

Criando ações complexas no botão de um formulário

Opção 1, usando o c# direto no cshtml

Nesse exemplo vamos supor que temos 2 formulários

1. Solicitação de Orçamento
2. Orçamento

Dentro do Solicitação de Orçamento você tem um botão chamado "Criar Orçamento" e ao clicar nesse Criar orçamento você quer criar um novo Formulário de Orçamento.

No Formulário Solciitação de Orçamento, crie um **javascript** como esse exemplo:

```
function criarOrcamento() {
    formContext.loadAjax("criarOrcamento", {
        onSuccess: function (e) {
            //pageMsg.showMsgSuccess("Orcamento criado", "Sucesso");

            var jsonResult = JSON.parse(e);

            if (!jsonResult.success) {
                pageMsg.showMsgError(jsonResult.msg, "Falha");
            } else {

                var idEstruturaFormToRedirect = "74ad209b-ee6a-4ae3-9920-2caad2bc7451";
                window.location.href = formContext.url.getBaseUrl() + "#/fluxo/index/" + idEstruturaFormToRedirect +
                "/" + jsonResult.idOrcamento;
            }

        },
        onError: function (err) {
```

```
        pageMsg.showMsgError("Erro ao criar o Orcamento", "Falha");
    }
});
}
```

No onclick do botão "Criar Orçamento" chame a função javascript

```
criarOrcamento()
```

Agora crie um Componente Cshtml, chamando **CriarOrcamento**

Dentro do componente coloque uma programação em C#.

C#

```
@{
    var success = false;
    var msg = "";
    var tel = PageContext.GetValue("seu_telefone");
    Guid? idOrcamento = null;

    if (!string.IsNullOrEmpty(tel))
    {
        // Verifica se o cliente já existe
        var query = $"SELECT id FROM x_tbl_clientes WHERE telefone = '{tel}' AND deletado = false LIMIT 1";
        var dtCliente = await PageContext.GetDataTableAsync(query);
        var clienteExiste = dtCliente.Rows.Count > 0;
        Guid idCliente;

        //se o cliente ainda nao existir, cria um cliente
        if (!clienteExiste)
        {
            var idFormularioCliente = Guid.Parse("cb4019ad-c731-42b9-8624-bb9c0fe5bf78");
            var valuesCliente = new DataDictionary
            {
                { "nome", PageContext.GetValue("seu_nome") },
                { "email", PageContext.GetValue("email") },
                { "instagram", PageContext.GetValue("instagram") },
            }
        }
    }
}
```

```

        { "telefone", tel }
    };

    idCliente = await PageContext.SaveEntityAsync(idFormularioCliente, valuesCliente);
}
else
{
    idCliente = Guid.Parse(dtCliente.Rows[0]["id"].ToString());
}

// Criação do orçamento
var idFormularioOrçamento = Guid.Parse("74ad209b-ee6a-4ae3-9920-2caad2bc7451");
var valuesOrçamento = new DataDictionary
{
    { "solicitacao", PageContext.FormId },
    { "radiosolicitacao", "dcef9208-3a3e-3f34-90f9-e514af597d93" },
    { "numero_convocado_adultos", PageContext.GetValue("qtd_adultos") },
    { "numero_convocado_crianças", PageContext.GetValue("qtd_crianças") },
    { "datahora", PageContext.GetValue("datahora_do_evento") },
    { "local", PageContext.GetValue("local_do_evento") },
    { "status", "6d5257d2-034a-459a-b57a-cc4800e74fb0" },
    { "tipo_de_orçamento", PageContext.GetValue("tipo_de_evento") },
    { "cliente", idCliente.ToString() }
};

idOrçamento = await PageContext.SaveEntityAsync(idFormularioOrçamento, valuesOrçamento);
success = true;
}
else
{
    msg = "O telefone não pode estar em branco";
}
}
{"success": @success.ToString().ToLower(), "msg": "@msg", "idOrçamento": "@idOrçamento" }

```

Opção 2, usando regra de negócio

Nesse exemplo vamos supor que temos 2 formulários

1. Solicitação de Orçamento
2. Orçamento

Dentro do Solicitação de Orçamento você tem um botão chamado "Criar Orçamento" e ao clicar nesse Criar orçamento você quer salvar o formulário de Solicitação de Orçamento mudando o status dele para "Orçamento Criado" e ao mesmo tempo, já criar um novo Formulário de Orçamento.

Importante, coloque uma regra no formulário de Solicitação de Orçamento pra deixar invisível o botão de "Criar Orçamento" sempre que já tiver um orçamento criado, ou seja, que o status seja igual ao "Orçamento Criado";

No Formulário Solciitação de Orçamento, crie um **javascript** como esse exemplo:

```
function criarOrcamento() {
  pageMsg.showMsgSuccess("Criando orçamento...");
  formContext.field.setValue("status", "041cc896-6048-4b56-a9ac-033f6144253c");//coloca o status "orçamento
em criação", supondo que o staus "orçamento em criação" é esse id 041cc896-6048-4b56-a9ac-033f6144253c
  setTimeout(function () {
    formContext.form.save();

  }, 500); //espera 500ms pra seguir o salvamento
}
```

No onclick do botão "Criar Orçamento" chame a função javascript

```
criarOrcamento()
```

Agora nas regras de negocio, crie uma regra para criar um registro novo de orçamento ao SALVAR.

Coloque um If pra enviar o orçamento apenas se o status seja "orçamento em criação"

Dentro do if coloque uma programação em C#.

C#

```
@{

    //muda o status da SOLicitçaão de ORçamento pra "Orçamento Criado"
    FormContext.SetValue("status", "9041cc899-7045-cb56-b9ac-033f61442599");//coloca o status "orçamento
    criado", supondo que o staus "orçamento criado" é esse id 9041cc899-7045-cb56-b9ac-033f61442599

    var tel = FormContext.GetValue("seu_telefone");
    if (!string.IsNullOrEmpty(tel))
    {
        // Verifica se o cliente já existe
        var query = $"SELECT id FROM x_tbl_clientes WHERE telefone = '{tel}' AND deletado = false LIMIT 1";
        var dtCliente = await FormContext.GetDataTableAsync(query);
        var clienteExiste = dtCliente.Rows.Count > 0;
        Guid idCliente;

        //se o cliente ainda nao existir, cria um cliente
        if (!clienteExiste)
        {
            var idFormularioCliente = Guid.Parse("cb4019ad-c731-42b9-8624-bb9c0fe5bf78");
            var valuesCliente = new DataDictionary
            {
                { "nome", FormContext.GetValue("seu_nome") },
                { "email", FormContext.GetValue("email") },
                { "instagram", FormContext.GetValue("instagram") },
                { "telefone", tel }
            };

            idCliente = await FormContext.SaveEntityAsync(idFormularioCliente, valuesCliente);
        }
        else
        {
            idCliente = Guid.Parse(dtCliente.Rows[0]["id"].ToString());
        }
    }
}
```

```
// Criação do orçamento
var idFormularioOrçamento = Guid.Parse("74ad209b-ee6a-4ae3-9920-2caad2bc7451");
var valuesOrçamento = new DataDictionary
{
    { "solicitacao", FormContext.FormId },
    { "radiosolicitacao", "dcef9208-3a3e-3f34-90f9-e514af597d93" },
    { "numero_convidado_adultos", FormContext.GetValue("qtd_adultos") },
    { "numero_convidado_crianças", FormContext.GetValue("qtd_crianças") },
    { "datahora", FormContext.GetValue("datahora_do_evento") },
    { "local", FormContext.GetValue("local_do_evento") },
    { "status", "6d5257d2-034a-459a-b57a-cc4800e74fb0" },
    { "tipo_de_orçamento", FormContext.GetValue("tipo_de_evento") },
    { "cliente", idCliente.ToString() }
};

}
else
{
    FormContext.WarningMessage("O telefone não pode estar em branco");
}
}
```

Revision #4

Created 11 February 2025 14:30:23 by agilityflow

Updated 5 March 2025 12:33:44 by agilityflow