

```

function addLinhaTb() {
    // Adicionar filho

    var indices = retornaIndices(['dadositensseguradora'])

    if(indices.length > 25) {

        exibirAviso("Não é possível incluir mais de 26
linhas de comissão.");

        return false;
    }

    var row = wdkAddChild('dadositensseguradora')
$:"#seqLinha__" + row).val(row);
//$("#txt_seguradoraSaldo__"+ row).mask("#.##0,00", {
reverse: true });
$(".monetario").mask("#00.000.000.000,00", {reverse:
true});
$(".validaCompetencia").mask("00/0000");
calendarioCompetencia(row)
FLUIGC.calendar('.campo-Data');

}

function addLinhaTbNew() {
    // Adicionar filho
    var row = wdkAddChild('dadositensNewProposta')
$:"#seqLinhaNew__" + row).val(row);

    $(".monetario").mask("#00.000.000.000,00", {reverse:
true});

}

function addLinhaTbBanco() {
    // Adicionar filho
    var row = wdkAddChild('dadositensBancarios')
$:"#idBanco__" + row).val(row);
$:"#agencia__" + row).mask("0000");
$:"#digitoAgencia__" + row).mask("0");
$:"#conta__" + row).mask("00000000");

}

function calcularSoma() {
    clearTimeout($ this).data('timeout'));
}

```

```

$(this).data('timeout', setTimeout(function() {
    var total = 0;

    $("input[id^='txt_seguradoraSaldo___']").each(function() {
        var valor = $(this).val();
        valor = parseFloat(valor.replace(/\./g,
        '') .replace(',', '.')) || 0;
        total += valor;
    });

    var totalFormatado = total.toLocaleString('pt-BR',
    {
        minimumFractionDigits: 2,
        maximumFractionDigits: 2
    });
    $("#valorTotal").val(totalFormatado.replace('.', ','));
}, 300));
}

function setSelectedZoomItem(selectedItem) {
    if (selectedItem.inputId.indexOf("___") > -1) {
        var indice = selectedItem.inputId.split("___")[1];
    }

    if (selectedItem.inputId == "zoomEmpresaProtheus") {
        $('#cdEmpresaProtheus').val(selectedItem.M0_CODIGO);
        $('#cdFilialProtheus').val(selectedItem.M0_CODFIL);
        $('#filialProtheus').val(selectedItem.M0_NOMECON);
    }

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

        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);
        })
    }
}

```

```

    });

    window['zoomcpfCnpj'].setValue(selectedItem.A1_CGC);
}

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

window['zoomNomeCompleto'].setValue(selectedItem.A1_NOME
);
}

if (selectedItem.inputId == "uf") {
    $('#codUF').val(selectedItem.X5_CHAVE);
    reloadZoomFilterValues("cidade",
"ESTADO,"+selectedItem.X5_CHAVE);
}

if (selectedItem.inputId == "cidade") {
    $('#codCidade').val(selectedItem.CC2_CODMUN);
}

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

    $('#codEstadoNascimento').val(selectedItem.X5_CHAVE);
    reloadZoomFilterValues("cidadeNascimento",
"ESTADO,"+selectedItem.X5_CHAVE);
}

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

```

```

        $('#codCidadeNascimento').val(selectedItem.CC2_CODMUN);
    }

    if (selectedItem.inputId == "banco____"+indice) {
        $('#idBanco____'+indice).val(selectedItem.codigoBanco);
    }

    if (selectedItem.inputId ==
"txt_seguradoraProp____"+indice) {
        $('#txt_cdSeguradora____'+indice).val(selectedItem.A1_COD);
    }

    if (selectedItem.inputId == "txt_natureza____"+indice) {
        $('#txt_cdNatureza____'+indice).val(selectedItem.ED_CODIGO);
    }
}

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

    if (removedItem.inputId.indexOf("___") > -1) {
        var indice = removedItem.inputId.split("___")[1];
    }

    if (removedItem.inputId == "zoomNomeCompleto") {
        window['zoomcpfCnpj'].clear();
        $('#dataNascimento').val("");
        $('#cep').val("");
        $('#rua').val("");
        $('#bairro').val("");
        $('#numero').val("");
        $('#cidade').val("");
        $('#uf').val("");
        $('#endExterior').val("");
        $('#nifIcamaId').val("");
        $('#nomeConjuge').val("");
    }
}

```

```

        $('#cpfConjuge').val("");
        $('#saidaDefinitiva').val("");
        $('#telefoneBrasil').val("");
        $('#telefoneExterior').val("");

        $('#banco').val("");
        $('#agencia').val("");
        $('#digitoAgencia').val("");
        $('#conta').val("");
        $('#digitoConta').val("");
        $('#situacao').val("");
        $('#tipoConta').val("");
        $('#formaPagDB').val("");
    }

    if (removedItem.inputId == "zoomcpfCnpj") {
        window['zoomNomeCompleto'].clear();
        $('#dataNascimento').val("");
        $('#cep').val("");
        $('#rua').val("");
        $('#bairro').val("");
        $('#numero').val("");
        $('#cidade').val("");
        $('#uf').val("");
        $('#endExterior').val("");
        $('#nifIcamaId').val("");
        $('#nomeConjuge').val("");
        $('#cpfConjuge').val("");
        $('#saidaDefinitiva').val("");
        $('#telefoneBrasil').val("");
        $('#telefoneExterior').val("");

        $('#banco').val("");
        $('#agencia').val("");
        $('#digitoAgencia').val("");
        $('#conta').val("");
        $('#digitoConta').val("");
        $('#situacao').val("");
        $('#tipoConta').val("");
        $('#formaPagDB').val("");
    }

    if (removedItem.inputId == "txt_seguradora__"+indice) {
        $('#txt_cdSeguradora__'+indice).val("");
    }

    if (removedItem.inputId == "txt_natureza__"+indice) {

```

```

        $('#txt_cdNatureza____'+indice).val("");
    }

    if (removedItem.inputId == "uf") {
        $('#codUF').val("");
        reloadZoomFilterValues("cidade", "ESTADO,");
        window["cidade"].clear();
        $('#codCidade').val("");
    }

    if (removedItem.inputId == "cidade") {
        $('#codCidade').val("");
    }

    if (removedItem.inputId == "estadoNascimento") {
        $('#codEstadoNascimento').val("");
        reloadZoomFilterValues("cidadeNascimento",
        "ESTADO,");
        window["cidadeNascimento"].clear();
        $('#codCidadeNascimento').val("");
    }

    if (removedItem.inputId == "cidadeNascimento") {
        $('#codCidadeNascimento').val("");
    }

    if (removedItem.inputId == "banco____"+indice) {
        $('#idBanco____'+indice).val("");
    }
}

$( document ).ready(function() {
    exibeSaldoOuComissao();

    if(CURRENT_STATE == FIM || CURRENT_STATE ==
    TRATAR_ERRO) {
        $(".retornoProtheus").parent().removeClass("fs-
display-none")
    }
})

```

```

$(document).on('change', "[name='rdTipoInclusao']",
function (e) {
    requiredFieldsRule();
    exibeSaldoOuComissao()
    limparCamposSaldoEComissao()
    calcularSoma()
    calcularSomaComissao()
});

$(document).on('keyup', '.caixa_alta', function (e) {
    $(this).val($(this).val().toUpperCase());
});

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

requiredFieldsRule();

$(".campo-data").change(function () {
    if (this.value != '') {
        validateDate($ (this));
    }
});

loadJsonDataText();
// if ($("#rdMenorIdade").val() == "sim") {
//     $("#menorIdadeInputs").show();
//     $("#menorIdadeInputsEnd").show();
//     $("#menorIdadeInputsEnd1").show();
// } else{
//     $("#menorIdadeInputs").hide();
//     $("#menorIdadeInputsEnd").hide();
//     $("#menorIdadeInputsEnd1").hide();
// }
$('input[name="rdMenorIdade"]').change(function () {

    if ($ (this).val() === "sim") {
        $("#menorIdadeInputs").show();
        $("#menorIdadeInputsEnd").show();
        $("#menorIdadeInputsEnd1").show();
    } else {
        $("#menorIdadeInputs").hide();
        $("#menorIdadeInputsEnd").hide();
        $("#menorIdadeInputsEnd1").hide();
    }
});
}

if (FORM_MODE == "VIEW") {

```

```

        $( "#btNovoItemNewProposta" ).addClass("fs-display-
none")
        $( "#btNovoItemseguradora" ).addClass("fs-display-
none")
        $( "#btNovoItemBancario" ).addClass("fs-display-
none")
        $(".remove-child").addClass("fs-display-none")
        $(".btnUpFile").addClass("fs-display-none")

    }

$(document).on('click', "input[name='rdMenorIdade']", 
function (e) {

    requiredFieldsRule();

    $("#nomeResponsavelMenor,#cpfResponsavelMenor,#dtNasResp
Menor,#parentescoMenor,#ruaRespMenor," +
"#bairroRespMenor,#numeroRespMenor,#cidadeRespMenor,#ufRespMe
nor,#telefoneRespMenor,#emailRespMenor," +
"#profissaoRespMenor,#rendaRespMenor").val("");

});

$("#rendaRespMenor").mask("#.##0,00", { reverse: true
});
$("#btn_incluirProposta").click(function () {
    var idx = wdkAddChild("tablePropostas");
});

$("#btn_incluirBeneficiario").click(function(){
    var idx = wdkAddChild("tableBeneficiarios");

    $("#idRow__"+idx).val("000" +
($(".classIdRow").length-1).toString().slice(-3));
    $("#benef__"+idx).val("");
    $("#nascBenef__"+idx).val("");
    $("#porcentagemBenef__"+idx).val("");
    $("#grauParent__"+idx).val("");
});

```

```

});;

//função para deletar linha da tabela pai x filho
function fnCustomDeleteBeneficiario(oElement) {
    FLUIGC.message.confirm({
        message: 'Deseja remover este beneficiário?',
        title: 'Removendo beneficiário',
        labelYes: 'Remover',
        labelNo: 'Cancelar'
    }, function (result, el, ev) {
        if(result){
            fnWdkRemoveChild(oElement);
            var seq = 1;
            $(".classIdRow").each(function (index) {
                var idx =
$(this).prop("id").split("__")[1];
                if(idx != undefined){
                    $("#idRow__"+idx).val("000" +
(seq).toString().slice(-3));
                    seq++;
                }
            });
        }
    });
}

//função para deletar linha da tabela pai x filho
function fnCustomDeleteProposta(oElement) {
    FLUIGC.message.confirm({
        message: 'Deseja remover esta proposta',
        title: 'Removendo proposta',
        labelYes: 'Remover',
        labelNo: 'Cancelar'
    }, function (result, el, ev) {
        if (result) {
            fnWdkRemoveChild(oElement);
        }
    });
}

function abrirModal() {
    let tabela = '<table id="tabelaDados" border="1"><tr><th>Nome do Fundo</th><th>Percentual de Aplicação</th><th>Remover</th></tr></table>';
}

```

```

        Swal.fire({
            title: 'Incluir Fundo',
            html: '<input id="nomeFundo" class="swal2-input" placeholder="Nome do Fundo">' +
                  '<input id="percentual" class="swal2-input" placeholder="Percentual de Aplicação">' +
                  tabela,
            showCancelButton: true,
            confirmButtonText: 'Adicionar Linha', // Mantendo o
botão de confirmação padrão
            preConfirm: () => {
                const nomeFundoInput =
                    Swal.getPopup().querySelector('#nomeFundo');
                const percentualInput =
                    Swal.getPopup().querySelector('#percentual');
                if (!nomeFundoInput || !percentualInput ||
                    !nomeFundoInput.value || !percentualInput.value) {
                    Swal.showValidationMessage('Por favor,
                    preencha todos os campos');
                    return false;
                }
                const novoDado = { nomeFundo:
                    nomeFundoInput.value, percentual: percentualInput.value };
                adicionarLinha(novoDado);
            }
        });
    }

function adicionarLinha(dado) {
    const tabela = document.getElementById('tabelaDados');
    const rowCount = tabela.rows.length;
    const newRow = tabela.insertRow(rowCount);
    const cell1 = newRow.insertCell(0);
    const cell2 = newRow.insertCell(1);
    const cell3 = newRow.insertCell(2);

    cell1.innerHTML = dado.nomeFundo;
    cell2.innerHTML = dado.percentual;
    cell3.innerHTML = '<button
onclick="removerLinha(this)">Remover</button>';

    let dadosAntigos = [];
    const txt_fundoJson =
        document.querySelectorAll('[name^="txt_fundoJson"]');
    txt_fundoJson.forEach(input => {
        if (input.value.trim() !== '') {
            dadosAntigos.push(JSON.parse(input.value));
    
```

```

        }
    });

    dadosAntigos.push(dado);
    txt_fundoJson.forEach((input, index) => {
        input.value = JSON.stringify(dadosAntigos[index]);
    });
}

function removerLinha(btn) {
    const row = btn.parentNode.parentNode;
    const rowIndex = row.rowIndex;
    row.parentNode.removeChild(row);

    let dadosAntigos = [];
    const txt_fundoJson =
        document.querySelectorAll('[name^="txt_fundoJson"]');
    txt_fundoJson.forEach(input => {
        if (input.value.trim() !== '') {
            dadosAntigos.push(JSON.parse(input.value));
        }
    });
    dadosAntigos.splice(rowIndex - 1, 1);
    txt_fundoJson.forEach((input, index) => {
        input.value = JSON.stringify(dadosAntigos[index]);
    });
}

function abrirModalFundo(elemento) {

    var index =
        $(elemento.nextElementSibling)[0].name.split("____")[1];

    var template = $('.tpl-fundos').html();
    var htmlFundos = Mustache.render(template);

    var modalFundo = FLUIGC.modal({
        title: 'Fundos',
        content: htmlFundos,
        id: 'fluig-modal-fundos',
        actions: [
            {
                'label': 'Salvar',
                'bind': 'data-salva-fundo',
            },
            {
                'label': 'Cancelar',
            }
        ]
    });
}

```

```

        autoClose: true
    });

}, function(err, data) {
    if(err) {
        // do error handling
    } else {
        $("#percentualFundo").mask("#00%", {reverse:
true});
        mydata = $("#jsonFundos__" + index).val() || '[]';

        var tableFundos = FLUIGC.datatable('#target', {
            dataRequest: JSON.parse(mydata),
            renderContent: ['fundo', 'percentual'],
            header: [
                {'title': 'Fundo'},
                {'title': 'Percentual'}
            ],
            search: {
                enabled: false,
            },
            classSelected: 'success',
            emptyMessage: '<div class="text-center">Nenhum
            dado para ser exibido.</div>',
        }, function(err, data) {
            // DO SOMETHING (error or success)
        });
    }
}

$("#data-salva-fundo").on("click", function() {
    console.log(JSON.stringify(tableFundos.getData()));
    var fundosData = tableFundos.getData();

    $("#jsonFundos__"+index).val(JSON.stringify(tableFundos
    .getData()));
    modalFundo.remove();

    updateTableFundo(fundosData, index);
});

$("#adicionarFundo").on("click", function() {
    var fundosData = tableFundos.getData();
    var row = {
        fundo: $("#nomeFundo").val(),
        percentual: $("#percentualFundo").val()
    };
    tableFundos.addRow(0, row);
});

```

```

        $( "#nomeFundo, #percentualFundo" ).val("");
        updateTableFundo( fundosData, index );
    });

    $( "#editarFundo" ).on("click", function() {
        var fundosData = tableFundos.getData();
        var itemSelecionado =
tableFundos.selectedRows();

        $( "#nomeFundo" ).val( tableFundos.getData()[itemSelecionado].fu-
ndo);

        $( "#percentualFundo" ).val( tableFundos.getData()[itemSeleciona-
do].percentual);

        tableFundos.removeRow(itemSelecionado);
        updateTableFundo( fundosData, index );
    });

    $( "#excluirFundo" ).on("click", function() {
        var fundosData = tableFundos.getData();
        var itemSelecionado =
tableFundos.selectedRows();
        tableFundos.removeRow(itemSelecionado);
        updateTableFundo( fundosData, index );
    });
});

}

function updateTableFundo( fundosData, index ) {
    var tableHtml = '<table class="table">';
    tableHtml += '<thead><tr><th>Fundo</th><th>Percentual</th></tr></thead>';
    tableHtml += '<tbody>';
    fundosData.forEach(function(row) {
        tableHtml += '<tr><td>' + row.fundo + '</td><td>' +
row.percentual + '</td></tr>';
    });
    tableHtml += '</tbody></table>';

    $( "#tableFundos___" + index ).html(tableHtml);
}

function abrirModalBeneficiario(elemento) {

```

```

    var index =
$(element.nextElementSibling)[0].name.split("____")[1];

var template = $('.tpl-beneficiario').html();
var htmlFundos = Mustache.render(template);

var modalFundo = FLUIGC.modal({
    title: 'Beneficiário',
    content: htmlFundos,
    id: 'fluig-modal-fundos',
    actions: [
        {
            'label': 'Salvar',
            'bind': 'data-salva-fundo',
        },
        {
            'label': 'Cancelar',
            autoClose: true
        }
    ],
    function(err, data) {
        if(err) {
            // do error handling
        } else {

            $("#nascimento").mask("00/00/0000");
            $("#porcentagem").mask("#00%", {reverse:
true});
            mydata = $("#beneficiarioJson____" + index).val() ||
'[]';

            var tableBen = FLUIGC.datatable('#target', {
                dataRequest: JSON.parse(mydata),
                renderContent: ['nomeBen',
'nascimento','porcentagem','parentesco'],
                header: [
                    {'title': 'Nome'},
                    {'title': 'Nascimento'},
                    {'title': '%'},
                    {'title': 'Grau de Parentesco'},
                ],
                search: [
                    enabled: false,
                ],
                classSelected: 'success',
                emptyMessage: '<div class="text-center">Nenhum
dado para ser exibido.</div>',
            }, function(err, data) {

```

```

        });

        $('[data-salva-fundo]').on("click", function() {
            var beneficiariosData =
tableBen.getData();

            console.log(JSON.stringify(tableBen.getData()));

            $('#beneficiarioJson__'+index).val(JSON.stringify(table
Ben.getData()));
            modalFundo.remove();
        });

        updateBeneficiarioTable(beneficiariosData, index);

    });

    $('#adicionarFundo').on("click", function() {
        var beneficiariosData =
tableBen.getData();
        var row = {
            nomeBen: $('#nomeBen').val(),
            nascimento: $('#nascimento').val(),
            porcentagem:
$('#porcentagem').val(),
            parentesco: $('#parentesco').val()
        };
        tableBen.addRow(0, row);
        $('#nomeBen, #nascimento', "#porcentagem"
, "#parentesco").val("");
        updateBeneficiarioTable(beneficiariosData, index);
    });

    $('#editarFundo').on("click", function() {
        var beneficiariosData =
tableBen.getData();
        var itemSelecionado = tableBen.selectedRows();

        $('#nomeBen').val(tableBen.getData()[itemSelecionado].nomeBen
);
        $('#nascimento').val(tableBen.getData()[itemSelecionado].nasc
imento);
    });
}

```

```

        $("#porcentagem").val(tableBen.getData()[itemSelecionado].porcentagem);

        $("#parentesco").val(tableBen.getData()[itemSelecionado].parentesco);

        tableBen.removeRow(itemSelecionado);

        updateBeneficiarioTable(beneficiariosData, index);
    });
    $("#excluirFundo").on("click", function() {
        var beneficiariosData =
            tableBen.getData();
        var itemSelecionado = tableBen.selectedRows();

        tableBen.removeRow(itemSelecionado);

        updateBeneficiarioTable(beneficiariosData, index);
    });

    });
});

function updateBeneficiarioTable(beneficiarioData, index) {
    var tableHtml = '<table class="table">';
    tableHtml += '<thead><tr><th>Nome</th><th>Nascimento</th><th>%</th><th>Gra-
u de Parentesco</th></tr></thead>';
    tableHtml += '<tbody>';
    beneficiarioData.forEach(function(row) {
        tableHtml += '<tr><td>' + row.nomeBen + '</td><td>' +
        row.nascimento + '</td><td>' + row.porcentagem + '</td><td>' +
        row.parentesco + '</td></tr>';
    });
    tableHtml += '</tbody></table>';

    $("#tableBene__" + index).html(tableHtml);
}

function loadJsonDataText() {
    $('[id^="jsonFundos__"]").each(function() {
        var index = $(this).attr("id").split("__")[1];
        var jsonData = $(this).val();

```

```

    if (jsonData && jsonData.trim() !== '') {
        var fundosData = JSON.parse(jsonData);
        var fundosHtml = '<table class="table">';
        fundosHtml +=
            '<thead><tr><th>Fundo</th><th>Percentual</th></tr></thead>';
        fundosHtml += '<tbody>';
        fundosData.forEach(function(row) {
            fundosHtml += '<tr><td>' + row.fundo +
            '</td><td>' + row.percentual + '</td></tr>';
        });
        fundosHtml += '</tbody></table>';
        $(fundosHtml).insertAfter($('#controleFundos___'+index));
    }
});

$("[id^='beneficiarioJson___']").each(function()) {
    var index = $(this).attr("id").split("___")[1];
    var jsonData = $(this).val();
    if (jsonData && jsonData.trim() !== '') {
        var beneficiarioData = JSON.parse(jsonData);
        var beneficiarioHtml = '<table class="table">';
        beneficiarioHtml +=
            '<thead><tr><th>Nome</th><th>Nascimento</th><th>%</th><th>Gra-
            u de Parentesco</th></tr></thead>';
        beneficiarioHtml += '<tbody>';
        beneficiarioData.forEach(function(row) {
            beneficiarioHtml += '<tr><td>' + row.nomeBen +
            '</td><td>' + row.nascimento + '</td><td>' +
            row.porcentagem + '</td><td>' + row.parentesco +
            '</td></tr>';
        });
        beneficiarioHtml += '</tbody></table>';
    }
    $(beneficiarioHtml).insertAfter($('#controleBen___'+ index));
}
);

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 retornaIndices(tabela) {
    var indices = [];

    var linhas = $("table[tablename='" + tabela + "'] tbody tr").filter(function() {
        return $(this).find('input').length > 0;
    });

    for (var i = 1; i < linhas.length; i++) {
        var campo =
        $(linhas[i]).find("input")[0].getAttribute("name").split("__")
        [1];
        indices.push(campo);
    }
    return indices;
}

function validacaoJson(str) {
    try {
        const json = JSON.parse(str);
        return Array.isArray(json) && json.length >= 1 &&
typeof json[0] === 'object';
    } catch (e) {
        return false;
    }
}

function calendarioCompetencia(index) {

    const picker = new Pikaday({
        field:
document.getElementById('txt_seguradoraMesComp__'+index),
        format: 'MM/YYYY',
        // Desabilita a seleção de datas
        disableDayPicker: true,
        // Mostra apenas os meses
        showMonths: true,
        // Mostra os anos
        showYears: true,
        // Define o idioma para português
    });
}

```

```

        i18n: {
            previousMonth: 'Mês anterior',
            nextMonth: 'Próximo mês',
            months: ['Janeiro', 'Fevereiro', 'Março',
'Abril', 'Maio', 'Junho', 'Julho', 'Agosto', 'Setembro',
'Outubro', 'Novembro', 'Dezembro'],
            weekdays: ['Domingo', 'Segunda', 'Terça',
'Quarta', 'Quinta', 'Sexta', 'Sábado'],
            weekdaysShort: ['Dom', 'Seg', 'Ter', 'Qua',
'Qui', 'Sex', 'Sáb']
        }
    });
}

function validarCompetencia(data) {
    // Expressão regular para verificar o formato MM/AAAA
    const regex = /^(0[1-9]|1[0-2])\//\d{4}$/;

    // Retorna true se a data for válida, false caso
    contrário
    return regex.test(data);
}

function exibeSaldoOuComissao() {

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

        $(".saldo").removeClass("fs-display-none")
        $(".comissao").addClass("fs-display-none")
        $(".resgate").addClass("fs-display-none")

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

        $(".saldo").addClass("fs-display-none")
        $(".comissao").removeClass("fs-display-none")

        $(".resgate").addClass("fs-display-none")

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

        $(".saldo").addClass("fs-display-none")
        $(".comissao").addClass("fs-display-none")
        $(".resgate").removeClass("fs-display-none")
    }
}

```

```

        )else{
            $(".saldo").addClass("fs-display-none")
            $(".comissao").addClass("fs-display-none")
            $(".resgate").addClass("fs-display-none")
        }

    }

function limparCamposSaldoEComissao(){
    var indices = retornaIndices("dadositensNewProposta");
    if(indices.length > 0){

        for (var i in indices){

            $("#txt_seguradoraSaldo____"+indices[i]).val("")
            $("#txt_seguradoraRentabilidade____"+indices[i]).val("")

            $("#txt_seguradoraComissao____"+indices[i]).val("")
                $("#txt_cdNatureza____"+indices[i]).val("")

                window["txt_natureza____"+indices[i]].clear()

        }
    }

}

function calcularSomaComissao(){

    clearTimeout($ this).data('timeout');
    $ this.data('timeout', setTimeout(function() {
        var total = 0;

        $("input[id^='txt_seguradoraComissao____']").each(function() {

            var valor = $ this.val();
            valor = parseFloat(valor.replace /\./g,
            '').replace(',', '.')) || 0;
            total += valor;
        })
    }, 100));
}

```

```

        });

        var totalFormatado = total.toLocaleString('pt-BR',
        {
            minimumFractionDigits: 2,
            maximumFractionDigits: 2
        });

    });

    $("#valorTotalComissao").val(totalFormatado.replace('.','',
',')).delay(300);

}

function calcularSoma() {
    clearTimeout($("#this").data('timeout'));
    $("#this").data('timeout', setTimeout(function() {
        var total = 0;

        $("input[id^='txt_seguradoraSaldo_']").each(function() {
            var valor = $("#this").val();
            valor = parseFloat(valor.replace(/\./g,
'')).replace(',', '.')) || 0;
            total += valor;
        });

        var totalFormatado = total.toLocaleString('pt-BR',
        {
            minimumFractionDigits: 2,
            maximumFractionDigits: 2
        });

        $("#valorTotal").val(totalFormatado.replace('.','',
',')).delay(300));
    }));
}

function linhasDuplicadasSaldoComissao(arr) {
    var counts = {};
    for (var i = 0; i < arr.length; i++) {
        if (counts[arr[i]] === undefined) {
            counts[arr[i]] = 1;
        } else {
    
```

```

        return true;
    }
}

return false;
}

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

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

        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){

                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();
                }
            },
            error : function(err){
                console.log("Não foi possível recuperar os
dados: ", err);
            }
        });
    });
}

```

```
        FLUIGC.toast({message : "Não foi possível  
recuperar os dados", type: "danger"});  
        loading.hide();  
    })  
});  
});  
  
function preencheCamposClientes (retornoDataset) {  
    const [registro] = retornoDataset;  
  
    const {  
        A1COD: alcod,  
        A1NOME: alnome,  
        A1LOJA: alloja,  
        A1CNAE: alcnae,  
        A1CEP: cep,  
        A1END: rua,  
        A1COMPLEM: numero,  
        A1PESSOA: alpessoa,  
        A1EST: uf,  
        A1ESTNOM: estado,  
        A1CODMUN: codCidade,  
        A1MUN: cidade,  
        A1BAIRRO: bairro,  
        A1EMAIL: email,  
        A1DTNAS: dataNascimento,  
        A1DDD: ddd,  
        A1DDI: ddi,  
        A1TEL: telefone,  
  
        A1XIBAN: iban,  
        A1XBAN: bancoIdent,  
        A1XSPREAD: spreadAcordado,  
        A1XRESPON: responsavel,  
        A1XTELEXT: telefoneExterior,  
        A1XNOMSOC: socioMarjoritario,  
        A1XCPFMAJ: cpfMarjoritario,  
        A1XENDSOC: enderecoSocio,  
        A1XENDEXT: enderecoExt,  
        A1XORISOL: orgemSolic,  
        A1XNIFIID: nifIcamaIdEstrangeiro,  
  
        A1XCONJ: conjugue,  
    } = registro;  
}
```

```

        A1XNACONJ: dataNascConjuge,
        A1XCPFCON: cpfConjuge,
        A1_XSAIDEF: saidaDefinitiva,
        A1XINCTR: dataInicioCt,
        A1XFIMCTR: dataFimCt,
        A1XVENCRG: dataVencRg,
        A1XVENPAS: dataVencPassaporte
    ) = registro;

    $("#iban").val(iban);
    $("#bancoIdent").val(bancoIdent);
    $("#spreadAcordado").val(spreadAcordado);
    $("#responsavel").val(responsavel);
    $("#telefoneExterior").val(telefoneExterior);
    $("#socioMarjoritario").val(socioMarjoritario);
    $("#cpfMarjoritario").val(cpfMarjoritario);
    $("#enderecoSocio").val(enderecoSocio);
    $("#enderecoExt").val(enderecoExt);
    $("#orgemSolic").val(orgemSolic);
    $("#nifIcamaIdEstrangeiro").val(nifIcamaIdEstrangeiro);

    $("#nomeConjuge").val(conjuge);

    $("#dtNasConjuge").val(formatarDataAAAAMMDDParaDDMMYYYY(dataN
ascConjuge));
    $("#cpfConjuge").val(cpfConjuge);
    //
    $("#saidaDefinitiva").val(formatarDataAAAAMMDDParaDDMMYYYY(sa
idaDefinitiva));

    $("#dataInicioCt").val(formatarDataAAAAMMDDParaDDMMYYYY(dataI
nicioCt));

    $("#dataFimCt").val(formatarDataAAAAMMDDParaDDMMYYYY(dataFimC
t));

    $("#a1cod").val(a1cod);
    $("#alloja").val(alloja);
    $("#cep").val(cep);
    $("#rua").val(rua);
    $("#numero").val(numero);
    $("#alpessoa").val(alpessoa);

    $("#dataVencRg").val(formatarDataAAAAMMDDParaDDMMYYYY(dataVen
cRg));

```

```

$( "#dataVencPassaporte" ).val( formatarDataAAAAMMDDParaDDMMYYYY
(dataVencPassaporte));

        window[ 'uf' ].setValue({ "X5_CHAVE":uf,
"X5_DESCRI":estado});
        window[ 'cidade' ].setValue({ "CC2_MUN": cidade,
"CC2_CODMUN": codCidade});

        $("#bairro").val(bairro);
        $("#email").val(email);
        $("#telefoneBrasil").val(telefone);
        $("#numero").val(numero);

$( "#dataNascimento" ).val( formatarDataAAAAMMDDParaDDMMYYYY( dataNascimento));
}

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

    // Verificar se é CPF (11 dígitos) ou CNPJ (14 dígitos)
    if (numeroLimpo.length === 11) {
        // Aplicar máscara de CPF: 999.999.999-99
        return
    numeroLimpo.replace(/(\d{3})(\d{3})(\d{3})(\d{2})/, '$1.$2.$3-$4');
    } else if (numeroLimpo.length === 14) {
        // Aplicar máscara de CNPJ: 99.999.999/9999-99
        return
    numeroLimpo.replace(/(\d{2})(\d{3})(\d{3})(\d{4})(\d{2})/, '$1.$2.$3/$4-$5');
    } else {
        // Se o número não corresponder a um CPF ou CNPJ
        // válido, retornar o número original
        return numero;
    }
}

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;
    }
}

function calcularSomaResgate() {

    clearTimeout($ this).data('timeout'));
    $ this.data('timeout', setTimeout(function() {
        var total = 0;

        $("input[id^='txt_seguradoraResgate__']").each(function() {

            var valor = $ this.val();
            valor = parseFloat(valor.replace(/\./g,
                '').replace(',', '.')) || 0;
            total += valor;
        });

        var totalFormatado = total.toLocaleString('pt-BR',
        {

```

```
        minimumFractionDigits: 2,
        maximumFractionDigits: 2
    }) ;

$( "#valorTotalResgate" ).val(totalFormatado.replace('.',',',1));
});
```