**Manual de utilização do Sistema Técnico para Teste**

**Tecnologias Envolvidas para o Projeto:**

**Sistema Windows 10 Home Single Language**

**SQL Server 2017 Communit Express**

**Visual Studio 2019 Communit**

**Bootstrap.Datepicker Version= 1.8.0.1**

**DinkToPdf Version= 1.0.8**

**Microsoft.AspNetCore.App**

**Microsoft.AspNetCore.Mvc.Core Version= 2.2.5**

**Microsoft.AspNetCore.Razor.Design Version= 2.2.0**

**Microsoft.VisualStudio.Web.CodeGen.Design Version= 2.2.4**

**Rotativa.AspNetCore Version= 1.1.1**

**Vercionamento GitHub na Azure**

**Servidor IIS Versão 10 Host .Net Core 3.1.0**

**Versao .Net Core 3.0.100**

**Infra Desktop Core i7 4.80Ghz 16Gb Ram SSD 480Gb**

****

**Banco de Dados:**

CREATE TABLE [dbo].[Cliente](

[IdCli] [int] IDENTITY(1,1) NOT NULL,

[Nome] [nvarchar](max) NULL,

[Cpf] [nvarchar](max) NULL,

[Contrato] [nvarchar](max) NULL,

[Data] [datetime2](7) NOT NULL,

[Valor\_principal] [decimal](18, 2) NOT NULL,

[Valor\_atualizado] [decimal](18, 2) NOT NULL,

CONSTRAINT [PK\_Cliente] PRIMARY KEY CLUSTERED

(

[IdCli] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

CREATE TABLE [dbo].[Debito](

[IdDeb] [int] IDENTITY(1,1) NOT NULL,

[Vencimento] [datetime2](7) NOT NULL,

[Valor\_inicial] [decimal](18, 2) NOT NULL,

[Valor\_final] [decimal](18, 2) NOT NULL,

[Juros] [decimal](18, 2) NOT NULL,

[Multa] [decimal](18, 2) NOT NULL,

[Dias] [int] NOT NULL,

CONSTRAINT [PK\_Debito] PRIMARY KEY CLUSTERED

(

[IdDeb] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Pagamento](

[IdPagto] [int] IDENTITY(1,1) NOT NULL,

[IdCli] [int] NOT NULL,

[Cpf] [nvarchar](max) NULL,

[Contrato] [nvarchar](max) NULL,

[Parcela] [int] NOT NULL,

[Data] [datetime2](7) NOT NULL,

[Valor] [decimal](18, 2) NOT NULL,

[Total] [varchar](50) NULL,

[Situacao] [nvarchar](max) NULL,

CONSTRAINT [PK\_Pagamento] PRIMARY KEY CLUSTERED

(

[IdPagto] ASC

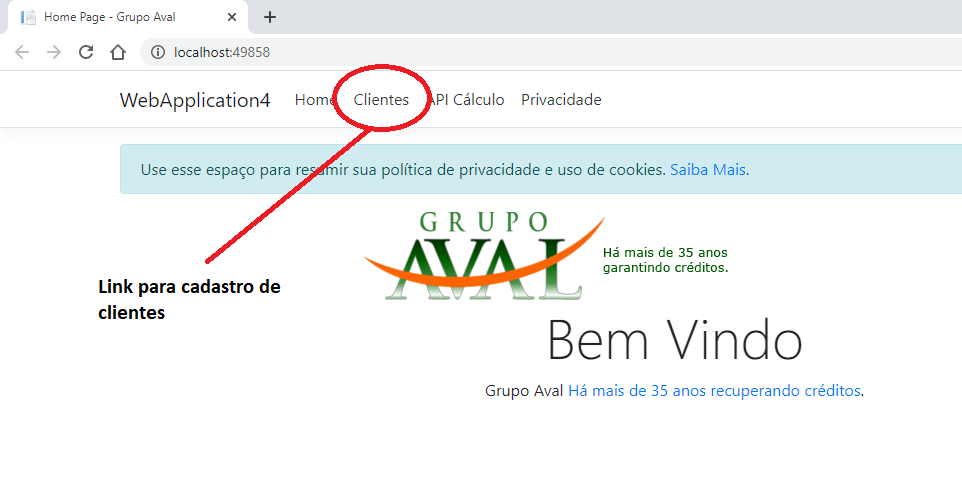
)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

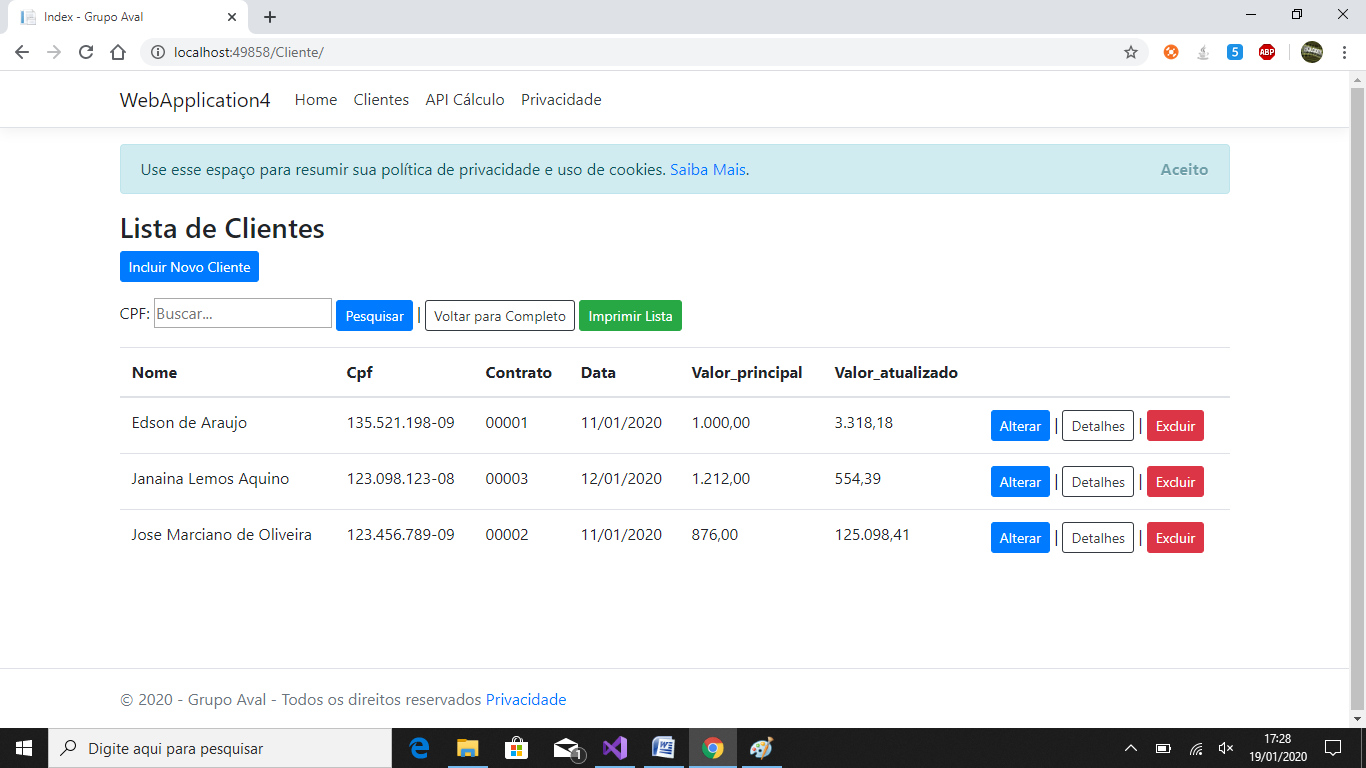
Após entrar no sistema no endereço abaixo:



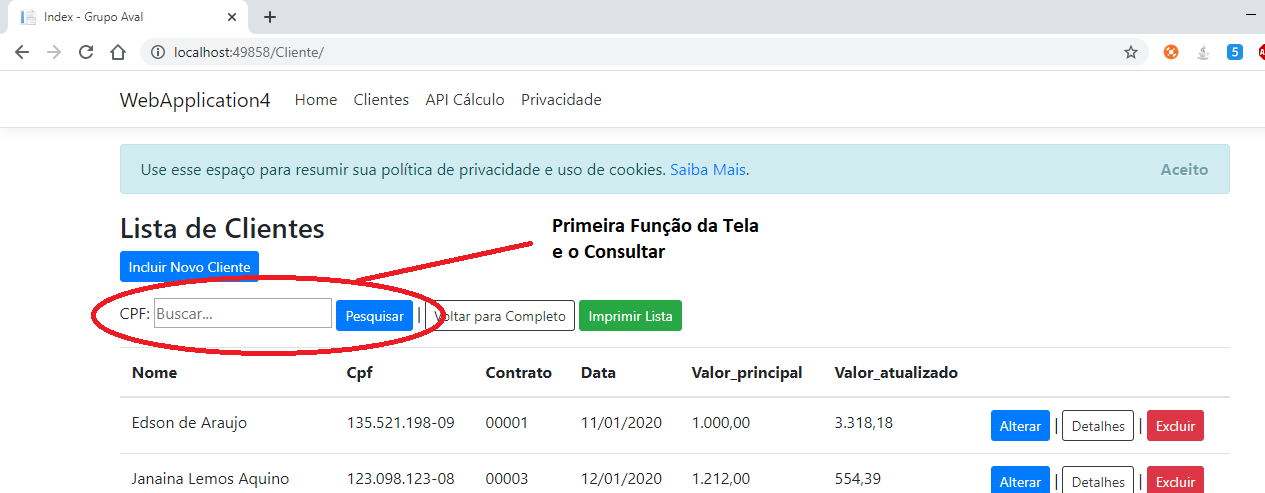
Vamos entrar no cadastro de **clientes** onde e possível incluir, alterar, excluir e imprimir.



Após clicar no link Cadastro teremos a tela como mostrada abaixo:



Vamos observar as opções que são mostradas na tela de cliente:



Primeira função e a consultar aqui no teste e apenas permitido consultar por CPF, o mesmo pode ser implementado a opção para outros filtros, como mostrado no código abaixo:

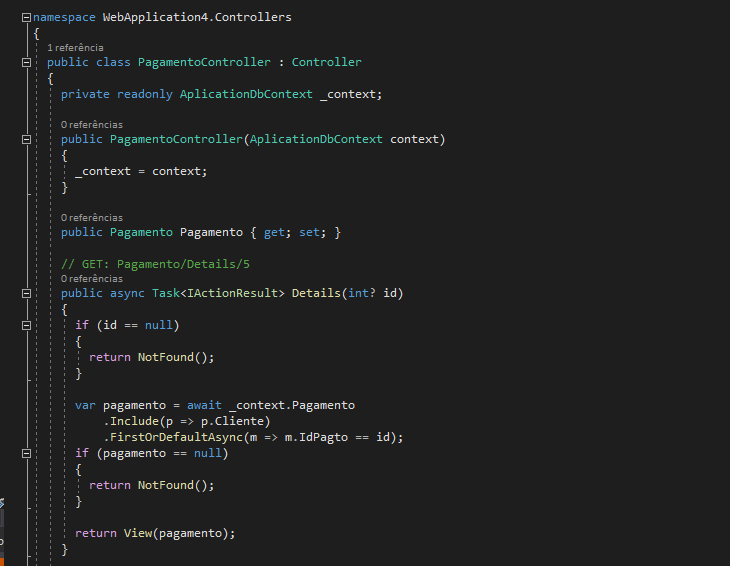


Abaixo todos os comando disponível para o operador:



Para elaboração do sistema foi usado três formas de processar os dados de entrada e saída:

Primeiro: **Entity framework core**



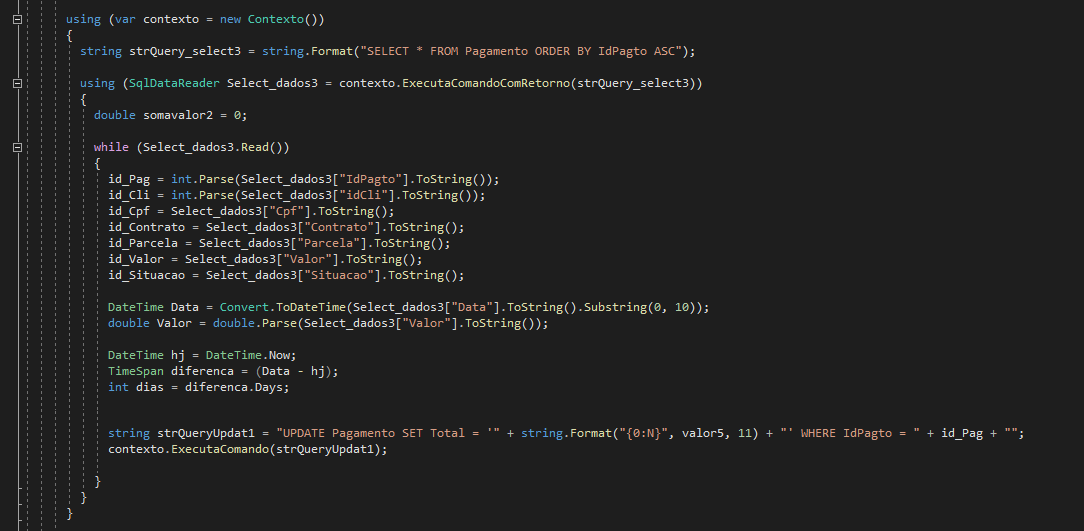
Criando a migração:

dotnet ef migrations add InitialCreate

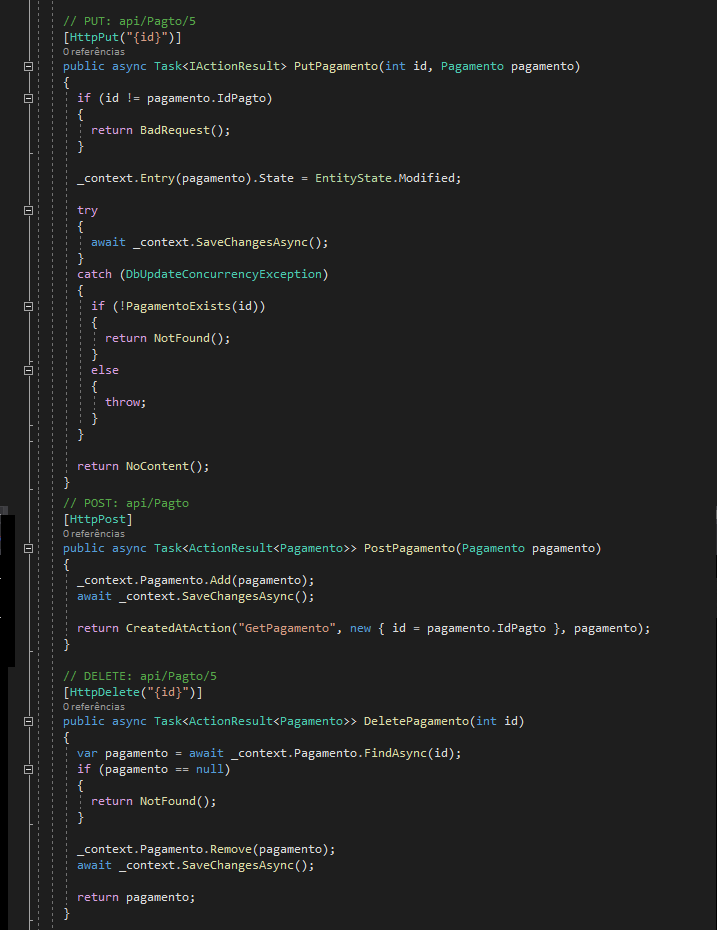
Atualizando o Banco:

dotnet ef database update

Segundo: **ADO.Net (Scripts SQL direto) LING**

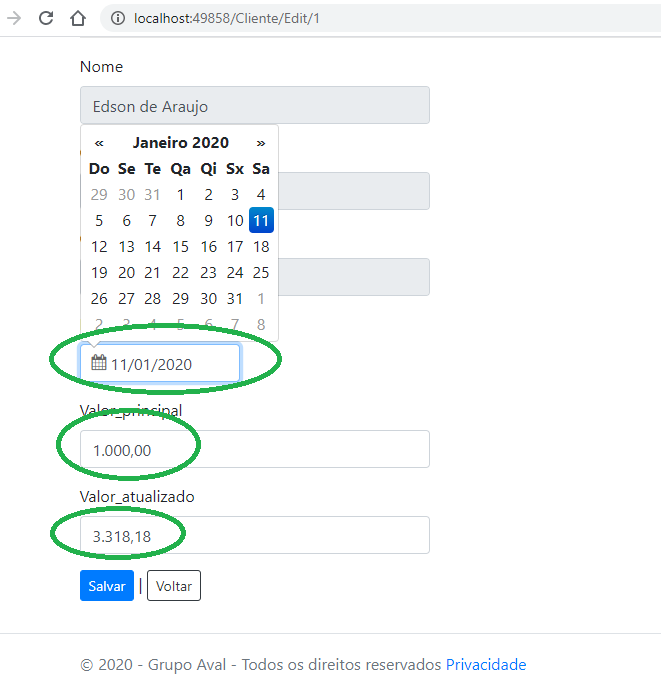
****

Terceira: **API Integração para busca no banco e Calculo do sistema.**

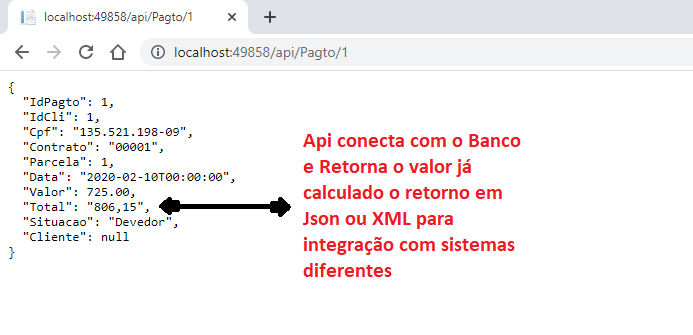
****

O Sistema de Teste conta com apoio de **Bootstrap 3.0 Jquery 4.0.1, JqueryUi** para formatação e mascara de campos como Data, Valor, CPF, CNPJ, Telefone.

Conforme imagem da tela abaixo:



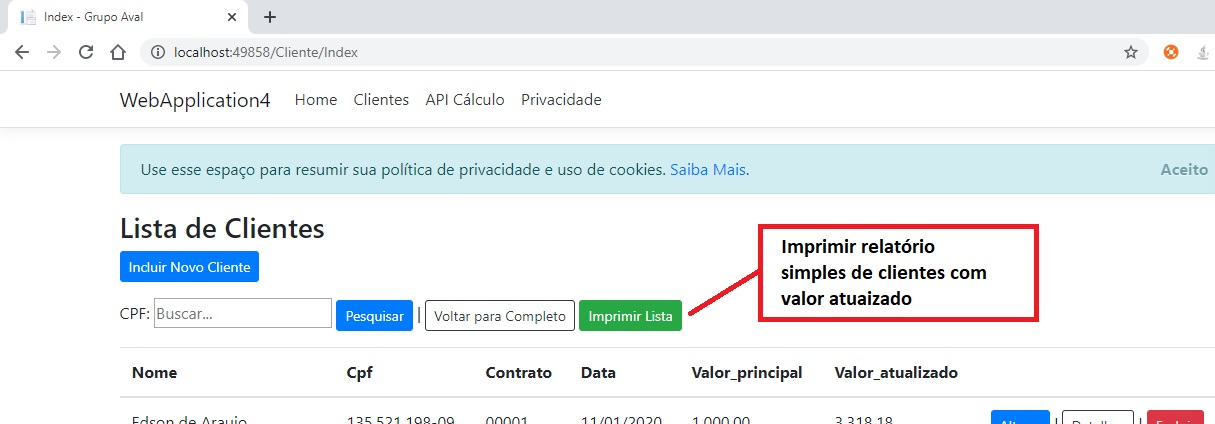
Nosso Retorno ao Consumir uma requisição API opções disponível **GET/POST/PUT/DELETE**

****

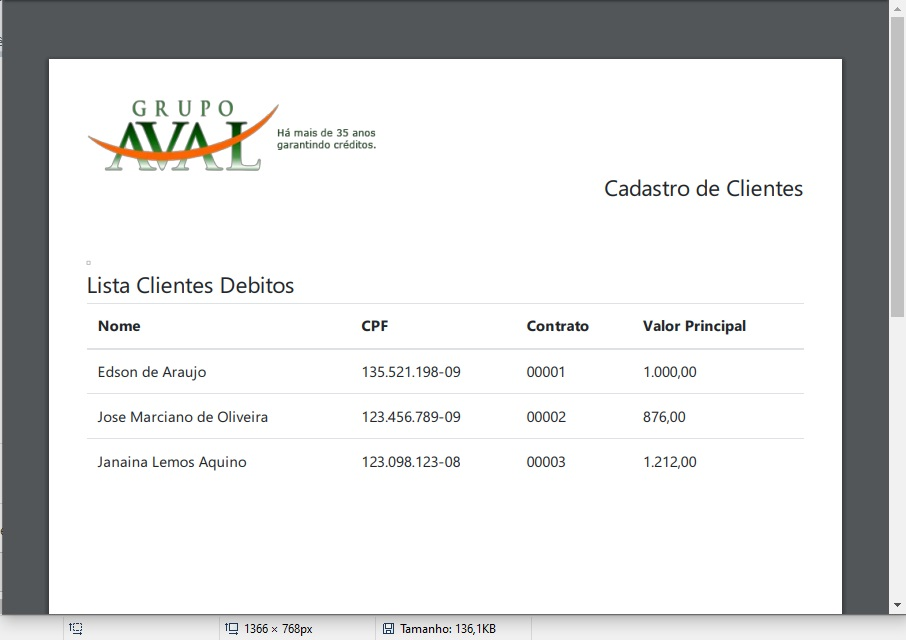
**Comando usado para a chamada Json.**

****

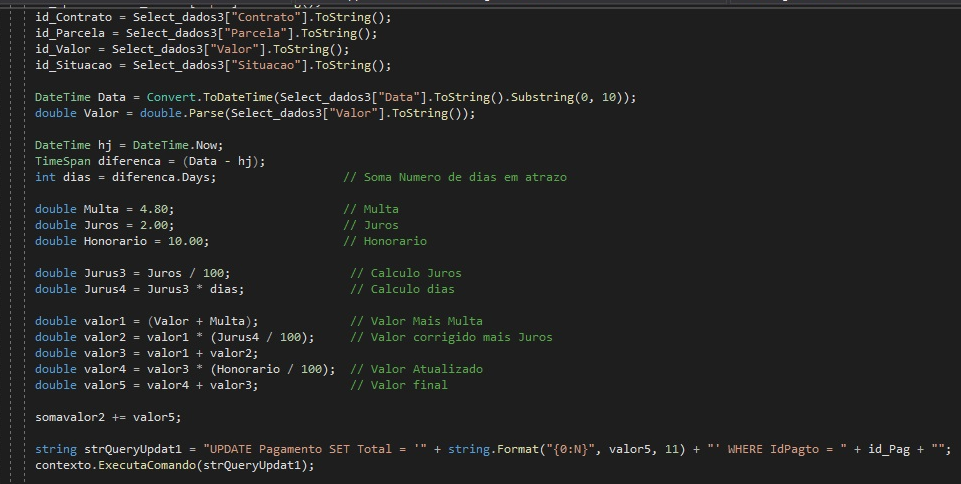
**Relatório gerado em PDF**

****

**Resultado Abaixo:**

****

Formula para Calculo:



Arvore do Projeto:

