11/04/2023 21:09	Adriano	43499234692	442478410	Volvo	CHZ-4724	30	7

Fonte: Os Autores

Os dados dos motoristas podem ser encontrados em <https://busprime.azurewebsites.net/motorista> como mostrado na figura 3.

Figura 3 — Listagem de Motoristas



#	Nome	RG	CNH
2	Adriano	442478410	43499234692
11	Amós	356800088	24526549340
4	Carlos	226080018	80652288817
19	Daniel	276590922	88066476091
29	Davi	303480592	83072391688
26	Esdras	476131698	82420526780
20	Ezequiel	502929133	93991731571
16	Habacuc	503268227	85190025248

Fonte: Os Autores

O cadastro e edição dos dados estão em <https://busprime.azurewebsites.net/motorista/cadastrar> e <https://busprime.azurewebsites.net/motorista/editar/<id>> respectivamente, sendo o de edição acessado pelo botão “Editar” do menu da lista. A exclusão também é realizada pelo menu da listagem em “exclusão”. A figura 4 mostra os campos para o cadastro e edição do motorista e a figura 5 mostra uma janela de confirmação para a exclusão do motorista.

Figura 4 — Dados de cadastro e edição do motorista

Form for driver registration and editing. The form has a dark background with light-colored text and input fields. At the top left, there is a '+' icon and the word 'Novo'. Below this, there are three input fields labeled 'Nome', 'RG', and 'CNH'. At the bottom, there is a blue button labeled 'Salvar'.

Fonte: Os Autores

Figura 5 — Exclusão do motorista

Screenshot of the driver management interface. A confirmation dialog is open, asking 'Tem certeza que deseja excluir?' (Are you sure you want to delete?). The dialog has a trash icon and an 'Excluir' button. In the background, there is a table with columns '#', 'Nome', 'RG', and 'CNH'. A search bar is visible with the text 'press enter'. A 'Cadastrar' button is also present.

Fonte: Os Autores

Os dados dos ônibus seguem o mesmo padrão dos motoristas, mudando apenas os dados inseridos, e a rota, que passa para: <<https://busprime.azurewebsites.net/onibus>>. A figura 6 mostra a página da listagem dos ônibus e a figura 7 mostra o cadastro e edição dos ônibus.

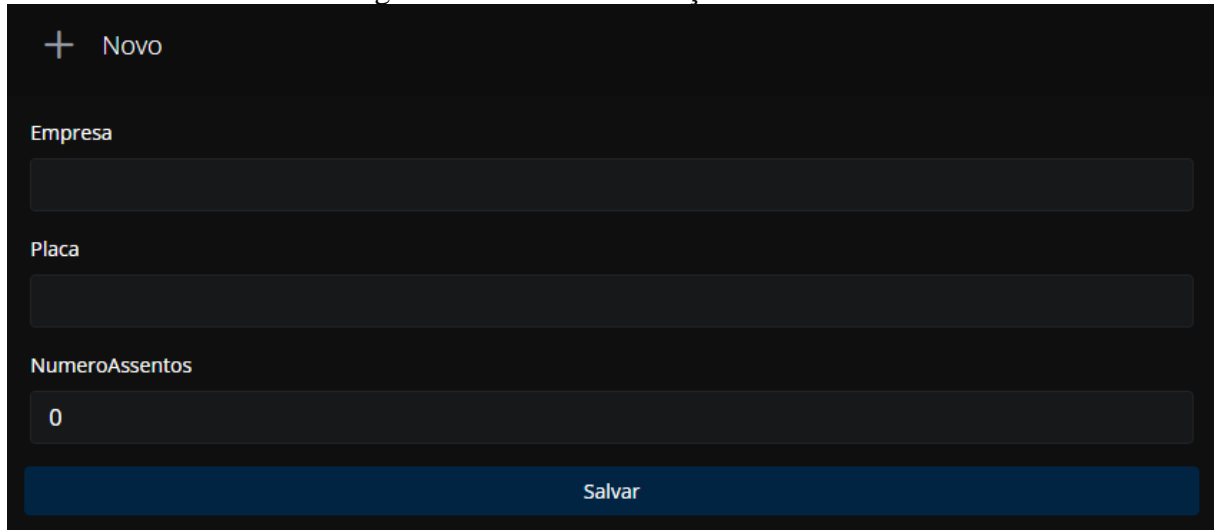
Figura 6 — Listagem de Ônibus

Screenshot of the bus listing interface. It shows a table with columns: '#', 'Empresa', 'Placa', and 'Número de Assentos'. The table contains 10 rows of data. A search bar is visible at the top right with the text 'press enter'. A 'Cadastrar' button is also present.

#	Empresa	Placa	Número de Assentos
33	Empresa X	HRW1802	50
9	Guanabara	JWE7381	40
10	Guanabara	NDC8669	40
11	Guanabara	MDL1817	40
12	Guanabara	NAP0478	40
18	Guanabara	HRW1801	50
3	Rimatur	HQA0346	40
4	Rimatur	MNJ6517	40
5	Rimatur	JTV4439	40
6	Rimatur	NEV0407	40

Fonte: Os Autores

Figura 7 — Cadastro e edição de Ônibus



The image shows a web form for bus registration and editing. At the top left, there is a plus icon and the text "Novo". Below this, there are three input fields: "Empresa", "Placa", and "NumeroAssentos". The "NumeroAssentos" field contains the value "0". At the bottom of the form, there is a blue button labeled "Salvar".

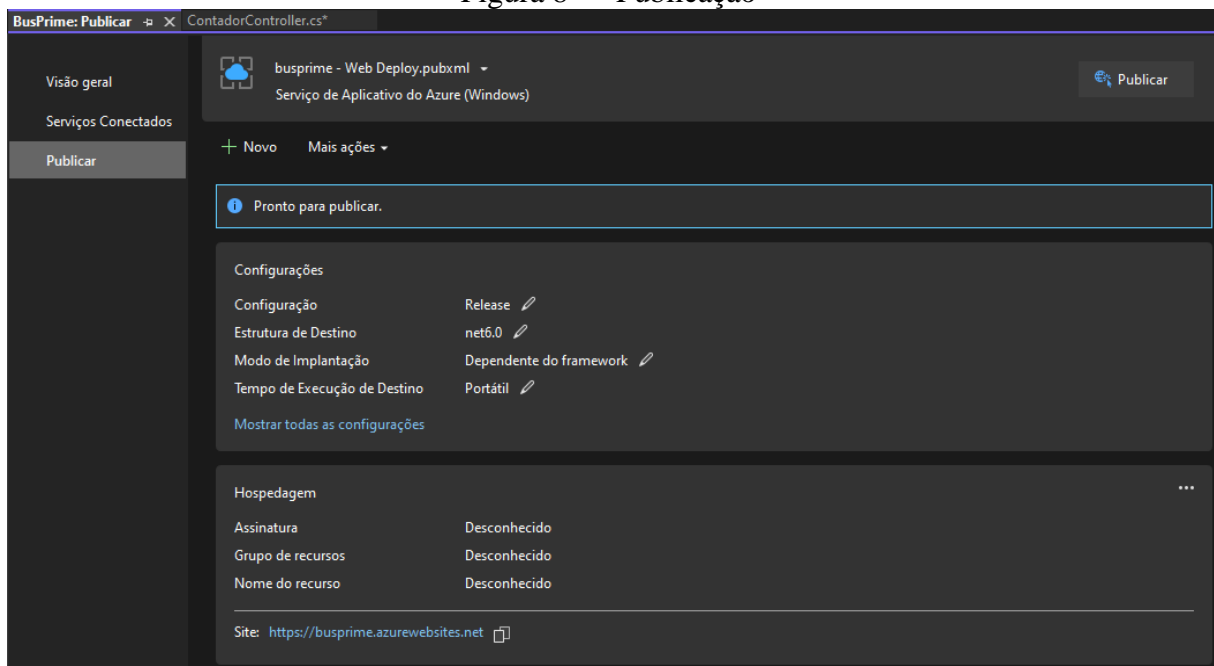
Fonte: Os Autores

#### 4. IMPLANTAÇÃO E PUBLICAÇÃO

O banco de dados foi facilmente implantado, já que a conexão no SSMS foi direta no serviço de banco de dados que foi anteriormente criado e disponibilizado.

Como mostrado na figura 8, a API e interface foi disposta por meio de um arquivo de publicação gerado pelo portal do Azure, sendo esse arquivo importado pelo Visual Studio, e sendo necessário apenas a ação do clique do botão exibido na interface.

Figura 8 — Publicação



Fonte: Os Autores