Sumário

[1. Descrição Geral do Sistema 2](#_Toc503888093)

[1.1. Descrição do Problema 2](#_Toc503888094)

[1.2. Principais envolvidos e suas características 2](#_Toc503888095)

[1.2.1. Usuários do Sistema 2](#_Toc503888096)

[1.2.2. Cliente da PA 2](#_Toc503888097)

[1.2.3. Desenvolvedores dos sistemas de atendimento (integradores) 2](#_Toc503888098)

[1.3. Regras de Negócio 2](#_Toc503888099)

[1.3.1. Criar Fatura (CriarFatura) 3](#_Toc503888100)

[1.3.2. Gerar Segunda Via (SegundaVia) 4](#_Toc503888101)

[1.3.3. Buscar Fatura (BuscarFatura) 4](#_Toc503888102)

[1.3.4. Listar Faturas (BuscarTodasFaturas) 4](#_Toc503888103)

[1.3.5. Extrato de Faturas Pagas (BuscarFaturasPagas – primeira sobrecarga) 4](#_Toc503888104)

[1.3.6. Extrato de Faturas Pagas (BuscarFaturasPagas – segunda sobrecarga) 4](#_Toc503888105)

[1.3.7. Extrato de Faturas Pendentes (BuscarFaturasPendentes – primeira sobrecarga) 5](#_Toc503888106)

[1.3.8. Extrato de Faturas Pendentes (BuscarFaturasPendentes – segunda sobrecarga) 5](#_Toc503888107)

[1.3.9. Extrato de Faturas Pagas Parcialmente (BuscarFaturasComPagamentoParcial – primeira sobrecarga) 5](#_Toc503888108)

[1.3.10. Extrato de Faturas Pagas Parcialmente (BuscarFaturasComPagamentoParcial – segunda sobrecarga) 5](#_Toc503888109)

[1.3.11. Extrato de Faturas Canceladas (BuscarFaturasCanceladas – primeira sobrecarga) 5](#_Toc503888110)

[1.3.12. Extrato de Faturas Canceladas (BuscarFaturasCanceladas – segunda sobrecarga) 5](#_Toc503888111)

[1.3.13. Cancelar Fatura (CancelarFatura) 5](#_Toc503888112)

[1.3.14. Criar Cliente (CriarCliente) 5](#_Toc503888113)

[1.4. Requisitos do Sistemas 5](#_Toc503888114)

[2. Ambiente de Desenvolvimento 5](#_Toc503888115)

[3. Componentes externos utilizados 6](#_Toc503888116)

[4. Testes 6](#_Toc503888117)

[4.1. Plano de Teste 6](#_Toc503888118)

[4.2. Execução do Plano de Teste 6](#_Toc503888119)

# Descrição Geral do Sistema

O sistema BtorIugu atua como intermediário entre o cliente final (gerador/pagador de pendências financeiras) e o banco responsável por receber o pagamento, conforme fluxo abaixo:

## Descrição do Problema

Atualmente, as faturas são tratadas para bancos específicos, gerando mão de obra em massa na tratativa dos layouts específicos do banco solicitado pelo cliente.

## Principais envolvidos e suas características

### Usuários do Sistema

Os usuários lotados nos PA’s (Pontos de Atendimento) responsáveis pelos lançamentos das vendas são os principais utilizadores da plataforma BtorIugu, uma vez que a estes são atribuídas as funções para emissão das faturas. Estes terão atuação somente na aplicação que incorpora o projeto BtorIugu não tendo participação direta na codificação

### Cliente da PA

Os clientes estão envolvido no que diz respeito a fatura gerada e recebem as notificações disparadas pelo BtorIugu no cadastro de sua fatura (geração do boleto), quando ocorre alguma alteração seja no cancelamento deste boleto ou na alteração da data de vencimento do boleto, gerando o cancelamento da fatura atual e a emissão de uma nova.

### Desenvolvedores dos sistemas de atendimento (integradores)

Os desenvolvedores têm a principal utilização da plataforma, visto que o BtorIugu é transparente para os dois primeiros envolvidos. É cargo do desenvolvedor capturar os registros gerados pelo sistema integrador e popular os objetos necessários para emissão do boleto de cobrança da fatura em questão. É cargo do desenvolvedor, também, manter o controle das mensagens de erro reportadas pela plataforma BtorIugu, assim como a melhor apresentação ao envolvido 1.2.1 deste documento.

## Regras de Negócio

O projeto BtorIugu, por ser construído em uma arquitetura de bibliotecas, é um sistema integrável por qualquer ambiente da plataforma Microsoft apenas com o apontamento destes arquivos de referência no software integrador. O BtorIugu é composto por 14 métodos que terão as suas regras e particularidades explicitadas a seguir, seguindo a premissa de que para utilizá-los é necessário criar uma instância da classe Iugu e, obrigatoriamente, a passagem do parâmetro TOKEN[[1]](#footnote-1) no método construtor

### Criar Fatura (CriarFatura)

Este método é o responsável por receber os dados gerados no sistema integrador e emitir o boleto da venda faturada. Seu retorno é do tipo object o que permite obter qualquer propriedade do Json convertido como: o código da nova fatura (id), a data de vencimento (due\_date), a data de criação (created\_at\_iso), o link de impressão da fatura (secure\_url) ou o código do cliente.

Para que este método não retorne uma exceção (erro) é necessário criar uma instância do objeto FaturaEnvio e popular as propriedades a seguir:

#### email (e-mail):

E-mail do cliente que receberá o boleto gerado em seu correio eletrônico. Este endereço também receberá as atualizações que, por ventura, acontecerão com este documento. O campo é do tipo caractere, não pode estar em branco e deve seguir o padrão [nome@dominio.com(.br](mailto:nome@dominio.com(.br) – opcional).

#### due\_date (data de vencimento):

A data de vencimento deve estar no formato “AAAA-MM-DD”, mas quando utilizado classes como DateTime é perfeitamente aceitável e a data informada não pode estar no passado.

#### ensure\_workday\_due\_date (data de vencimento em dia útil)

Booleano que, quando verdadeiro, garante que a data de vencimento seja apenas em dias úteis (segunda a sexta). Obs.: Não verifica feriados.

#### items (Itens da fatura):

Nesta coleção do tipo Produto, são informados os itens que compõem a venda e possui propriedades que obrigatórias pois terão seus valores calculados para gerar o totalizador do boleto a ser emitido, sendo assim, pelo menos um item deve ser informado.

##### *Items.description (descrição do produto/serviço)*

O campo *description* (caractere) é um campo não obrigatório, mas terá seu valor impresso no boleto

##### *Items.price (valor do produto/serviço)*

O campo *price* (caractere) é um campo não obrigatório, mas terá seu valor impresso no boleto

##### *Items.price\_cents (valor do produto em centavos)*

O campo *price\_cents* (inteiro) é um campo obrigatório e será utilizado juntamente com o campo *quantity* para gerar o valor total do boleto a ser pago. O menor valor possível para este campo é 100 (cem) pois sua expressão é feita em centavos, ficando o valor mínimo de R$ 1,00 (um real).

##### *Items.quantity*

O campo *quantity* (inteiro) é um campo obrigatório e será utilizado para gerar o valor total do boleto a ser pago. O menor valor possível é 1 (um).

#### payer (nome que será impresso no boleto)

Neste objeto do tipo Cliente, são informadas as propriedades obrigatórias para a emissão de boletos registrados e para o sistema de antifraude.

##### *payer.cpf\_cnpj (CPF ou CNPJ do pagador)*

O campo *cpf\_cnpj* (caractere) tem a verificação de documento existente.

##### *payer.name (nome do pagador)*

O campo *name* (caractere) não pode ficar em branco.

##### *payer.address (endereço do pagador)*

O objeto *address* do tipo Endereco possui propriedades obrigatórias para a conclusão da emissão do boleto e estão destacadas a seguir:

##### *payer.address.zip\_code (CEP do pagador)*

O campo *zip\_code* (caractere) deve ser preenchido e terá seu valor verificado no padrão 99999-999.

##### *payer.address.number (número da residência do pagador)*

O campo *number* (inteiro) torna-se obrigatório quando informado o CEP do pagador. Quando não informado, é assumido o valor 0 (zero).

#### payable\_with (pago com)

Método de pagamento que será disponibilizado para esta Fatura (“all”, “credit\_card” ou “bank\_slip”). Obs: Caso esta Fatura esteja atrelada à uma Assinatura, a propriedade é herdar o valor atribuído na Assinatura; Caso esta esteja atribuído o valor “all”, o sistema considerará o “payable\_with” da Fatura, se não, o sistema considerará o que estiver na Assinatura.

### Gerar Segunda Via (SegundaVia)

Este método gera a segunda via de uma fatura cancelando a primeira fatura e gerando uma nova com o status *pending* (pendente). Apenas faturas com o status pendente são passíveis de segunda via e para que sua ação seja executada são necessários 2 (dois) parâmetros sendo eles o *idIugu* (caractere) que deve conter o código identificador da fatura a ser substituída e o *dataAlteracao* (datetime) que é a nova data de vencimento da fatura a ser gerada. Em seu retorno é possível obter o código da nova fatura gerada.

### Buscar Fatura (BuscarFatura)

Este método busca a fatura especificada em seu parâmetro *idFatura* (caractere).

### Listar Faturas (BuscarTodasFaturas)

Retorna todas as faturas tendo como filtro o parâmetro *customer\_id[[2]](#footnote-2)* (caractere).

### Extrato de Faturas Pagas (BuscarFaturasPagas – primeira sobrecarga)

Na primeira sobrecarga do método, apenas o parâmetro *customer­\_id* (caractere) e são retornadas todas as faturas com status *paid* (pago) filtrando os pagamentos efetuados em D-1 (data anterior a data atual).

### Extrato de Faturas Pagas (BuscarFaturasPagas – segunda sobrecarga)

Na segunda sobrecarga do método, além do parâmetro *customer\_id* (caractere) a *dataInicio* (datetime) e *dataFim* (datetime) são necessários para serem filtrados os boletos com status *paid\_at* (pago em) entre o período informado.

### Extrato de Faturas Pendentes (BuscarFaturasPendentes – primeira sobrecarga)

Neste método onde o *customer\_id* (caractere) é necessário, são retornados todos os boletos com status *pending* (pendente) deste cliente.

### Extrato de Faturas Pendentes (BuscarFaturasPendentes – segunda sobrecarga)

Neste método são retornados todos os boletos com status *pending* (pendente) que possuem *due\_date* (data de vencimento) entre os dias *dataInicio* (datetime) e *dataFim* (datetime).

### Extrato de Faturas Pagas Parcialmente (BuscarFaturasComPagamentoParcial – primeira sobrecarga)

Neste método são retornados todos os boletos com status *partially\_paid* (parcialmente pago) filtrando pelo *customer\_id* (caractere).

### Extrato de Faturas Pagas Parcialmente (BuscarFaturasComPagamentoParcial – segunda sobrecarga)

Aqui são retornados os boletos do *customer\_id* (caractere) considerando como filtro os boletos com status *partially\_paid* (parcialmente pago) com o registro de *paid\_at* (pago em) entre *dataInicio* (datetime) e *dataFim* (datetime).

### Extrato de Faturas Canceladas (BuscarFaturasCanceladas – primeira sobrecarga)

Neste método são retornados todos os boletos com status *canceled* (cancelado) filtrando apenas o *customer\_id* (caractere).

### Extrato de Faturas Canceladas (BuscarFaturasCanceladas – segunda sobrecarga)

Neste método são retornados todos os boletos com status *canceled* (cancelado) e que estão com o registro de *created\_at* (criado em) entre *dataInicio* (datetime) e *dataFim* (datetime).

### Cancelar Fatura (CancelarFatura)

Cancela a futura do registro informado no parâmetro *idFatura* (caractere).

### Criar Cliente (CriarCliente)

Método utilizado para criar a chave do *customer\_id* onde o parâmetro informado é do tipo Cliente e as propriedades *email* (caractere) e *name* (caractere) são obrigatórias.

## Requisitos do Sistemas

A contratação prévia do serviço Iugu é básica para que todo fluxo descrito funcione. O contrato com esta empresa possibilita a criação do *customer\_id* e o *token* primário, assim como a especificação da conta bancária que receberá as receitas dos boletos quitados.

# Ambiente de Desenvolvimento

O desenvolvimento da solução BtorIugu foi no seguinte ambiente:

* Visual Studio 2015;
* .NET Framework 4.5.1;
* Linguagem de Programação C#;

Para otimizar o desempenho e liberar recursos para melhorar a performance do projeto, foram criados métodos que eliminam os objetos alocados antes sem que isso fique atrelado ao fluxo natural do Garbage Collector.

# Componentes externos utilizados

Para conversão dos objetos Json em objetos complexos C#, foram utilizadas as classes da biblioteca Newtonsoft.Json;

# Testes

Para validar os métodos criados no projeto BtorIugu...

## Plano de Teste

## Execução do Plano de Teste

1. TOKEN é uma chave de autenticação atribuída ao ponto de atendimento ou ao provedor responsável pelos pontos de atendimento. Este registro somente pode ser gerado no portal da Iugu e deve ser feito por um usuário com acesso (autenticação) do contratante do serviço. [↑](#footnote-ref-1)
2. O *customer\_id* é uma chave agrupadora de boletos gerados, quando não informado no registro da fatura, o boleto não terá afinidade a nenhum cliente e estará avulsa dentro dos registros emitidos. [↑](#footnote-ref-2)