**PREPARAÇÃO**

**Atribuição do projeto**

O P.O. fornecerá o nome e a versão atual da automação a ser adequada.

**Baixar projeto do RTC**

Com o nome do projeto, verificar com Cleber ou na lista das automações no RTC (em construção) o caminho da última versão da automação no RTC.

**Verificar versão**

Com o projeto baixado, verificar versão informada e validar se é a mesma. Se for, prosseguir, se não, informar o P.O. para definição.

**Localizar .cs importantes**

Localizar os arquivos de código-fonte nos quais se encontram as informações de infraestrutura do projeto, sistemas acessados e funcionalidades implementadas na automação.

**ANÁLISE**

**Levantar infraestrutura**

Levantar as seguintes informações:

* Banco de dados utilizado (Servidores PB4, TB5 e Bancos 40X, 50X)
* Tabelas utilizadas
* Arquitetura (web ou desktop)
* Se é web, onde está publicado? (IIS PB4, condomínio TB5, portal de automações)

**Levantar sistemas acessados**

Levantar quais são os sistemas acessados/consultados pela automação, sejam:

* Mainframe/Terminal (tela-preta). Exemplo: F5, L7, F7, CA, SNG
* Intranet Web. Exemplo: VP, CCI, OD, Painel de Serviços
* API/WebService interno ou externo. Exemplo: API interna do CCI, WS externo DETRAN, Zflow

**Levantar funcionalidades**

Levantar quais funcionalidades performadas pela automação nos SISTEMAS ACESSADOS, exemplos:

* Raspagem no VP
* Requisição à API Interna do CCI
* Transação no F5
* Upload no Painel de Serviços
* Download da Zflow
* Login no SNG

**DOCUMENTAÇÃO**

**PRML**

Sigla para Planilha de Recursos de Monitoração para o Legado. Consiste em documentar os sistemas e funcionalidades levantadas utilizando este template. Após preenchimento, enviar para o P.O.

**Diagrama de Arquitetura**

Construção do diagrama de arquitetura no padrão da TI, com foco no ambiente da automação, baseado na infraestrutura e nos sistemas levantados. Após preenchimento, enviar para o P.O.

**Especificação Técnica**

Construção da especificação técnica no modelo do MEO, preenchendo as seguintes informações:

* Tecnologias/Linguagens de Programação
* Segurança
* DER
* Diagrama de Arquitetura

Após preenchimento, enviar para o P.O.

**Cadastro RaaS/Calculadora**

Cadastros obrigatórios por política, realizados pelo P.O. É nessa etapa que será gerado o ID do robô para serem feitos os logs no Splunk.

**Cadastro GMS e devolução PRML**

Cadastro das informações feito pelo P.O. nos bancos do GMS para obter os ids necessários para os logs no Splunk. A PRML é devolvida para o desenvolvedor prosseguir.

**MAPEAMENTO**

**Encontrar e marcar ponto global**

Procurar o ponto adequado para instanciar o objeto do GMS da DLL. Deve ser um ponto global para a automação, para que seja utilizado por qualquer ponto do código. Marcar esse ponto utilizando comentários.

**Encontrar e marcar loop**

Procurar o ponto onde começa o loop principal do programa, onde será tratado item a item da massa a processar e idealmente contém a parte mais importante do processamento da automação. Marcar esse ponto utilizando comentários.

**Encontrar e marcar login S.P.**

Procurar o ponto onde ocorre o login em cada sistema acessado (isso pode ocorrer tanto dentro do loop principal quando fora dele). Marcar esse ponto utilizando comentários.

**Encontrar e marcar pontos**

Procurar os pontos onde cada funcionalidade levantada para cada sistema é efetivamente executada, marcando esses pontos utilizando comentários. Marcar também onde o processamento de um item termina (log final para o item). Se possível, já definir as informações a serem enviadas. IMPORTANTE: Marcar APENAS um log final para cada item tratado.

**Encontrar e marcar final**

Encontrar o final geral da automação, onde todas as funcionalidades e processamentos já foram executadas. Marcar esse ponto utilizando comentários.

**IMPLEMENTAÇÃO**

**Importação DLL**

Incluir no projeto a DLL GMS2Dll como referência.

**Instância do objeto GMS2**

Instanciar um novo objeto GMS2 no ponto marcado como global na etapa de mapeamento. Passar o IdRobo que constará na PRML devolvida e a versão da automação, preferencialmente puxando essa informação do config ou de alguma outra forma automática, para que não seja uma string chumbada. Reforçando que este deve ser um objeto acessível a qualquer ponto do código e preferencialmente no início da aplicação, pois o construtor enviará o log START GERAL.

**Chamadas ao método de disparo**

Para cada ponto marcado na etapa de mapeamento, realizar a chamada ao método de disparo do objeto global definido, utilizando comentários para descrever o ponto de disparo e preenchendo os campos da seguinte forma, com as seguintes orientações:

Para login no S.P.:

|  |  |  |
| --- | --- | --- |
| Campo | Orientação | Obrigatório |
| logType | EnumLogType.NEGOCIO | SIM |
| logLevel | EnumLogLevel.LOGON | SIM |
| idFuncionalidade | 10 | SIM |
| IdApp | Id do sistema informado na PRML (consistente com o sistema que está sendo logado) | SIM |
| user | RACF do usuário que está logando no sistema | SIM |
| acao | “Realizando login no <<nome do sistema>>” | SIM |

Para cada funcionalidade executada:

|  |  |  |
| --- | --- | --- |
| Campo | Orientação | Obrigatório |
| logType | EnumLogType.NEGOCIO | SIM |
| logLevel | EnumLogLevel.INFO | SIM |
| idFuncionalidade | Verificar na seção 4.2.1 do Manual no Confluence | SIM |
| IdApp | Id do sistema informado na PRML (consistente com o sistema que está sendo logado) | SIM |
| user | RACF do usuário que está logado no sistema | SIM |
| acao | Descrever sucintamente o que a automação está fazendo | SIM |
| detalheObjetoNegocio | Informações-chave que identificam unicamente o item tratado, ex: new {‘Contrato’: ‘112981’} | NÃO |

Para o log final do item tratado com sucesso:

|  |  |  |
| --- | --- | --- |
| Campo | Orientação | Obrigatório |
| logType | EnumLogType.NEGOCIO | SIM |
| logLevel | EnumLogLevel.SUCCESS | SIM |
| IdApp | Id do sistema informado na PRML (consistente com o sistema que está sendo logado) | SIM |
| user | RACF do usuário que está logado no sistema | SIM |
| acao | “Processamento para o item finalizado com sucesso.” | SIM |
| detalheObjetoNegocio | Informações-chave que identificam unicamente o item tratado, ex: new {‘Contrato’: ‘112981’} | NÃO |

Para logs de erro:

|  |  |  |
| --- | --- | --- |
| Campo | Orientação | Obrigatório |
| logType | EnumLogType.NEGOCIO | SIM |
| logLevel | EnumLogLevel.ERROR | SIM |
| idFuncionalidade | Verificar na seção 4.2.1 do Manual no Confluence | SIM |
| IdApp | Id do sistema informado na PRML (consistente com o sistema que está sendo logado) | SIM |
| user | RACF do usuário que está logado no sistema | SIM |
| acao | Descrever sucintamente o que a automação está fazendo | SIM |
| detalheObjetoNegocio | Informações-chave que identificam unicamente o item tratado, ex: new {‘Contrato’: ‘112981’} | NÃO |
| descricaoErro | Preferencialmente, descrever o erro mapeado, caso contrário, informar a mensagem da exceção subida | NÃO |

ATENÇÃO: Logs de erro devem ser enviados apenas UMA vez por item tratado.

Para finalizar, chamar o método EnviarFimGeral no ponto de fim geral marcado no mapeamento.

IMPORTANTE: Implementar os logs de forma assíncrona (não esperar response) e em caso de erros, trata-los sem ação para NÃO impedir o processamento da automação.

**Publicação**

Alterar a versão da automação, verificar se a automação aponta para o servidor de produção e efetuar a publicação da nova versão. Por fim, atualizar o código no RTC.

**Monitoramento e sustentação**

A nova versão será monitorada pelo time de POs e pela operação durante um tempo para garantir que está funcionando corretamente. Caso ocorram problemas, será necessária a atuação em Sustentação para resolvê-lo.