

# Integração do PJE 2G com o SEEU

<b>Integração do PJE 2G com o SEEU</b>	<b>1</b>
Objetivo Geral	1
Objetivos Específicos	1
Integração	2
Recepção do Processo no PJE 2G	2
FLX_INICIAL	2
Envio do Processo ao SEEU	5
(SG) Fluxo básico geral - TJBA	6
(SG) Remessa a outra instância - SEEU	6
Fluxo de Integração	10
Classes	10
IntercomunicacaoRetornoSEEUHome	10
retornarSEEU	11
converterManifestacaoSEEU	11
posEnviarProcessoSEEU	11
IntercomunicacaoSEEUClient	11
ProcessoDocumentoParaDocumentoConverter	11
converterSEEU	11
ProcessoJudicialService	12
observaRetornoProcessoSEEU	12
FluxoDAO	12
recuperarUltimaTaskInstancePorVariavel	12
Parâmetros	12
Siglas	12

## Objetivo Geral

Integrar as aplicações PJE 2G e SEEU, para possibilitar o envio e o recebimento de processos entre as aplicações.

## Objetivos Específicos

- Permitir que o PJE 2G receba manifestações do sistema SEEU, protocolando os novos processos, quando estes não estiverem presentes na base de dados;
- Permitir que o PJE 2G receba manifestações em diligência do sistema SEEU, promovendo a continuidade dos procedimentos, que já existe na base de dados;
- Etiquetar todos os processos novos de origem do SEEU;
- Permitir que o PJE 2G envie ao SEEU processo em diligência, sinalizando ao usuário que o processo foi remetido e aguarda retorno;
- Permitir que o PJE 2G envie ao SEEU o processo por baixa definitiva, sinalizando ao usuário que o processo não será mais retornado.

## Integração

A integração entre as aplicações foi realizada em duas etapas: Recepção do Processo no PJE 2G e Envio do Processo ao SEEU. Em ambos os casos foi adotado o MNI como protocolo de intercomunicação.

### Recepção do Processo no PJE 2G

Na recepção do processo, foram utilizados os mesmos recursos aplicados na recepção do processo provenientes do PJE 1G, realizando apenas algumas refatorações, sendo: no método atualizarProcesso, é verificado se na manifestação processual possui o atributo tipo de retorno e se esse tipo é igual ao valor "D". Também é verificado se o processo com o idManifestante (Identificador do usuário que enviou o processo) é o mesmo contido no parâmetro ID\_MANIFESTANTE\_SEEU\_PJE. Se essas verificações forem verdadeiras, conclui-se que o processo é de origem do SEEU e está sendo enviado em diligência. Dessa forma é disparado um evento, atualizando o fluxo que esse processo se encontra, finalizando a tarefa atual (SG) Remetidos ao primeiro grau - Aguardar e vinculando-o a tarefa (SG) Recebimento do primeiro grau SEEU. Nessa tarefa, o usuário terá apenas a transição "Prosseguir", que ao ser selecionado, encaminha o processo para o (SG) Fluxo básico geral - TJBA, vinculando a tarefa (SG) Secretaria - Analisar.

### FLX\_INICIAL

Código do Fluxo: FLX\_INICIAL

- O fluxo foi alterado, incluído um nó de decisão e um nó de sistema. No nó de decisão, verifica-se o usuário que está enviando o processo (idManifestante) e caso seja o usuário SEEU, o nó de sistema é chamado para incluir a etiqueta (Processo SEEU). Se não for o usuário SEEU, segue o fluxo normalmente como um processo de origem do PJE 1G.
- A etiqueta aplicada no nó de decisão tem a localização TJBA, ou seja, é visível a todos os usuários e não é possível removê-la via sistema

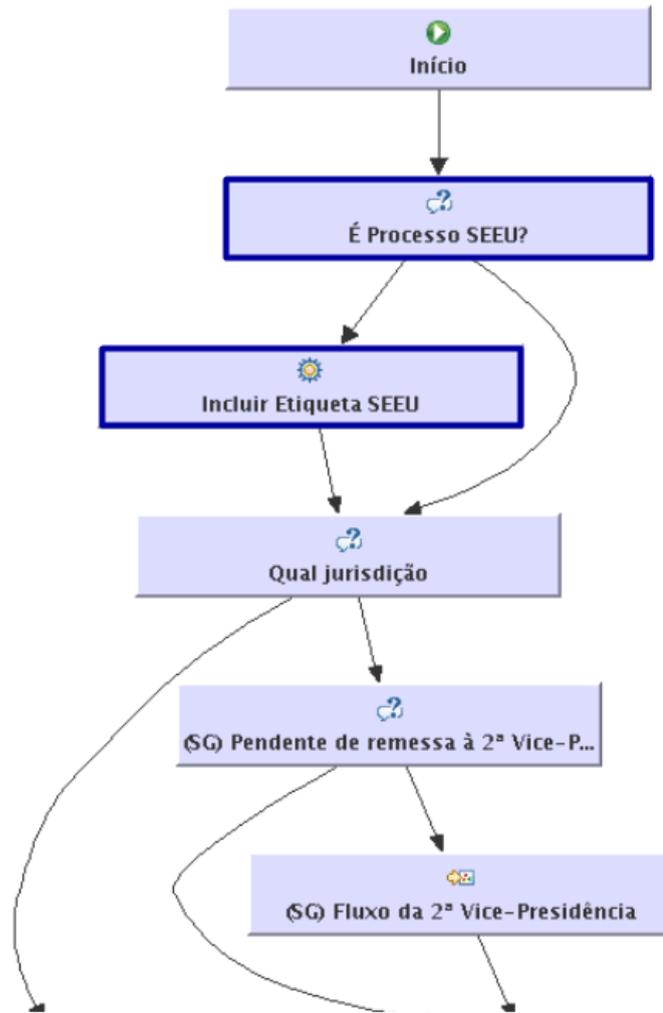


Figura XXX: Fluxo FLX\_INICIAL alterado, com a inclusão do nó de decisão É Processo SEEU e o nó de sistema Incluir Etiqueta SEEU.

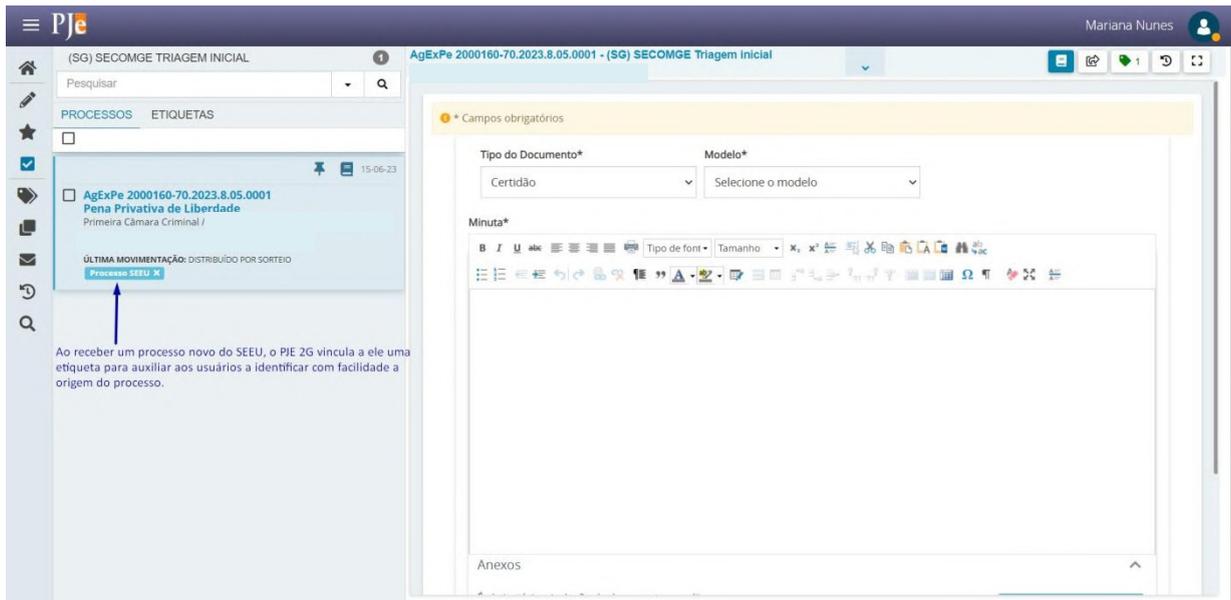


Figura XXX: Processo do SEEU (Etiqueta Processo SEEU) recebido no PJE.

```

private String atualizarProcesso(ManifestacaoProcessual tipoManifestacaoProcessual) throws Exception {
    // carregar o processo
    ProcessoTrf processoTrf = obterProcessoTrf(tipoManifestacaoProcessual.getNumeroProcesso(), false);
    ProcessoTrfHome instance().setInstance(processoTrf);
    validarTipoDocumento(tipoManifestacaoProcessual.getDocumento(), false, true, Integer.valueOf(processoTrf.getClasseJudicial().getCodClasseJudicial()));
    validarProcessoRemetido(processoTrf);
    validarProcessoPeticaoBloqueada(processoTrf);
    validarAtendimentoPlantao(tipoManifestacaoProcessual);

    //PJEEI-18990
    //Requisição feita por pessoa não associada ao processo.
    boolean isPessoaAssociada = isParteAssociada(Authenticator.getPessoaLogada(), processoTrf);
    String numeroProcessoIG = getNumeroProcessoIGraU(tipoManifestacaoProcessual);
    List<ProcessoDocumento> documentos = carregarColecaoDocumentoProcessual(
        tipoManifestacaoProcessual.getDocumento(),
        processoTrf.getProcesso(),
        null,
        numeroProcessoIG);

    if (isAtendimentoPlantao(tipoManifestacaoProcessual)) {
        PlantaoJudicialService.getInstance().registraSeDeveTrParaPlantao();
    }

    // Armazena os documentos da requisição para apresentá-los no recibo.
    setDocumentosRequisicao(documentos);

    //PJEEI-22205 Adição da condição isRemessa
    boolean isRemessa = MNIParametroUtil.obterValor(tipoManifestacaoProcessual, MNIParametro.PARAM_URL_ORIGEM_ENVIO) != null;
    if (isPessoaAssociada && isRemessa) {
        ProcessoDocumentoPeticaoNaOLidaManager processoDocumentoPeticaoNaOLidaManager = ComponentUtil.getComponent(ProcessoDocumentoPeticaoNaOLidaManager.class);
        for (ProcessoDocumento processoDocumento : documentos) {
            ProcessoDocumentoPeticaoNaOLida peticaoAvulsa = new ProcessoDocumentoPeticaoNaOLida();
            peticaoAvulsa.setProcessoDocumento(processoDocumento);
            peticaoAvulsa.setRetificado(false);
            peticaoAvulsa.setRetirado(false);
            processoDocumentoPeticaoNaOLidaManager.persist(peticaoAvulsa);
        }
    }

    ProcessoDocumento documentoPrincipal = iniciarFluxoDocumentoPrincipal(processoTrf, documentos);

    // TODO atualizar os movimentos
    salvarManifestacao(tipoManifestacaoProcessual, processoTrf);
    Events.getInstance().raiseEvent(AFTER_EVENT_NAME, tipoManifestacaoProcessual, processoTrf, documentoPrincipal);

    String tipoRetornoEmDiligenciaSEEU = MNIParametroUtil.obterValor(tipoManifestacaoProcessual.getDadosBasicos().getOutroParametro(), MNIParametro.PARAM_SEEU_TIPO_RETORNO_2_GRAU);

    if (tipoManifestacaoProcessual.getIdManifestante().equals(ParametroUtil.getParametro(Parametros.ID_MANIFESTANTE_SEEU_PJE))) {
        if (tipoRetornoEmDiligenciaSEEU != null && tipoRetornoEmDiligenciaSEEU.equals("D")) {
            Events.getInstance().raiseEvent(ProcessoJudicialService.EVENTO_SINALIZAR_FLUXO, processoTrf);
        }
    }
}

```

Figura XX: Verificação do manifestante e do tipo de retorno.

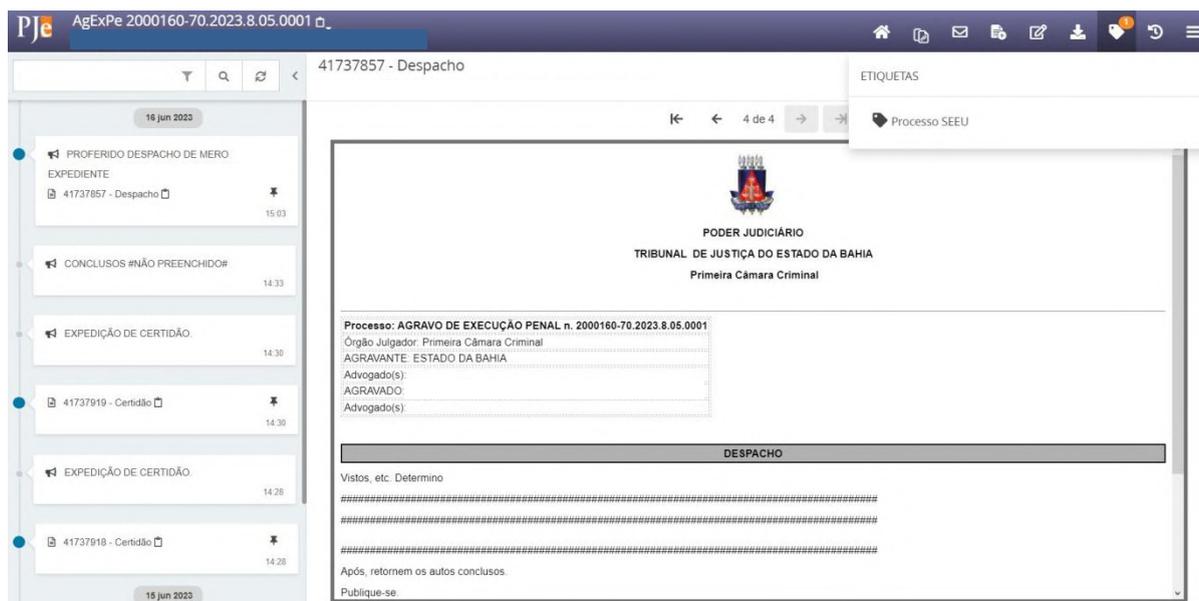


Figura XXX: Visualização da etiqueta nos autos digitais.

## Envio do Processo ao SEEU

Para o envio do processo ao SEEU, foi utilizado como modelo o padrão já aplicado no envio de processos ao PJE 1G. Dessa forma, foi realizada uma cópia da classe `IntercomunicacaoRetornoHome.java` e refatorada para atender as características que o SEEU exige. A classe foi nomeada de `IntercomunicacaoRetornoSEEUHome.java` e também implementa a classe abstrata `AbstractIntercomunicacaoHome`.

No método `retornarSEEU`, após o usuário selecionar o motivo da remessa e clicar no botão `Retornar` para a Instância de Origem - SEEU, é iniciado o processamento da solicitação.

Primeiro é obtido os objetos `processo` e `mp`, que são respectivamente o processo que será retornado e a manifestação processual vinculado a esse processo. Em seguida, instanciado o objeto `intercomunicacaoSEEUClient.java` do tipo `IntercomunicacaoSEEUClient.java` para trabalhar os dados de conexão do web método de retorno. Depois é instanciado o objeto `manifestacaoProcessualSEEU` do tipo `ManifestacaoProcessual.java` que recebe o objeto retornado do método `converterManifestacaoSEEU` que está presente na classe `IntercomunicacaoRetornoSEEUHome.java`. Esse método é o responsável por ajustar os dados para serem retornados no padrão que o SEEU existe. Vale destacar os atributos de autenticação da manifestação. É nesse método que são vinculados o usuário e a senha que serão utilizados para autenticação na entrega da manifestação no SEEU. O usuário foi fornecido pela equipe do SEEU e a senha, que é criptografada no padrão MD5, é composta por uma chave que também foi fornecida pela equipe do SEEU e incluída de forma estática no código, concatenado à data do momento do envio da manifestação ao SEEU(data atual). Além disso, no método `converterManifestacaoSEEU` são convertidos os documentos do processo e construído o cabeçalho da manifestação processual.

Após a conversão do processo, o mesmo é enviado por meio do método `web entregarManifestacaoProcessual` da classe `IntercomunicacaoSEEUClient.java`. Se o processo for entregue com sucesso ao SEEU, o mesmo será vinculado a tarefa (SG) `Remetidos ao primeiro grau - Aguardar se o processo foi enviado com o motivo da remessa "em diligência"` ou na tarefa (SG) `Remetidos ao primeiro grau SEEU se o motivo for "baixa`

definitiva”, e o usuário receberá a mensagem de retorno “Processo enviado ao Sistema SEEU.”. Caso contrário, o processo continuará na tarefa e será devolvido ao usuário a mensagem “ Erro ao entregar a manifestação: Motivo do erro.”.

(SG) Fluxo básico geral - TJBA

Código do Fluxo: SG\_FBG\_TJBA

- O fluxo foi alterado, incluindo um nó de decisão (SG) Verifica se é processo SEEU, após o nó de subprocesso (SG) Encerrar expediente para remessa, para verificar se o processo é de origem do sistema SEEU. Em caso positivo, o processo é redirecionado para o fluxo (SG) Remessa a outra instância - SEEU, caso contrário, para o fluxo (SG) Remessa a outra instância.

(SG) Remessa a outra instância - SEEU

Código do Fluxo: SG\_REM\_OUT\_INS\_SEEU

- O fluxo foi desenvolvido para controlar as ações de remessa para os processos de origem do sistema SEEU.

```
public void retornarSEEU() {
    try {
        ProcessoTrf processo = getInstance();
        br.jus.pje.nucleo.entidades.ManifestacaoProcessual mp = manifestacaoProcessualManager.findByProcessoTrfSEEU(getInstance());

        EnderecoWsdManager enderecoWsd = ComponentUtil.getComponent(EnderecoWsdManager.class);
        EnderecoWsd enderecoWsdSEEU = enderecoWsd.findById(5);
        setEnderecoWsd(enderecoWsdSEEU);

        IntercomunicacaoSEEUClient intercomunicacaoSEEUClient = new IntercomunicacaoSEEUClient(enderecoWsdSEEU);

        ManifestacaoProcessual manifestacaoProcessualSEEU = converterManifestacaoSEEU();

        RespostaManifestacaoProcessual respostaSEEU = intercomunicacaoSEEUClient.entregarManifestacaoProcessual(manifestacaoProcessualSEEU);

        if(respostaSEEU.getMensagem() != null && respostaSEEU.isSucesso() == Boolean.TRUE) {
            processoJudicialService.sinalizarFluxo(processo, "remessaSEEU", 0, true, true);
            posEnviarProcessoSEEU(processo, "remessaRetornoSEEU", processo.getNumeroProcesso(), "Processo devolvido ao sistema SEEU.", enderecoWsdSEEU);
            FacesMessages.instance().add("Processo enviado ao Sistema SEEU.");
        } else {
            throw new Exception(respostaSEEU.getMensagem());
        }
    } catch (Exception e) {
        FacesUtil.adicionarMensagemInfo(true, e);
    }
}
```

Figura XX: Método retornarSEEU.

```

public ManifestacaoProcessual converterManifestacaoSEEU() {
    ManifestacaoProcessual manifestacaoProcessualSEEU = new ManifestacaoProcessual();
    Date dataAtual = new Date();
    SimpleDateFormat dateFormatyyyyMdd = new SimpleDateFormat("yyyyMdd");
    String dataFormatada = dateFormatyyyyMdd.format(dataAtual);

    manifestacaoProcessualSEEU.setIdManifestante(ParametroUtil.getParametro(Parametros.ID_MANIFESTANTE_PJE_SEEU));
    final String md5Modelo = Crypto.encodeMD5(ParametroUtil.getParametro(Parametros.SENHA_MANIFESTANTE_PJE_SEEU)+dataFormatada);
    manifestacaoProcessualSEEU.setSenhaManifestante(md5Modelo);

    ProcessoDocumentoManager procManager = ComponentUtil.getComponent(ProcessoDocumentoManager.NAME);

    List<ProcessoDocumento> listaDocumentos = procManager.obterDocumentosPorProcesso(getInstance());

    List<String> listaIdCarregarBinario = new ArrayList<String>(0);
    listaIdCarregarBinario.add(TODOS);

    for (ProcessoDocumento processoDocumento : listaDocumentos) {
        manifestacaoProcessualSEEU.getDocumento().add(processoDocumentoParaDocumentoConverter.converterSEEU(processoDocumento, true, listaIdCarregarBinario, true));
    }

    NumeroUnico numeroUnico = new NumeroUnico();
    numeroUnico.setValue(getInstance().getProcesso().getNumeroProcesso());
    manifestacaoProcessualSEEU.setNumeroProcesso(numeroUnico);

    CabecalhoProcessual cabecalho = new CabecalhoProcessual();
    cabecalho.setClasseProcessual(Integer.valueOf(getInstance().getClasseJudicial().getCodClasseJudicial()));
    cabecalho.setNivelSigilo(0);
    cabecalho.setNumero(numeroUnico);

    manifestacaoProcessualSEEU.setDataEnvio(ConversorUtil.converterParaDataHora(new Date()));

    Parametro parametroSEEU = new Parametro();

    if(this.getMotivoRemessa().getValor().equals(EM_DILIGENCIA)) {
        parametroSEEU.setNome(TIPO_RETORNO_SEEU);
        parametroSEEU.setValor(TIPO_VALOR_EM_DILIGENCIA_SEEU);

        cabecalho.getOutroParametro().add(parametroSEEU);
    }

    if(this.getMotivoRemessa().getValor().equals(POR_BAIXA_DEFINITIVA)) {
        parametroSEEU.setNome(TIPO_RETORNO_SEEU);
        parametroSEEU.setValor(TIPO_VALOR_POR_BAIXA_DEFINITIVA_SEEU);

        cabecalho.getOutroParametro().add(parametroSEEU);
    }

    manifestacaoProcessualSEEU.setDadosBasicos(cabecalho);

    return manifestacaoProcessualSEEU;
}

```

Figura XX: converterManifestacaoSEEU

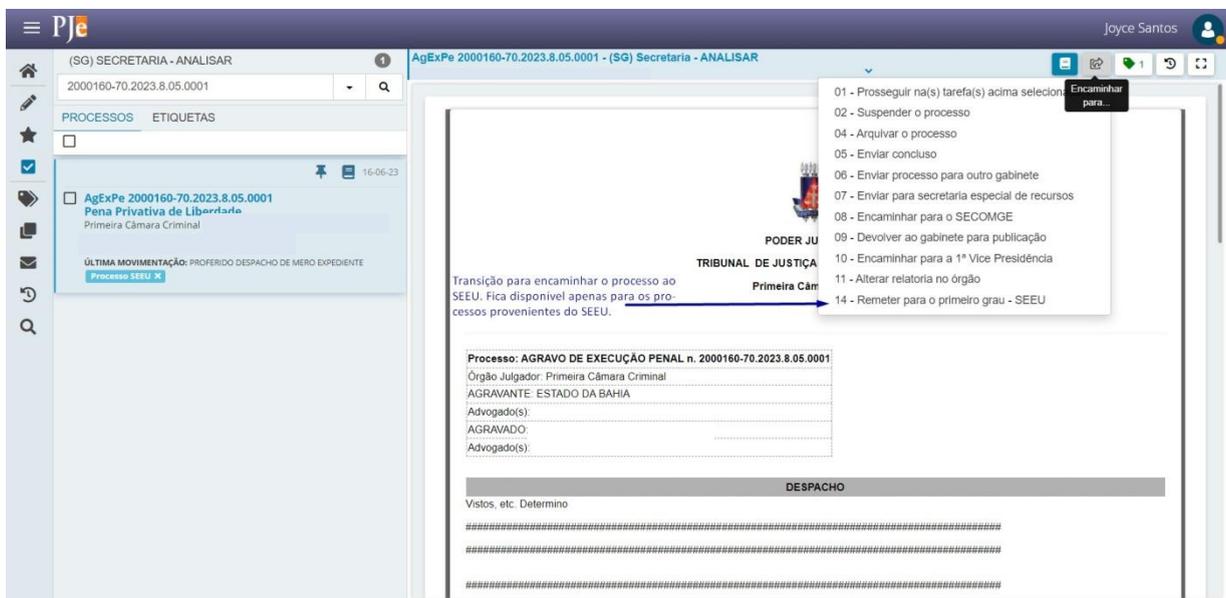


Figura XX: Transição 14 - Remeter para o primeiro grau - SEEU.

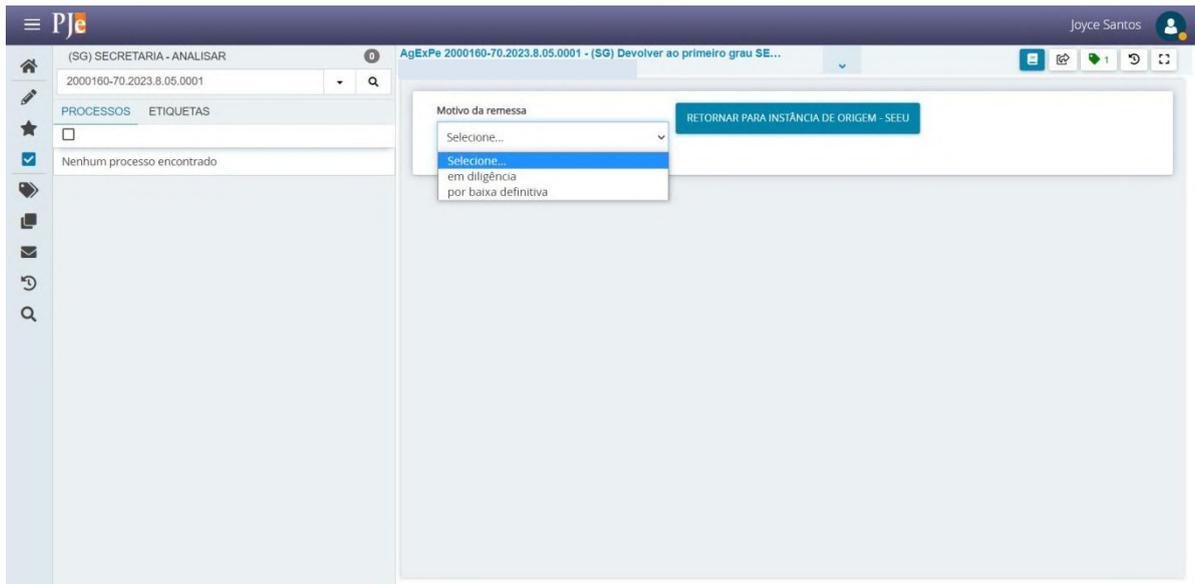


Figura XX: Processo na tarefa (SG) Devolver ao primeiro grau SEEU.

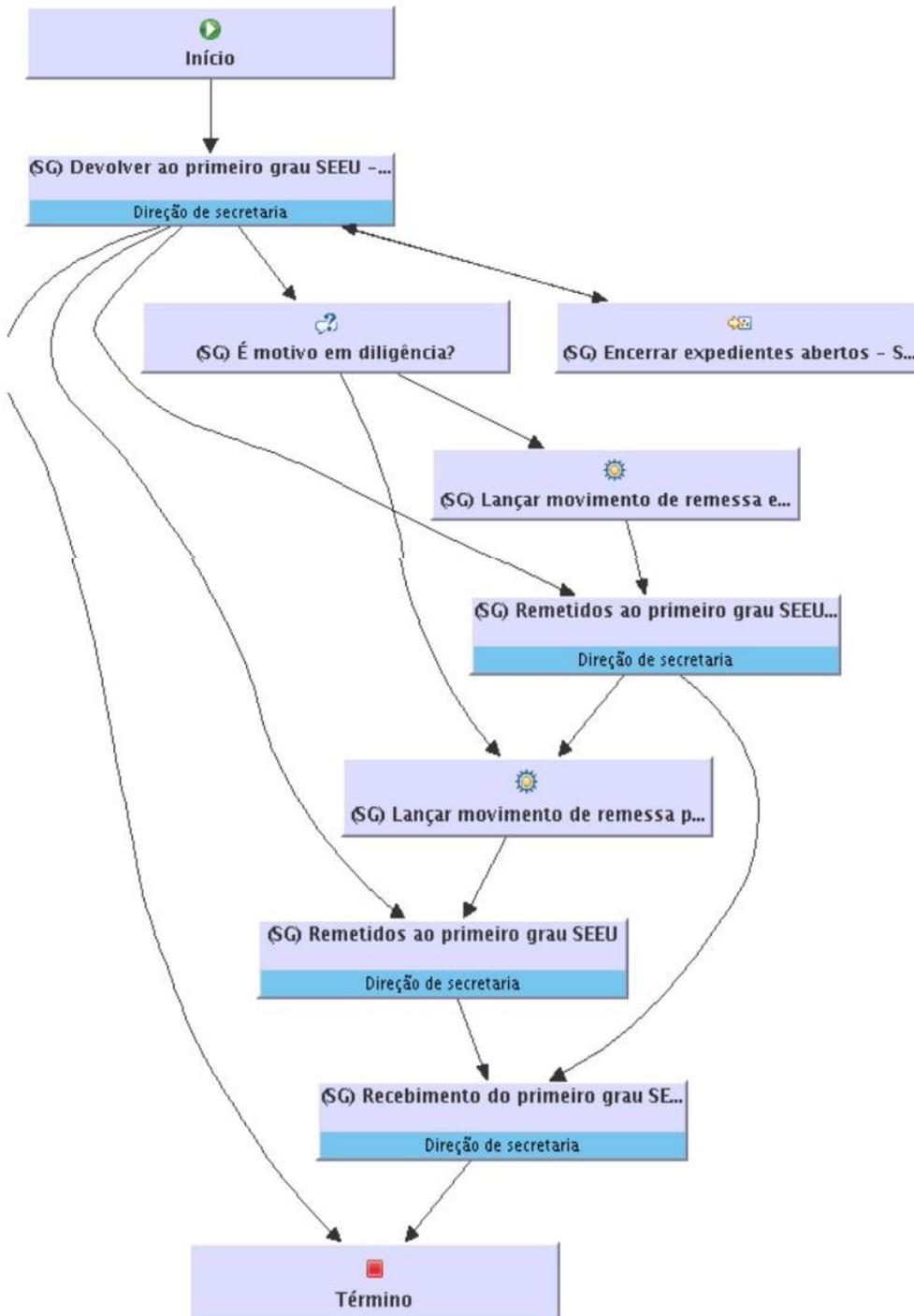
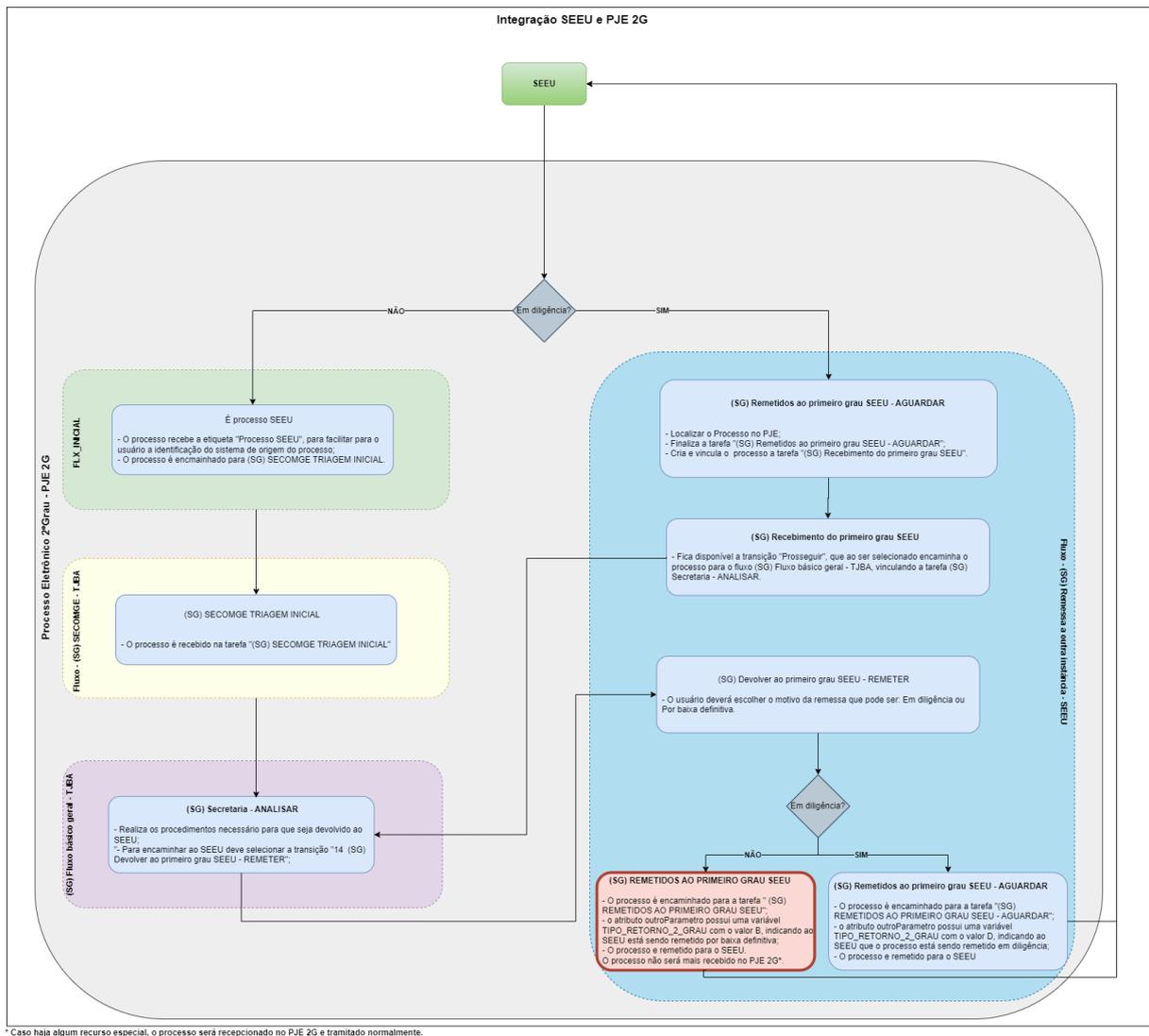


Figura XXX: Fluxo (SG) Remessa a outra instância - SEEU

# Fluxo de Integração



## Classes

IntercomunicacaoRetornoSEEUHome

Pacote: br.com.infox.cliente.home

- A classe estende da classe AbstractIntercomunicacaoHome;
- Além dos métodos, foram criados os seguintes atributos estáticos do tipo string, que são utilizados para configurar o tipo de retorno ao SEEU:

NOME	VALOR	DESCRIÇÃO
EM_DILIGENCIA	em diligência	Descrição que aparece como opção no front-end de motivos de remessa que o usuário pode selecionar.

POR_BAIXA_DEFINITIVA	por baixa definitiva	Descrição que aparece como opção no front-end de motivos de remessa que o usuário pode selecionar.
TIPO_RETORNO_SEEU	TIPO_RETORNO_2_GRAU	Nome do atributo que recebe o valor da opção que o usuário selecionou.
TIPO_VALOR_EM_DILIGENCIA_SEEU	D	Informação passado como valor quando o usuário seleciona a opção “em diligência” como motivo de remessa.
TIPO_VALOR_POR_BAIXA_DEFINITIVA_SEEU	B	Informação passado como valor quando o usuário seleciona a opção “por baixa definitiva” como motivo de remessa.

#### retornarSEEU

- Método chamado para iniciar a devolução do processo ao SEEU;

#### converterManifestacaoSEEU

- Trabalha os objetos que compõem a manifestação processual para ficar apto para ser enviado ao SEEU.

#### posEnviarProcessoSEEU

- Após o SEEU confirmar o recebimento do processo com sucesso, esse método é chamado para realizar a atualização do processo, variáveis do processo, tarefas e outros objetos que precisam ser atualizados após a confirmação de recebimento

#### IntercomunicacaoSEEUClient

Pacote: br.jus.cnj.pje.ws.client

- Classe com as configurações utilizadas para comunicar com o webservice de intercomunicação do SEEU.

#### ProcessoDocumentoParaDocumentoConverter

Pacote: br.jus.cnj.pje.intercomunicacao.v222.converter

- Classe de conversão de ProcessoDocumento para Documento da intercomunicação.

#### converterSEEU

- Converte um ProcessoDocumento para DocumentoProcessual com a opção de passar se os binários serão carregados e se o conteúdo será criptografado. Chamada os métodos carregarBinarioSEEU, carregarConteudoDocumentosSEEU, obterColecaoDocumentosVinculadosSEEU para auxiliar na conversão dos objetos.

ProcessoJudicialService

Pacote: br.jus.cnj.pje.nucleo.service

observaRetornoProcessoSEEU

- Método com a anotação @Observer, que fica escutando o retorno dos processos provenientes do SEEU para redirecionar para a tarefa (SG) Recebidos do primeiro grau SEEU

FluxoDAO

Pacote: br.jus.cnj.pje.business.dao

recuperarUltimaTaskInstancePorVariavel

- Esse método foi criado para obter as informações da última TaskInstance de acordo com os dados informados para que ele possa obter a transição de saída para a próxima tarefa, que no caso é a Recebimento do primeiro grau SEEU.

Parâmetros

**tjba:integracao:pje:seeu:idManifestante:pje:seeu**

- Identificador do sistema PJE no sistema SEEU.

**tjba:integracao:pje:seeu:senhaManifestante:pje:seeu**

- Senha do sistema PJE no sistema SEEU.

**tjba:integracao:pje:seeu:idManifestante:seeu:pje**

- Identificador do sistema SEEU no sistema PJE.

**tjba:integracao:pje:seeu:movimento**

- Código do movimento de remessa ao sistema SEEU.

**tjba:integracao:pje:seeu:tipo:documento**

- Código do tipo de documento de remessa ao sistema SEEU.

**tjba:integracao:pje:seeu:endereco:wSDL**

- Endereço wSDL de remessa ao sistema SEEU.

**pje:expressao:IdUsuarioCadastradorProcessoSEEU**

- Expressão utilizada no fluxo identificar que o processo é do SEEU.

Siglas

SEEU - Sistema Eletrônico de Execução Unificado.

PJE 2G - Processo Judicial Eletrônico do 2º Grau.

MNI - Modelo Nacional de Interoperabilidade