

# Como adicionar e atualizar itens em uma tabela de formulário filho

# Como adicionar e atualizar itens em uma tabela de formulário filho

Este guia explica como **inserir e atualizar registros em uma tabela de formulário filho** a partir de um **formulário pai**, utilizando **JavaScript + componente CsHtml (partial)**.

O exemplo usa dados fictícios apenas para demonstrar o funcionamento.

---

## Visão geral da solução

O fluxo funciona assim:

1. Criamos um **Formulário Filho** (referenciando o pai).
  2. Criamos um **Formulário Pai** e associamos o formulário filho como uma tabela.
  3. No formulário pai, criamos um **botão** que chama um método JavaScript no onclick.
  4. Esse método JavaScript carrega um **componente CsHtml** (partial).
  5. Dentro do CsHtml, executamos um loop para **inserir registros na tabela do formulário filho**.
  6. Após a execução, a tabela é **atualizada automaticamente na tela**.
-

# 1. Criar o Formulário Pai

Primeiro, crie um **formulário pai** contendo os campos que você julgar necessário.

- Valor da Operação
- Responsável pela operação

# 2. Criar o Formulário Filho

Em seguida, crie um **formulário filho** contendo os campos que serão inseridos na tabela.

Exemplo de campos:

- Tipo de recebível
- Cedente
- Sacado
- Quantidade de parcelas
- Data de vencimento
- Valor nominal
- **id\_pai (IMPORTANTE, aqui você cria o campo que vai associar o form pai e filho)**  
\*\*\*

\*\*\* detalhes de como fazer a relação Pai e Filho podem ser encontrado em outros docs na Wiki do agilityflow

---

# 2. Associar o Formulário Pai ao Filho

No formulário pai:

- Associe o formulário filho como uma tabela
- Garanta que ele esteja configurado como **tabela de formulário filho**

Recupere o Id dessa tabela, no nosso exemplo é: "0e7f3b2f-dd68-a2d4-fe4e-b68e1c3957ef"

Usaremos mais abaixo, algo assim:

```
var tableid_recebivel = Guid.Parse("0e7f3b2f-dd68-a2d4-fe4e-b68e1c3957ef");
```

---

## 3. Criar o componente CsHtml (partial)

Agora criamos o **componente CsHtml** que será carregado pelo JavaScript, crie com o nome **import\_fake**

Esse componente será responsável por:

- Inserir registros no formulário filho
  - Retornar o controle para o JavaScript
- 

## 4. Lógica do CsHtml para inserir registros

Dentro do CsHtml, usamos C# para inserir os dados.

Neste exemplo:

- Só inserimos se o formulário for **novo**
- Ignoramos formulários em **rascunho**
- Criamos **10 registros fictícios** apenas para demonstração

Exemplo:

```
@{  
  
    //id da tabela  
    var tableid_recebivel = Guid.Parse("0e7f3b2f-dd68-a2d4-fe4e-b68e1c3957ef"); //table id DO FORMULÁRIO  
    FILHO  
  
  
    //testa se é um form NOVO e se NÃO é RASCUNHO pois nesse exemplo quero inserir apenas para formulário
```

NOVOS que não sejam RASCUNHOS

```
var isNovo = Model.Id.IsNullOrEmpty();
var isRascunho = !Model.RascunhoId.IsNullOrEmpty();

if(isNovo && !isRascunho){

    for (var i = 0;i<10;i++){

        var parcelas = Random.Shared.Next(1, 19);

        //info do formulario filho
        var campos_e_valores = new DataDictionary();
        campos_e_valores["tipo_de_recebivel"] = "e253f97a-8315-4ed9-b5f8-9820f7152a60";
        campos_e_valores["cedente"] = "335e3af4-f97d-4cf4-9b5c-faf5d14c726d";
        campos_e_valores["sacado"] = "12a5c07d-d955-4dc2-85cf-55b20d889761";
        campos_e_valores["qtd_de_parcelas"] = parcelas.ToString();
        campos_e_valores["data_de_vencimento"] = DateTime.Now.AddMonths(1).ToUserTimeZone().ToString();
        campos_e_valores["valor_nominal"] = "35000";

        //adiciona na tabela filha
        //retorna o id do rascunho desse formulario filho, pois o ID final só será gerado ao Salvar o Formulario Pai
        var formRascunhoId = await PageContext.AddChildFormAsync( tableid_recebivel, campos_e_valores);

    }

}
}
```

---

## 5. Criar o botão no Formulário Pai

No formulário pai, crie um botão com um **onclick** chamando uma função JavaScript.

Exemplo:

```
import_fake();
```

Esse botão será responsável por iniciar o processo de inserção.

---

## 6. JavaScript do Formulário Pai

No JavaScript do formulário pai, carregamos o **componente CsHtml** usando `loadCsHtmlComponent`.

Esse componente cshtml executará a lógica de inserção e ao retornar faremos via Javascript o "refresh" da tabela para que os dados inseridos apareçam

Exemplo:

```
function import_fake() {  
  
    formContext.loadCsHtmlComponent("import_fake", {  
  
        onSuccess: function (result) {  
            // formContext.msg.success("Executado com sucesso", "Confirmação");  
  
            var tableid_recebivel = "0e7f3b2f-dd68-a2d4-fe4e-b68e1c3957ef"  
  
            formContext.childForm.datatable.refresh(tableid_recebivel);  
  
        }  
  
    });  
}
```

Pontos importantes:

- O `onSuccess` é executado após o CsHtml terminar.
- O `datatable.refresh` garante que os novos registros apareçam na tela.

## 7. O que acontece internamente

- Cada chamada de `AddChildFormAsync` cria um **registro no formulário filho**
  - O ID final do registro só será gerado quando o **formulário pai for salvo**
  - O retorno do método é o **ID do rascunho**, usado internamente pelo sistema
- 

## 8. Atualizando os dados da tabela

Após o CsHtml finalizar:

- O JavaScript executa `datatable.refresh`
  - A tabela do formulário filho é recarregada
  - Os registros recém-inseridos aparecem imediatamente na tela
- 

## Observações importantes

- Este exemplo usa dados **fictícios**
- Sempre valide se o formulário não está em rascunho
- Para atualizações, a lógica pode ser adaptada para:
  - Buscar registros existentes
  - Alterar valores
  - Inserir apenas registros que não existam

Se você precisar atualizar um registro basta seguir o código usando o **PageContext.UpdateChildForm** como no exemplo abaixo:

@{

//id da tabela

```
var tableid_relacaoFormularioPaiFilhold = Guid.Parse("0dd874a9-9c39-4e4f-acc3-02015a60f6bb"); //table id  
DO FORMULÁRIO FILHO
```

//info do formulario filho

```
var campos_e_valores = new DataDictionary();  
campos_e_valores["nome_filho"] = "Exemplo 01 - " + DateTime.Now;  
campos_e_valores["valor_numerico"] = PageContext.FormatNumberToString_EnglishFormat(999999.99, 2);
```

//testa se é um form NOVO e se NÃO é RASCUNHO pois nesse exemplo quero inserir apenas para formulário NOVOS que não sejam RASCUNHOS

```
var isNovo = Model.Id.IsNullOrEmpty();  
var isRascunho = !Model.Rascunhold.IsNullOrEmpty();
```

```
if(isNovo && !isRascunho){
```

//adiciona na tabela filha

```
//retorna o id do rascunho desse formlario filho, pois o ID final só será gerado ao Salvar o Formulario PAi  
var formRascunhold = PageContext.AddChildForm( tableid_relacaoFormularioPaiFilhold, campos_e_valores);
```

//atualiza o item que acabamos de adicionar, apenas para demonstração

```
[[Guid? idFormularioFilho = null;
```

```
campos_e_valores["nome_filho"] = "Exemplo Atualizado 0123456789 - " + DateTime.Now;
```

```
PageContext.UpdateChildForm(
```

```
    //se ainda não foi salvo pela primeira vez, o item só terá o Id do Rascunho  
    rascunhold,
```

```
    //se já foi salvo pela primeira vez, o item já terá o Id
```

```
    //então passa aqui o id (nesse exemplo estamos passando null para atualizar apenas o  
    idFormularioFilho,
```

```
    tableid_relacaoFormularioPaiFilhold,  
    campos_e_valores);
```

□}

}

---

Revision #1

Created 23 December 2025 22:22:18 by agilityflow

Updated 23 December 2025 22:36:22 by agilityflow