

```

alfabeto = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',
'j', 'k', 'l', 'm',
'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
'x', 'y', 'z' ];

function servicetask11(attempt, message) {

    var indicesSaldoComissao =
retornaIndices("txt_cdSeguradora");
    var indicesProposta =
retornaIndices("txt_seguradoraProp");
    log.info("### rdTipoInclusao: " +
hAPI.getCardValue("rdTipoInclusao"))
    if (hAPI.getCardValue("rdTipoInclusao") == "proposta") {

        for ( var i in indicesProposta) {

            incluiAtualizaRegistroPropostaFluig(indicesProposta[i])
        }

    } else if (hAPI.getCardValue("rdTipoInclusao") == "saldo") {

        for ( var i in indicesSaldoComissao) {

            incluiAtualizaRegistroFluig(indicesSaldoComissao[i],
"saldo")
        }

    } else if (hAPI.getCardValue("rdTipoInclusao") == "resgate") {

        for ( var i in indicesSaldoComissao) {

            incluiAtualizaRegistroFluig(indicesSaldoComissao[i],
"resgate")
        }

    } else if (hAPI.getCardValue("rdTipoInclusao") == "comissao") {

        var contador = 0;

        for ( var i in indicesSaldoComissao) {

            if (hAPI.getCardValue("retornoProtheus__" +
indicesSaldoComissao[i]) != "OK")

```

```

        incluiTituloProtheus(indicesSaldoComissao[i], contador)

        contador++;
    }

}

}

function incluiTituloProtheus(indice, contador) {

    try {
        var clientService =
fluigAPI.getAuthorizeClientService();

        var codigoTitulo = preencherZerosEsquerda(
            parseInt(getValue("WKNNumProces")) + "" +
alfabeto[contador], 7);

        var data = {
            companyId : '1',
            serviceCode : "rest-protheus",
            endpoint : '/TITREC',
            method : 'post',
            timeoutService : '100',
            headers : [
                "Accept-Charset" : "UTF-8",
                "Content-Type" : "application/json",
            ],
            params : [
                "EMPRESA" : "" +
hAPI.getCardValue("cdEmpresaProtheus"),
                "FILIAL" : "" +
hAPI.getCardValue("cdFilialProtheus"),
                "E1_PREFIXO" : "PRV",
                "E1_NUM" : "" + codigoTitulo,
                "E1_PARCELA" : "A",
                "E1_TIPO" : "BOL",
                "E1_NATUREZ" :
hAPI.getCardValue("txt_cdNatureza___" + indice),
                "E1_CLIENTE" :
hAPI.getCardValue("alcod"),
                "E1_LOJA" : hAPI.getCardValue("alloja"),
                "E1_EMISSAO" : dataFormatada(hAPI
                    .getCardValue("txt_dataEmissao")),
            ]
        }
    }
}

```

```

"E1_VENCTO" : dataFormatada(hAPI

.getCardValue("txt_seguradoraVencimento____" + indice)),
"E1_VALOR" :
parseFloat(hAPI.getCardValue(
                           "txt_seguradoraComissao____" +
indice).replaceAll("\\.",
                           "").replaceAll(", ", ".")),
"E1_HIST" :
hAPI.getCardValue("txtHistoricoComissao____"
                  + indice),
"E1_MOEDA" : 1,
"E1_TXMOEDA" : 1.0,
"E1_XNOMSOL" : "",
"E1_XEMAILS" : "",
"E1_XNOME" : "",
"E1_XCPF" : new java.math.BigInteger(hAPI
                                         .getCardValue("zoomcpfCnpj")),
"E1_XTPOPER" : "",
"E1_XDTFCH" : "",
"E1_XMOEDA" : "1",
"E1_XMOEDEX" : "",
"E1_XCONTRA" : "",
"E1_XTXCLI" : parseInt(0),
"E1_XTXSPOT" : parseInt(0),
"E1_XDESPBC" : parseInt(0),
"E1_XIOF" : parseInt(0),
"E1_XVLRTOT" : parseInt(0),
"E1_XSPREAD" : parseInt(0),
"E1_XVLRSP" : parseInt(0),
"E1_XCOMIS" : parseInt(0),
"E1_XOBSENV" : "Mês de Competência: "
                  + hAPI

.getCardValue("txt_seguradoraMesComp____"
                  + indice),
"E1_XOPER" : ""
+
hAPI.getCardValue("txt_cdSeguradora____" + indice)
}
};

// log.dir(data);
log.info("JSON de integração " +
JSONUtil.toJSONString(data));

```

```

        var retornoApi =
clientService.invoke(JSONUtil.toJSON(data));
        log.info("retornoApi.getResult() " +
retornoApi.getResult());

        var retornoApiGetResult = "" +
retornoApi.getResult();
        retornoApiGetResult = retornoApiGetResult.replace(
/[^\u0000-\u001F\u007F-\u009F]/g, '')

        var retornojson = JSON.parse(retornoApiGetResult);

        var statusHttp = retornoApi.getHttpStatusResult();

        // VERIFICA SE O RETORNO ESTA VAZIO
        if (retornoApi.getResult() == null ||
retornoApi.getResult().isEmpty())
            throw java.lang.Exception("Ocorreu uma falha
no retorno da API!");

        if (statusHttp >= "400") {
            throw retornoApi.getResult();
        }

        hAPI.setCardValue("retornoProtheus____" + indice,
retornojson.Mensagem);

        if (retornojson.Mensagem == "OK") {
            hAPI.setCardValue("codigoTitulo____" + indice,
codigoTitulo);
        }

        if (retornojson.Mensagem != "OK") {
            throw retornojson["Mensagem Detalhada"];
        }

    } catch (err) {
        // throw err +" - "+ retornoApi.getResult()+" - "+
JSON.stringify(data);
        log.error(err);
        throw err + " - Linha: " + err.lineNumber;
    }
}

function incluiAtualizaRegistroFluig(indice, rdTipoInclusao)
{

```

```

var cpfCnpj = hAPI.getCardValue("zoomcpfCnpj")
var zoomNomeCompleto =
hAPI.getCardValue("zoomNomeCompleto")
var txt_cdSeguradora =
hAPI.getCardValue("txt_cdSeguradora____" + indice)
var txt_seguradora =
hAPI.getCardValue("txt_seguradora____" + indice)
var txt_seguradoraMesComp =
hAPI.getCardValue("txt_seguradoraMesComp____"
+ indice)

var txt_seguradoraSaldo =
hAPI.getCardValue("txt_seguradoraSaldo____"
+ indice)
var txt_seguradoraRentabilidade = hAPI
.getCardValue("txt_seguradoraRentabilidade____"
+ indice)
var txt_seguradoraResgate =
hAPI.getCardValue("txt_seguradoraResgate____"
+ indice)
var txt_propostaSeguradora =
hAPI.getCardValue("txt_propostaSeguradora____"
+ indice)

var dataCadastro = retornaData();

var txt_investEspiradico =
hAPI.getCardValue("txt_investEspiradico____"
+ indice)
var txt_investMensal =
hAPI.getCardValue("txt_investMensal____" + indice)

if (rdTipoInclusao == "saldo") {

    var retornoRegistroExistente =
consultaRegistroSaldoSeguradoraPorCliente(
            cpfCnpj, txt_cdSeguradora,
            txt_seguradoraMesComp,
            txt_propostaSeguradora);
    log.info("## retornoRegistroExistente: " +
retornoRegistroExistente)
    var retornoIncluiAtualizaRegistro = "";

        retornoIncluiAtualizaRegistro =
incluirRegistroFluig(cpfCnpj,

```

```

                zoomNomeCompleto, txt_cdSeguradora,
txt_seguradora,
                    txt_seguradoraMesComp,
txt_seguradoraSaldo,
                        txt_seguradoraRentabilidade,
dataCadastro, rdTipoInclusao,
                            txt_seguradoraResgate,
txt_propostaSeguradora,
                                txt_investEspiradico, txt_investMensal);

/*if (retornoRegistroExistente != "") {
    retornoIncluiAtualizaRegistro =
atualizaRegistroFluig(
    zoomNomeCompleto,
        txt_cdSeguradora, txt_seguradora,
    txt_seguradoraMesComp,
        txt_seguradoraSaldo,
    txt_seguradoraRentabilidade,
        dataCadastro, rdTipoInclusao,
    txt_propostaSeguradora,
        txt_investEspiradico,
    txt_investMensal)
} else {
    retornoIncluiAtualizaRegistro =
incluiRegistroFluig(cpfCnpj,
    zoomNomeCompleto, txt_cdSeguradora,
    txt_seguradora,
        txt_seguradoraMesComp,
    txt_seguradoraSaldo,
        txt_seguradoraRentabilidade,
    dataCadastro, rdTipoInclusao,
            txt_seguradoraResgate,
    txt_propostaSeguradora,
        txt_investEspiradico,
    txt_investMensal)
} */
}

} else {

    retornoIncluiAtualizaRegistro =
incluiRegistroFluig(cpfCnpj,
    zoomNomeCompleto, txt_cdSeguradora,
txt_seguradora,
        txt_seguradoraMesComp,
txt_seguradoraSaldo,

```

```

                txt_seguradoraRentabilidade,
dataCadastro, rdTipoInclusao,
                txt_seguradoraResgate,
txt_propostaSeguradora)

}

}

/*******
*****
*/
function incluiAtualizaRegistroPropostaFluig(indice) {

    var cpfCnpj = hAPI.getCardValue("zoomcpfCnpj")
    var zoomNomeCompleto =
hAPI.getCardValue("zoomNomeCompleto")
    var txt_cdSeguradora = hAPI
        .getCardValue("txt_cdSeguradoraProp__" + indice)
    var txt_seguradora =
hAPI.getCardValue("txt_seguradoraProp__" + indice)
    var txt_NewProposta =
hAPI.getCardValue("txt_NewProposta__" + indice)

    var mensalidade = "";
    var aporte = ""

    if (hAPI.getCardValue("txt_mensAportNew__" + indice) == "aporte") {
        aporte =
hAPI.getCardValue("txt_inptMensalAportNew__" + indice)
    } else {
        mensalidade =
hAPI.getCardValue("txt_inptMensalAportNew__" + indice)
    }

    var txt_proReg = hAPI.getCardValue("txt_proReg__" + indice)
    var txt_Vgb1PgblNew =
hAPI.getCardValue("txt_Vgb1PgblNew__" + indice)

    var dataCadastro = retornaData();

    var jsonFundos = hAPI.getCardValue("jsonFundos__" + indice)
    log.info("### jsonFundos:" + jsonFundos);
}

```

```

var jsonFundosParse = JSON.parse(jsonFundos);
log.info("### jsonFundosParse: " +
jsonFundosParse.length);

var retornoRegistroExistente =
consultaRegistroPropostaSeguradoraPorCliente(
    cpfCnpj, txt_cdSeguradora, txt_NewProposta)
log.info("## retornoRegistroExistente: " +
retornoRegistroExistente)
var retornoIncluiAtualizaRegistro = "";

for var i = 0; i < jsonFundosParse.length; i++) {
    var fundo = jsonFundosParse[i].fundo;
    var percentual = jsonFundosParse[i].percentual;

        // Chama diretamente a inclusão para garantir o
histórico
        retornoIncluiAtualizaRegistro =
incluirRegistroPropostaFluig(cpfCnpj,
    zoomNomeCompleto, txt_cdSeguradora,
    txt_seguradora,
    txt_NewProposta, mensalidade, aporte,
    dataCadastro, txt_proReg,
    txt_VgblPgblNew, fundo, percentual);

/*if (retornoRegistroExistente != "") {
    retornoIncluiAtualizaRegistro =
atualizaRegistroPropostaFluig(
    retornoRegistroExistente, cpfCnpj,
    zoomNomeCompleto,
    txt_cdSeguradora, txt_seguradora,
    mensalidade, aporte, dataCadastro,
    txt_VgblPgblNew, fundo, percentual)
} else {
    retornoIncluiAtualizaRegistro =
incluirRegistroPropostaFluig(
    cpfCnpj, zoomNomeCompleto,
    txt_cdSeguradora, txt_NewProposta,
    mensalidade, aporte,
    txt_VgblPgblNew, fundo,
    percentual)
} */

```

```

        }

    }

function atualizaRegistroFluig(retornoRegistroExistente,
cpfCnpj, nomeCompleto,
        txt_cdSeguradora, txt_seguradora,
txt_seguradoraMesComp,
        txt_seguradoraSaldo, txt_seguradoraRentabilidade,
dataCadastro,
        rdTipoInclusao, txt_propostaSeguradora,
txt_investEspiradico,
        txt_investMensal) {
    log.info("## atualizaRegistroFluig ###")
try {

    var config = recuperaDadosFixos();

    var serviceProvider = ServiceManager
        .getServiceInstance("ECMCardservice"); //  

codigo do serviço
    var wkServiceEngine = serviceProvider

    .instantiate("com.totvs.ECMCardServiceService");
    var service = wkServiceEngine.getCardServicePort();

    var cardFieldDtoArray = serviceProvider

    .instantiate("com.totvs.CardFieldDtoArray");

    cardFieldDto1 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
    cardFieldDto1.setField("cpfCnpj");
    cardFieldDto1.setValue(cpfCnpj);

    cardFieldDto2 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
    cardFieldDto2.setField("nomeCompleto");
    cardFieldDto2.setValue(nomeCompleto);

    cardFieldDto3 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
    cardFieldDto3.setField("txt_cdSeguradora");
    cardFieldDto3.setValue(txt_cdSeguradora);

    cardFieldDto4 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
}

```

```

        cardFieldDto4.setField("txt_seguradora");
        cardFieldDto4.setValue(txt_seguradora);

        cardFieldDto5 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto5.setField("txt_seguradoraMesComp");
        cardFieldDto5.setValue(txt_seguradoraMesComp);

        cardFieldDto6 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto6.setField("txt_seguradoraSaldo");
        cardFieldDto6.setValue(txt_seguradoraSaldo);

        cardFieldDto7 =
serviceProvider.instantiate("com.totvs.CardFieldDto");

        cardFieldDto7.setField("txt_seguradoraRentabilidade");
        cardFieldDto7.setValue(txt_seguradoraRentabilidade);

        cardFieldDto8 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto8.setField("campoDescriptor");
        cardFieldDto8.setValue(nomeCompleto + " - " +
txt_seguradora + " - "
                           + txt_propostaSeguradora + " - " +
txt_seguradoraMesComp);

        cardFieldDto9 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto9.setField("dataCadastro");

        cardFieldDto9.setValue(formataDataParaYYYYMMDD(dataCadastro));

        cardFieldDto10 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto10.setField("rdTipoInclusao");
        cardFieldDto10.setValue(rdTipoInclusao);

        cardFieldDto11 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto11.setField("txt_propostaSeguradora");
        cardFieldDto11.setValue(txt_propostaSeguradora);

        cardFieldDto12 =
serviceProvider.instantiate("com.totvs.CardFieldDto");

```

```

        cardFieldDto12.setField("txt_investEsporadico");
        cardFieldDto12.setValue(txt_investEsporadico);

        cardFieldDto13 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto13.setField("txt_investMensal");
        cardFieldDto13.setValue(txt_investMensal);

        cardFieldDtoArray.getItem().add(0, cardFieldDto1);
        cardFieldDtoArray.getItem().add(1, cardFieldDto2);
        cardFieldDtoArray.getItem().add(2, cardFieldDto3);
        cardFieldDtoArray.getItem().add(3, cardFieldDto4);
        cardFieldDtoArray.getItem().add(4, cardFieldDto5);
        cardFieldDtoArray.getItem().add(5, cardFieldDto6);
        cardFieldDtoArray.getItem().add(6, cardFieldDto7);
        cardFieldDtoArray.getItem().add(7, cardFieldDto8);
        cardFieldDtoArray.getItem().add(8, cardFieldDto9);
        cardFieldDtoArray.getItem().add(9, cardFieldDto10);
        cardFieldDtoArray.getItem().add(10,
cardFieldDto11);
            cardFieldDtoArray.getItem().add(11,
cardFieldDto12);
            cardFieldDtoArray.getItem().add(12,
cardFieldDto13);

        var result = service.updateCardData(1, "" +
config.usuarioFluig, ""
                + config.senhaFluig,
parseInt(retornoRegistroExistente),
                cardFieldDtoArray)

        log.info("#result =====> "
                + result.getItem().get(0).documentId);
        var docId = result.getItem().get(0).documentId;
        if (docId > 0) {
            return docId;
        } else {
            return "nok"
        log
            .error("Erro para salvar registro de
saldo x seguradora por cliente..." +
result.getItem().get(0).webServiceMessage);
    }

} catch (e) {
    return "nok"
}

```

```

        log
            .error("Erro para salvar registro de
saldo x seguradora por cliente..." +
e.toString());
    }

}

function incluiRegistroFluig(cpfcnpj, nomeCompleto,
txt_cdSeguradora,
    txt_seguradora, txt_seguradoraMesComp,
txt_seguradoraSaldo,
    txt_seguradoraRentabilidade, dataCadastro,
rdTipoInclusao,
    txt_seguradoraResgate, txt_propostaSeguradora,
txt_investEsporadico,
    txt_investMensal) {

try {

    var config = recuperaDadosFixos();

    var serviceProvider = ServiceManager
        .getServiceInstance("ECMCardservice"); //  

codigo do serviço
    var wkServiceEngine = serviceProvider

    .instantiate("com.totvs.ECMCardServiceService");
    var service = wkServiceEngine.getCardServicePort();
    var cardDtos =
serviceProvider.instantiate("com.totvs.CardDtoArray");
    var cardDto =
serviceProvider.instantiate("com.totvs.CardDto");
    cardDto.setParentDocumentId(parseInt("" +
        + recuperarIdFormPaiSaldoSeguradora()));
    cardDto.setColleagueId("'" + config.usuarioFluig);
    cardDto.setInheritSecurity(true);
    var cardData = serviceProvider

    .instantiate("com.totvs.CardFieldDtoArray");

    cardDto.getCardData().add(0,
        serviceProvider.instantiate("com.totvs.CardFieldDto"));
    cardDto.getCardData().get(0).setField("cpfcnpj");
    cardDto.getCardData().get(0).setValue(cpfcnpj);
}
}
```

```
        cardDto.getCardData().add(1,  
  
        serviceProvider.instantiate("com.totvs.CardFieldDto"));  
        cardDto.getCardData().get(1).setField("nomeCompleto");  
        cardDto.getCardData().get(1).setValue(nomeCompleto);  
  
        cardDto.getCardData().add(2,  
  
        serviceProvider.instantiate("com.totvs.CardFieldDto"));  
        cardDto.getCardData().get(2).setField("txt_cdSeguradora");  
    );  
  
    cardDto.getCardData().get(2).setValue(txt_cdSeguradora);  
  
    cardDto.getCardData().add(3,  
  
        serviceProvider.instantiate("com.totvs.CardFieldDto"));  
        cardDto.getCardData().get(3).setField("txt_seguradora");  
        cardDto.getCardData().get(3).setValue(txt_seguradora);  
  
    cardDto.getCardData().add(4,  
  
        serviceProvider.instantiate("com.totvs.CardFieldDto"));  
        cardDto.getCardData().get(4).setField("txt_seguradoraMesComp");  
        cardDto.getCardData().get(4).setValue(txt_seguradoraMesComp);  
  
    cardDto.getCardData().add(5,  
  
        serviceProvider.instantiate("com.totvs.CardFieldDto"));  
        cardDto.getCardData().get(5).setField("campoDescriptor");  
        cardDto.getCardData().get(5).setValue(  
            nomeCompleto + " - " + txt_seguradora + "  
- "  
            + txt_propostaSeguradora + " - "  
"  
            + txt_seguradoraMesComp);
```

```
        cardDto.getCardData().add(6,
        serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(6).setField("dataCadastro");
        cardDto.getCardData().get(6).setValue(
            formataDataParaYYYYMMDD(dataCadastro));

        cardDto.getCardData().add(7,
        serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(7).setField("rdTipoInclusao");
        cardDto.getCardData().get(7).setValue(rdTipoInclusao);

        cardDto.getCardData().add(8,
        serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(8).setField("txt_propostaSeguradora");
        cardDto.getCardData().get(8).setValue(txt_propostaSeguradora);

        if (rdTipoInclusao == "saldo") {
            cardDto.getCardData().add(9,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

            cardDto.getCardData().get(9).setField("txt_seguradoraSaldo");
            cardDto.getCardData().get(9).setValue(txt_seguradoraSaldo);

            cardDto.getCardData().add(10,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
            cardDto.getCardData().get(10).setField(
                "txt_seguradoraRentabilidade");

            cardDto.getCardData().get(10).setValue(txt_seguradoraRentabilidade);
```

```

        cardDto.getCardData().add(11,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(11).setField("txt_investEsporadic");
        cardDto.getCardData().get(11).setValue(txt_investEsporadic);

        cardDto.getCardData().add(12,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(12).setField("txt_investMensal");
        cardDto.getCardData().get(12).setValue(txt_investMensal);
    }

    } else if (rdTipoInclusao == "resgate") {
        cardDto.getCardData().add(9,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(9).setField("txt_seguradoraResgate");
        cardDto.getCardData().get(9).setValue(txt_seguradoraResgate);
    }

    cardDtos.getItem().add(0, cardDto);

    var result = service.create("1", "" +
config.usuarioFluig, ""
        + config.senhaFluig, cardDtos)

    log.info("#result =====> "
        + result.getItem().get(0).documentId);
    var docId = result.getItem().get(0).documentId;
    if (docId > 0) {
        return docId;
    } else {
        return "nok"
    }
}

```

```

        log
                .error("Erro para salvar registro de
saldo x seguradora por cliente..." +
result.getItem().get(0).webServiceMessage);
        }

    } catch (e) {
        return "nok"
        log
                .error("Erro para salvar registro de
saldo x seguradora por cliente..." +
e.toString());
    }
}

function
atualizaRegistroPropostaFluig(retornoRegistroExistente,
cpfCnpj,
        zoomNomeCompleto, txt_cdSeguradora, txt_seguradora,
txt_NewProposta,
        mensalidade, aporte, dataCadastro, txt_proReg,
txt_VgblPgblNew, fundo,
        percentual) {
log.info("## atualizaRegistroPropostaFluig ###")
try {

    var config = recuperaDadosFixos();

    var serviceProvider = ServiceManager
            .getServiceInstance("ECMCardservice"); //  

codigo do serviço
    var wkServiceEngine = serviceProvider

    .instantiate("com.totvs.ECMCardServiceService");
    var service = wkServiceEngine.getCardServicePort();

    var cardFieldDtoArray = serviceProvider

    .instantiate("com.totvs.CardFieldDtoArray");

    cardFieldDto1 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
    cardFieldDto1.setField("cpfCnpj");
    cardFieldDto1.setValue(cpfCnpj);
}

```

```

        cardFieldDto2 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto2.setField("nomeCompleto");
        cardFieldDto2.setValue(zoomNomeCompleto);

        cardFieldDto3 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto3.setField("txt_cdSeguradora");
        cardFieldDto3.setValue(txt_cdSeguradora);

        cardFieldDto4 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto4.setField("txt_seguradora");
        cardFieldDto4.setValue(txt_seguradora);

        cardFieldDto5 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto5.setField("txt_propostaSeguradora");
        cardFieldDto5.setValue(txt_NewProposta);

        cardFieldDto6 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto6.setField("txt_mensalidade");
        cardFieldDto6.setValue(mensalidade);

        cardFieldDto7 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto7.setField("txt_aporte");
        cardFieldDto7.setValue(aporte);

        cardFieldDto8 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto8.setField("campoDescriptor");
        cardFieldDto8.setValue(zoomNomeCompleto + " - " +
txt_seguradora
                + " - " + txt_NewProposta);

        cardFieldDto9 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto9.setField("dataCadastro");

        cardFieldDto9.setValue(formataDataParaYYYYMMDD(dataCadastro));

        cardFieldDto10 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto10.setField("txt_proReg");

```

```

        cardFieldDto10.setValue(txt_proReg);

        cardFieldDto11 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto11.setField("txt_VgblPgblNew");
        cardFieldDto11.setValue(txt_VgblPgblNew);

        cardFieldDto12 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto12.setField("nomeFundo");
        cardFieldDto12.setValue(fundo);

        cardFieldDto13 =
serviceProvider.instantiate("com.totvs.CardFieldDto");
        cardFieldDto13.setField("percentualFundo");
        cardFieldDto13.setValue(percentual);

        cardFieldDtoArray.getItem().add(0, cardFieldDto1);
        cardFieldDtoArray.getItem().add(1, cardFieldDto2);
        cardFieldDtoArray.getItem().add(2, cardFieldDto3);
        cardFieldDtoArray.getItem().add(3, cardFieldDto4);
        cardFieldDtoArray.getItem().add(4, cardFieldDto5);
        cardFieldDtoArray.getItem().add(5, cardFieldDto6);
        cardFieldDtoArray.getItem().add(6, cardFieldDto7);
        cardFieldDtoArray.getItem().add(7, cardFieldDto8);
        cardFieldDtoArray.getItem().add(8, cardFieldDto9);
        cardFieldDtoArray.getItem().add(9, cardFieldDto10);
        cardFieldDtoArray.getItem().add(10,
cardFieldDto11);
        cardFieldDtoArray.getItem().add(11,
cardFieldDto12);
        cardFieldDtoArray.getItem().add(12,
cardFieldDto13);

        var result = service.updateCardData(1, "" +
config.usuarioFluig, ""
                                + config.senhaFluig,
parseInt(retornoRegistroExistente),
                                cardFieldDtoArray)

        log.info("#result =====> "
                + result.getItem().get(0).documentId);
        var docId = result.getItem().get(0).documentId;
        if (docId > 0) {
            return docId;
        } else {
            return "nok"
    }
}

```

```

        log
                .error("Erro para salvar registro de
Proposta x Seguradora por cliente..." +
result.getItem().get(0).webServiceMessage);
        }

    } catch (e) {
        return "nok"
        log
                .error("Erro para salvar registro de
saldo x seguradora por cliente..." +
e.toString());
    }
}

function incluiRegistroPropostaFluig(cpfCnpj,
zoomNomeCompleto,
        txt_cdSeguradora, txt_seguradora, txt_NewProposta,
mensalidade, aporte,
        dataCadastro, txt_proReg, txt_Vgb1PgblNew, fundo,
percentual) {
    log.info("### incluiRegistroPropostaFluig ###");
    try {

        var config = recuperaDadosFixos();

        var serviceProvider = ServiceManager
                .getServiceInstance("ECMCardservice"); // codigo do serviço
        var wkServiceEngine = serviceProvider

                .instantiate("com.totvs.ECMCardServiceService");
        var service = wkServiceEngine.getCardServicePort();
        var cardDtos =
serviceProvider.instantiate("com.totvs.CardDtoArray");
        var cardDto =
serviceProvider.instantiate("com.totvs.CardDto");

        cardDto.setParentDocumentId(parseInt("" +
recuperaIdFormPaiPropostaSeguradora()));
        cardDto.setColleagueId("'" + config.usuarioFluig);
        cardDto.setInheritSecurity(true);
        var cardData = serviceProvider

```

```
.instantiate("com.totvs.CardFieldDtoArray");

        cardDto.getCardData().add(0,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
            cardDto.getCardData().get(0).setField("cpfCnpj");
            cardDto.getCardData().get(0).setValue(cpfCnpj);

        cardDto.getCardData().add(1,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
            cardDto.getCardData().get(1).setField("nomeCompleto");
            cardDto.getCardData().get(1).setValue(zoomNomeCompleto);

        cardDto.getCardData().add(2,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
            cardDto.getCardData().get(2).setField("txt_cdSeguradora");
        );

        cardDto.getCardData().get(2).setValue(txt_cdSeguradora);

        cardDto.getCardData().add(3,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
            cardDto.getCardData().get(3).setField("txt_seguradora");
            cardDto.getCardData().get(3).setValue(txt_seguradora);

        cardDto.getCardData().add(4,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
            cardDto.getCardData().get(4).setField("txt_propostaSeguradora");
            cardDto.getCardData().get(4).setValue(txt_NewProposta);

        cardDto.getCardData().add(5,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
```

```
        cardDto.getCardData().get(5).setField("campoDescriptor");
        cardDto.getCardData().get(5).setValue(
            zoomNomeCompleto + " - " + txt_seguradora
+ " - "
            + txt_NewProposta);

        cardDto.getCardData().add(6,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(6).setField("dataCadastro");
        cardDto.getCardData().get(6).setValue(
            formataDataParaYYYYMMDD(dataCadastro));

        cardDto.getCardData().add(7,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(7).setField("txt_aporte");
        cardDto.getCardData().get(7).setValue(aporte);

        cardDto.getCardData().add(8,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(8).setField("txt_mensalidade")
;
        cardDto.getCardData().get(8).setValue(mensalidade);

        cardDto.getCardData().add(9,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
        cardDto.getCardData().get(9).setField("nomeFundo");
        cardDto.getCardData().get(9).setValue(fundo);

        cardDto.getCardData().add(10,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(10).setField("percentualFundo"
);
        cardDto.getCardData().get(10).setValue(percentual);

        cardDto.getCardData().add(11,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));
```

```

        cardDto.getCardData().get(11).setField("txt_proReg");
        cardDto.getCardData().get(11).setValue(txt_proReg);

        cardDto.getCardData().add(12,
            serviceProvider.instantiate("com.totvs.CardFieldDto"));

        cardDto.getCardData().get(12).setField("txt_VgblPgblNew"
    );

    cardDto.getCardData().get(12).setValue(txt_VgblPgblNew);

    cardDtos.getItem().add(0, cardDto);

    var result = service.create("1", "" +
config.usuarioFluig, ""
                                + config.senhaFluig, cardDtos)
    log.info("## result: " + result);
    log.info("#result =====> "
                                + result.getItem().get(0).documentId);
    var docId = result.getItem().get(0).documentId;
    if (docId > 0) {
        return docId;
    } else {
        return "nok";
    }
    log
        .error("Erro para salvar registro de
saldo x seguradora por cliente..." +
result.getItem().get(0).webServiceMessage);
}

} catch (e) {
    return "nok";
    log
        .error("Erro para salvar registro de
proposta x seguradora por cliente..." +
e.toString());
}
}

function consultaRegistroSaldoSeguradoraPorCliente(cpfCnpj,
txt_cdSeguradora,
txt_seguradoraMesComp, txt_propostaSeguradora) {

```

```

    var vCpf = String(cpfcnpj || "");
    var vSeg = String(txt_cdSeguradora || "");
    var vMes = String(txt_seguradoraMesComp || "");
    var vProp = String(txt_propostaSeguradora || "");

    var c1 = DatasetFactory.createConstraint('cpfcnpj',
vCpf, vCpf,
                    ConstraintType.MUST);
    var c2 =
DatasetFactory.createConstraint('txt_cdSeguradora', vSeg,
vSeg,
                    ConstraintType.MUST);
    var c3 =
DatasetFactory.createConstraint('txt_seguradoraMesComp',
vMes,
                    vMes, ConstraintType.MUST);
    var c4 =
DatasetFactory.createConstraint('metadata#active', "true",
"true",
                    ConstraintType.MUST);
    var c5 =
DatasetFactory.createConstraint('rdTipoInclusao', "saldo",
"saldo", ConstraintType.MUST);
    var c6 =
DatasetFactory.createConstraint('txt_propostaSeguradora',
vProp,
                    vProp, ConstraintType.MUST);

    // 2. Passar arrays vazios [] em vez de null para evitar
erro de assinatura do método
    var constraints = new Array(c1, c2, c3, c4, c5, c6);

    var dataset = DatasetFactory.getDataset(
        'ds_registrosSaldoSeguradoraCliente', [],
constraints, []);

    if (dataset != null && dataset.rowsCount > 0) {
        return dataset.getValue(0, "metadata#id");
    }
    return "";
}

function
consultaRegistroPropostaSeguradoraPorCliente(cpfcnpj,
txt_cdSeguradora, txt_NewProposta) {

```

```

    // 1. Sanitização das variáveis para solicitações
antigas que podem conter nulos
    var vCpf = String(cpfCnpj || "");
    var vSeg = String(txt_cdSeguradora || "");
    var vProp = String(txt_NewProposta || "");

    var c1 = DatasetFactory.createConstraint('cpfCnpj',
vCpf, vCpf,
                           ConstraintType.MUST);
    var c2 =
DatasetFactory.createConstraint('txt_cdSeguradora', vSeg,
vSeg,
                           ConstraintType.MUST);
    var c3 =
DatasetFactory.createConstraint('txt_propostaSeguradora',
vProp,
                           vProp, ConstraintType.MUST);
    var c4 =
DatasetFactory.createConstraint('metadata#active', "true",
"true",
                           ConstraintType.MUST);

    // 2. Uso de arrays vazios para compatibilidade de
assinatura no getDataset
    var constraints = new Array(c1, c2, c3, c4);
    var dataset = DatasetFactory.getDataset(
        'ds_registrosPropostaSeguradoraCliente', [],
constraints, []);

    if (dataset != null && dataset.rowsCount > 0) {
        return dataset.getValue(0, "metadata#id");
    }
    return "";
}

function recuperaDadosFixos() {

    var constraintDs_config1 =
DatasetFactory.createConstraint(
    'metadata#active', 'true', 'true',
ConstraintType.MUST);
    var colunasDs_config = new Array('usuarioFluig',
'senhaFluig');
    var datasetDs_config =
DatasetFactory.getDataset('ds_configIntegracao',
    colunasDs_config, new
Array(constraintDs_config1), null);
}

```

```

try {

    if (datasetDs_config.rowsCount > 0) {
        return [
            "usuarioFluig" :
datasetDs_config.getValue(0, "usuarioFluig"),
            "senhaFluig" :
datasetDs_config.getValue(0, "senhaFluig"),
        ]
    } else {
        log.error("Não há usuário e senha cadastrados para integração...")
        throw "Não há usuário e senha cadastrados para integração"
    }
}

) catch (e) {
    log.error("Não há usuário e senha cadastrados para integração..." + e)
    throw "Não há usuário e senha cadastrados para integração" + e
}
}

function recuperaIdFormPaiSaldoSeguradora() {

    var constraintDocument1 =
DatasetFactory.createConstraint('datasetName',
                                'ds_registrosSaldoSeguradoraCliente',
                                'ds_registrosSaldoSeguradoraCliente',
ConstraintType.MUST);
    var constraintDocument2 =
DatasetFactory.createConstraint('activeVersion',
                                'true', 'true', ConstraintType.MUST);
    var colunasDocument = new
Array('documentPK.documentId');
    var datasetDocument =
DatasetFactory.getDataset('document',
                            colunasDocument,
                            new Array(constraintDocument1,
constraintDocument2), null);

    return datasetDocument.getValue(0,
"documentPK.documentId")
}

```

```

function recuperaIdFormPaiPropostaSeguradora() {
    var constraintDocument1 =
        DatasetFactory.createConstraint('datasetName',
            'ds_registrosPropostaSeguradoraCliente',
            'ds_registrosPropostaSeguradoraCliente',
            ConstraintType.MUST);
    var constraintDocument2 =
        DatasetFactory.createConstraint('activeVersion',
            'true', 'true', ConstraintType.MUST);
    var colunasDocument = new
        Array('documentPK.documentId');
    var datasetDocument =
        DatasetFactory.getDataset('document',
            colunasDocument,
            new Array(constraintDocument1,
            constraintDocument2), null);
    return datasetDocument.getValue(0,
        "documentPK.documentId");
}

function retornaIndices(campoBase) {
    var indices = [];
    var campos = hAPI.getCardData(getValue("WKNumProces"));
    var chaves = campos.keySet().toArray();

    for (var i in chaves) {
        if (chaves[i].indexOf(campoBase + "____") > -1) {
            indices.push(chaves[i].split("____")[1]);
        }
    }
    return indices;
}

function preencherZerosEsquerda(numero, tamanhoTotal) {
    var numeroString = String(numero);
    while (numeroString.length < tamanhoTotal) {
        numeroString = '0' + numeroString;
    }
    return numeroString;
}

function retornaData() {
    // Buscar a data e hora atual
    var data = new Date();
}

```

```

var dia = data.getDate();
var mes = data.getMonth() + 1;
var ano = data.getFullYear();
var hora = data.getHours();
var min = data.getMinutes();
var seg = data.getSeconds();

if (dia < 10) {
    dia = "0" + dia;
}
if (mes < 10) {
    mes = "0" + mes;
}
if (hora < 10) {
    hora = "0" + hora;
}
if (min < 10) {
    min = "0" + min;
}
if (seg < 10) {
    seg = "0" + seg;
}

return dataInclusaoCompleta = dia + '/' + mes + '/' +
ano;
}

function dataFormatada(data) {
    // Expressão regular para verificar o formato dd/mm/yyyy
    var regexDDMMYYYY = /^(\d{2})/(\d{2})/(\d{4})$/;

    // Expressão regular para verificar o formato yyyy-mm-dd
    var regexYYYYMMDD = /^(\d{4})-(\d{2})-(\d{2})$/;

    if (regexYYYYMMDD.test(data)) {
        // Se a data estiver no formato yyyy-mm-dd, fazemos
        a conversão para
        // yyyy-mm-dd
        var partes = data.split('-');
        var dia = partes[2];
        var mes = partes[1];
        var ano = partes[0];
        return dia + '/' + mes + '/' + ano;
    } else if (regexDDMMYYYY.test(data)) {
        // Se a data estiver no formato dd/mm/yyyy, não é
        necessário converter
        return data;
}

```

```
    } else {
        // Se a data não estiver em nenhum dos formatos
        // suportados, retorna null
        // ou a própria data
        // Isso pode ser ajustado de acordo com a
        // necessidade do seu caso.
        return null;
    }
}

function formataDataParaYYYYMMDD(dataString) {
    // Separa os componentes da data em um array
    var partesData = dataString.split("/");
    // Retorna a data no formato aaaamdd
    return partesData[2] + " " + partesData[1] + " " +
partesData[0];
}
```