

```

function setSelectedZoomItem(selectedItem) {

    if (selectedItem.inputId == "zoomEmpresaProtheus") {

        $('#cdEmpresaProtheus').val(selectedItem.M0_CODIGO);
        $('#cdFilialProtheus').val(selectedItem.M0_CODFIL);
        $('#filialProtheus').val(selectedItem.M0_NOMECON);
    }

    if (selectedItem.inputId == "txt_natureza") {
        $('#txt_cdNatureza')
        .val(selectedItem.ED_CODIGO);
    }

    if (selectedItem.inputId == "zoomNomeCompleto") {

        window['zoomcpfCnpj'].setValue(selectedItem.A1_CGC);
        $('#altelefone').val(selectedItem.A1_TEL);

        var retornaDadosClientes =
recuperaDadosClientes(selectedItem.A1_COD+selectedItem.A1_LOJA)
        retornaDadosClientes.then(function(dados) {
            console.log(dados)
            preencheCamposClientes(dados)
        }).catch(erro => {
            FLUIGC.toast({message : "Problema para recuperar os
dados.", type: "warning"});
            console.error(erro);
        });
    }

    if (selectedItem.inputId == "zoomcpfCnpj") {

        window['zoomNomeCompleto'].setValue(selectedItem.A1_NOME
);
        var retornaDadosClientes =
recuperaDadosClientes(selectedItem.A1_COD+selectedItem.A1_LOJA)
        retornaDadosClientes.then(function(dados) {
            console.log(dados)
            preencheCamposClientes(dados)
        }).catch(erro => {
            FLUIGC.toast({message : "Problema para recuperar os
dados.", type: "warning"});
        });
    }
}

```

```

        console.error(error);
    });

}

if (selectedItem.inputId == "zoomOperadora") {
    $('#cdOperadora').val(selectedItem.A1_COD);
    $('#allojaOper').val(selectedItem.A1_LOJA);
    $('#a1CgcOper').val(selectedItem.A1_CGC);
}

if (selectedItem.inputId == "zoomMoeda") {
    $('#cdMoeda').val(selectedItem.CODIGO);
}

if (selectedItem.inputId == "zoomreferencia") {

    $('#cdMoeda').val(selectedItem.E1_MOEDA);
    $('#moeda').val(selectedItem.NOME_MOEDA);
    $('#valorOrdem').val(selectedItem.E1_VALOR);
    $('#nomeCliente').val(selectedItem.A1_NOME);
    $('#alcod').val(selectedItem.E1_CLIENTE);
    $('#cdNatureza').val(selectedItem.E1_NATUREZ);

}
}

function removedZoomItem(removedItem) {

if (removedItem.inputId == "zoomEmpresaProtheus") {
    $('#cdEmpresaProtheus').val("");
    $('#cdFilialProtheus').val("");
    $('#filialProtheus').val("");
}

if (removedItem.inputId == "txt_natureza") {
    $('#txt_cdNatureza').val("");
}

if (removedItem.inputId == "zoomNomeCompleto") {
    window['zoomcpfCnpj'].clear();
    $('#spreadUtilizado').val("");
}

if (removedItem.inputId == "zoomcpfCnpj") {
    window['zoomNomeCompleto'].clear();
    $('#spreadUtilizado').val("");
}
}

```

```

        }

        if (removedItem.itemId == "zoomOperadora") {
            $('#cdOperadora').val("");
            $('#allojaOper').val("");
            $('#a1CgcOper').val("");
        }

        if (removedItem.itemId == "zoomMoeda") {
            $('#cdMoeda').val("");
        }

        if (removedItem.itemId == "zoomreferencia") {

            $('#cdMoeda').val("");
            $('#moeda').val("");
            $('#valorOrdem').val("");
            $('#nomeCliente').val("");
            $('#a1cod').val("");
            $('#cdNatureza').val("");

        }
    }

$( document ).ready(function() {

    // Verifica qual opção de rádio está marcada quando houver uma mudança
    $('input[name="rdTipoInclusao"]').change(function() {
        // Verifica se a opção "Ordem" está marcada
        if ($(this).val() == 'ordem') {
            // Esconde todos os elementos com a classe "classFechamento"
            $('.classFechamento').hide();
            $('.classFechamentoRef').hide();
            $('.classOrdemRef').show();

            limparCamposFechamento();

        } else if ($(this).val() == 'fechamento') {
            // Caso contrário, mostra os elementos com a classe "classFechamento"
            $('.classFechamento').show();
            $('.classOrdemRef').hide();
            $('.classFechamentoRef').show();

            limparCamposOrdem();
        }
    });
});

```

```

        }  

        else if($(this).val() == 'anexo') {  

            }  

            requiredFieldsRule();  

        });  

        requiredFieldsRule();  

        enableFields();  

  

        if(CURRENT_STATE == INICIO_0 || CURRENT_STATE ==  

INICIO){  

    $('.classFechamento').hide();  

    $('.classFechamentoRef').hide();  

} else {  

    enableField($('#zoomEmpresaProtheus'), false);  

}  

  

if(CURRENT_STATE == COMISSAO || CURRENT_STATE == FIM){  

    enableField($('#comissao'), true);  

  

    enableField($('#rdTipoInclusaoOrdem'), false);  

    enableField($('#zoomNomeCompleto'), false);  

    enableField($('#referencia'), false);  

    enableField($('#zoomcpfCnpj'), false);  

    enableField($('#dataCredito'), false);  

    enableField($('#zoomreferencia'), false);  

    enableField($('#tipoOperacao'), false);  

    enableField($('#zoomOperadora'), false);  

    enableField($('#zoomMoeda'), false);  

    enableField($('#moedaEstrangeira'), false);  

    enableField($('#dataFechamento'), false);  

    enableField($('#dataFechamento'), false);  

    enableField($('#taxaCliente'), false);  

    enableField($('#taxaSpot'), false);  

    enableField($('#despesaBanc'), false);  

    enableField($('#iof'), false);  

    enableField($('#spreadUtilizado'), false);  

    enableField($('.btnUpFile'), false);  

  

    if($("#input[name='rdTipoInclusao']:checked").val()  

== "ordem") {  

}

```

```

        $("#nomeCliente").parent().parent().addClass("fs-
display-none");

        $("#zoomreferencia").parent().parent().addClass("fs-
display-none");

        $("#cpfCnpjCliente").parent().parent().addClass("fs-
display-none");

        $("#valorOrdem").parent().parent().addClass("fs-
display-none");
        $("#moeda").parent().parent().addClass("fs-
display-none");

    }

}

if(CURRENT_STATE == FECHAMENTO) {
    enableField($('rdTipoInclusaoOrdem'), false);
    enableField($('zoomNomeCompleto'), false);
    enableField($('referencia'), false);
    enableField($('zoomcpfCnpj'), false);
    enableField($('dataCredito'), false);

    $("#dataCredito").removeClass("campo-data")

    enableField($('moeda'), false);
    //enableField($('moedaEstrangeira'), false);

    if($('input[name="rdTipoInclusao"]:checked').val() ==
    "ordem") {
        $('.classFechamentoRef').hide();
        $('.classOrdemRef').show();

    } else if
    ($('input[name="rdTipoInclusao"]:checked').val() ==
    "fechamento"){
        $('.classOrdemRef').hide();
        $('.classFechamentoRef').show();
    }
}

FLUIGC.calendar('.campo-Data');

$(".campo-data").change(function() {

```

```

        if (this.value != '') {
            validateDate($('#' + this));
        });

        $(".mask-money").maskMoney({
            prefix: '',
            thousands: ',',
            decimal: ',',
            precision: 4,
            affixesStay: true,
            allowZero: true
        });
    });
}

function calcTotal(elemento) {

    var moedaEstrangeira =
    parseFloat($("#moedaEstrangeira").maskMoney("unmasked")[0]);
    var taxaCliente =
    parseFloat($("#taxaCliente").maskMoney("unmasked")[0]);
    var taxaSpot =
    parseFloat($("#taxaSpot").maskMoney("unmasked")[0]);
    var iof =
    parseFloat($("#iof").maskMoney("unmasked")[0]);
    var despesaBancaria =
    parseFloat($("#despesaBanc").maskMoney("unmasked")[0]);

    if ($("#tipoOperacao").val() == "1") {
        var total = (taxaCliente * moedaEstrangeira) -
        despesaBancaria;
        var spreadReais = (taxaSpot - taxaCliente) *
        moedaEstrangeira;

        // Formata os totais para exibir corretamente
        var totalFormatado = total.toLocaleString('pt-
        BR', { minimumFractionDigits: 4, maximumFractionDigits: 4 });
        var spreadReaisFormatado =
        spreadReais.toLocaleString('pt-BR', { minimumFractionDigits:
        4, maximumFractionDigits: 4 });

        $("#valorTotal").val(totalFormatado);
        $("#spreadEmReais").val(spreadReaisFormatado);
    }

    if ($("#tipoOperacao").val() == "2") {
}

```

```

        var total = (taxaCliente * moedaEstrangeira) +
despesaBancaria;
        var spreadReais = (taxaCliente - taxaSpot) *
moedaEstrangeira;

        // Formata os totais para exibir corretamente
        var totalFormatado = total.toLocaleString('pt-
BR', { minimumFractionDigits: 4, maximumFractionDigits: 4 });
        var spreadReaisFormatado =
spreadReais.toLocaleString('pt-BR', { minimumFractionDigits:
4, maximumFractionDigits: 4 });

        $("#valorTotal").val(totalFormatado);
        $("#spreadEmReais").val(spreadReaisFormatado);
    }

    if($("#tipoOperacao").val() == "3"){
        var total = moedaEstrangeira * taxaCliente -
(moedaEstrangeira * taxaCliente * (iof / 100)) -
despesaBancaria;
        var spreadReais = (taxaSpot - taxaCliente) *
moedaEstrangeira;

        // Formata os totais para exibir corretamente
        var totalFormatado = total.toLocaleString('pt-
BR', { minimumFractionDigits: 4, maximumFractionDigits: 4 });
        var spreadReaisFormatado =
spreadReais.toLocaleString('pt-BR', { minimumFractionDigits:
4, maximumFractionDigits: 4 });

        $("#valorTotal").val(totalFormatado);
        $("#spreadEmReais").val(spreadReaisFormatado);
    }

    if($("#tipoOperacao").val() == "4"){
        var total = taxaCliente * moedaEstrangeira +
(taxaCliente * moedaEstrangeira * (iof / 100)) +
despesaBancaria;
        var spreadReais = (taxaCliente - taxaSpot) *
moedaEstrangeira;

        // Formata os totais para exibir corretamente
        var totalFormatado = total.toLocaleString('pt-
BR', { minimumFractionDigits: 4, maximumFractionDigits: 4 });
        var spreadReaisFormatado =
spreadReais.toLocaleString('pt-BR', { minimumFractionDigits:
4, maximumFractionDigits: 4 });

```

```

        $("#valorTotal").val(totalFormatado);
        $("#spreadEmReais").val(spreadReaisFormatado);
    }

}

function validateDate(objeto){
    var aDate = moment(objeto.val(), 'DD/MM/YYYY', true);
    var isValid = aDate.isValid();

    if(!isValid){
        FLUIGC.toast({
            title: 'Data: ',
            message: 'Data Inválida',
            type: 'warning'
        });
        objeto.val("");
        objeto.focus();
    }
}

function limparCamposFechamento(){
    $('.campoFechamento').each(function(){
        const fieldType = $(this).prop('tagName').toLowerCase();
        if ($(this).is('input[type="text"]')) {
            $(this).val('');
        } else if (fieldType === 'select') {
            $(this).val('');
        }
    });
}

window["zoomreferencia"].clear()
$("#tipoOperacao").change()

}

function limparCamposOrdem(){
    $('.campoOrdem').each(function(){
        const fieldType = $(this).prop('tagName').toLowerCase();
        if ($(this).is('input[type="text"]')) {
            $(this).val('');
        } else if (fieldType === 'select') {
            $(this).val('');
        }
    });
}

```

```

        window["zoomNomeCompleto"].clear()
        window["zoomcpfCnpj"].clear()
        window["zoomMoeda"].clear()
        window["zoomOperadora"].clear()

        $("#tipoOperacao").change()
    }

function recuperaDadosClientes(id_cliente) {
    return new Promise(resolve => {

        let loading = FLUIGC.loading(window, {
            textMessage : "Recuperando dados..."
        });
        loading.show();

        let errorHandled = false; // Variável para rastrear se o erro já foi tratado

        var cId_Cliente =
        DatasetFactory.createConstraint('A1_COD', id_cliente,
        id_cliente, ConstraintType.MUST);

        DatasetFactory.getDataset("ds_retornaClienteProtheus",
        null, new Array(cId_Cliente), null, {
            success : function(dados) {

                if (!errorHandled) { // Verifica se o erro já foi tratado
                    try {
                        if(dados != null &&
                        dados.values.length > 0 && dados.values[0].status ==
                        undefined ){
                            FLUIGC.toast({message : "Dados recuperados com sucesso.", type: "success"});
                        } else {
                            FLUIGC.toast({message : "Não foi possível recuperar os dados.", type: "warning"});
                        }
                        loading.hide();
                        resolve(dados.values);
                    } catch (err) {
                }
            }
        })
    }
}

```

```

                console.log("Não foi possível recuperar
os dados: ", err);
                FLUIGC.toast({message : "Não foi possível
recuperar os dados", type: "danger"});
                loading.hide();
                errorHandled = true; // Define a variável
como true para indicar que o erro foi tratado
            }
        }

    ), error : function(err){
        if (!errorHandled) { // Verifica se o erro já
foi tratado
            console.log("Não foi possível recuperar
os dados: ", err);
            FLUIGC.toast({message : "Não foi possível
recuperar os dados", type: "danger"});
            loading.hide();
            errorHandled = true; // Define a variável
como true para indicar que o erro foi tratado
        }
    }
}

));
}

}

}

function preencheCamposClientes (retornoDataset) {
    const [registro] = retornoDataset;

    const {
        A1COD: alcod,
        A1LOJA: alloja,
        A1PESSOA: alpessoa,
        A1XTIPOCL: tipoCliente
    } = registro;

    $("#alcod").val(alcod);
    $("#alloja").val(alloja);
    $("#alpessoa").val(alpessoa);
    $("#tipoCliente").val(tipoCliente);

}

function formatarDataAAAAMMDDParaDDMMYYYY (dataAAAAMMDD) {

```

```

    // Extrair o ano, mês e dia da data fornecida
    var ano = dataAAAAMMDD.slice(0, 4);
    var mes = dataAAAAMMDD.slice(4, 6);
    var dia = dataAAAAMMDD.slice(6, 8);

    // Formatar a data como DD/MM/AAAA
    var dataFormatada = dia + '/' + mes + '/' + ano;

    return dataFormatada;
}

function formatarTelefone(numeroTelefone) {
    // Remover todos os caracteres não numéricos
    var numeroLimpo = numeroTelefone.replace(/\D/g, '');

    // Verificar se é um número de telefone com 8 ou 9
    // dígitos
    if (numeroLimpo.length === 8) {
        // Aplicar máscara para números de 8 dígitos: 1234-
5678
        return numeroLimpo.replace(/(\d{4})(\d{4})/, '$1-$2');
    } else if (numeroLimpo.length === 9) {
        // Aplicar máscara para números de 9 dígitos: 12345-
6789
        return numeroLimpo.replace(/(\d{5})(\d{4})/, '$1-$2');
    } else {
        // Se o número não corresponder a um número de
        // telefone válido, retornar o número original
        return numeroTelefone;
    }
}

```