

Technical Design Document

Bot Reporte TT8

Tabla de Contenido

1	Propósito del Documento	3
2	Alcance	3
3	Prerequisitos	3
4	Estructura de los archivos/carpetas.....	4
5	Diseño del Robot	4
	5.1 Main.xaml	4
	5.2 Inicialización.xaml.....	4
	5.3 Filtrar.xaml.....	5
	5.4 Seccionamiento_de_Columnas.xaml	5
	5.5 Procesar Datos Pre TT2.xaml.....	6
	5.6 Procesar Nivel de Tensión.xaml	6
	5.7 Codigo Dane.xaml.....	6
	5.8 Procesar Datos Post TT2.xaml	7
	5.9 Generar Reporte TT8.xaml	7
	5.10 Escribir Información TT2.xaml.....	8
6	Árbol de Invocación de Workflows	9
7	Dependencias del Proyecto.....	10
8	Riesgos y Puntos de Atención.	10

1 Propósito del Documento

Este documento explica los aspectos técnicos en detalle del robot diseñado y desarrollado utilizando UiPath. Esto permitirá una vista más amplia al proceso de diseño y permitirá a otros desarrolladores o terceros entender los prerequisites y requerimientos necesarios para la correcta ejecución del robot.

2 Alcance

El alcance de este documento incluye:

- Especificaciones del entorno
- Requerimientos del Sistema
- Prerequisites
- Versionamiento de UiPath
- Estructura de las carpetas y archivos
- Diseño del Robot
- Riesgos y puntos de atención.

El alcance de este documento no incluye:

- Disponibilidad de los sistemas.
- Cambios en los sistemas.
- Cambios a los insumos de entrada o la estructura de los datos.
- Cambios al proceso.

Nota – Este es un Documento de Diseño Técnico, por lo cual, en caso de necesitar información acerca del proceso de negocio, por favor remitarse al documento PDD o Documento de Diseño del Proceso.

3 Prerequisites

Los prerequisites para que la ejecución del robot sea exitosa son los siguientes:

- 4.1 Las siguientes aplicaciones se encuentran instaladas en el Sistema:
 - UiPath 21.4.6.0 o superior
 - Excel
 - Microsoft Onedrive
- 4.2 El sistema cuenta con una licencia válida de robot desatendido de UiPath.
- 4.3 El robot cuenta con acceso autorizado a todas las aplicaciones y rutas requeridas.
- 4.4 Los archivos utilizados se encuentran en el sistema siguiendo la estructura indicada en el archivo de configuración cuya ubicación se encuentra en la sección 5.

4.5 El archivo de Excel de Seguimiento AGPE no se encuentra siendo utilizado por miembros de la organización durante la ejecución.

4 Estructura de los archivos/carpetas

- **Carpeta Temporal:**
Ruta en la que los archivos temporales son almacenados.
- **Carpeta de Resultados:**
Ruta en la que los archivos generados por el robot son almacenados.
- **Repositorio del Código**
Ruta en la que el proyecto de UiPath es almacenado.
- **Carpeta de Configuración**
Ruta en la que el archivo de configuración para el proyecto es almacenado.

Archivo/Carpeta	Ubicación	Obligatorio
Temporal	N/A	N/A
Resultado(s)	D:\OneDrive - Grupo EPM\Escritorio\TT8Plantilla.xlsx	SI
Código/Proyecto	C:\Users\Gabriel Andre\Documents\UiPath\CreacionReporteAuto_Windows	SI
Configuración	C:\Users\Gabriel Andre\Documents\UiPath\CreacionReporteAuto_Windows\Resources	SI

5 Diseño del Robot

5.1 Main.xaml

En este workflow se invocan todos los demás workflows creados de manera secuencial para la correcta ejecución del proyecto.

Ubicación: \Main.xaml

5.2 Inicialización.xaml

En este workflow se asentan las bases para la correcta ejecución del proyecto. Se lee el archivo de Excel de configuración y se guardan las rutas de carpetas y archivos a utilizar, así como también otro tipo de información como lo pueden ser las credenciales de acceso a un sistema.

Ubicación: \Inicialización.xaml

Parámetros:

Nombre	Tipo	Descripción
io_Config	InOutArgument(scg:Dictionary(x:String, x:Object))	Rutas de archivos y carpetas

5.3 Filtrar.xaml

El propósito de este workflow es filtrar la información del archivo de seguimiento por fechas para posteriormente extraer dicha información y almacenarla correctamente. Para ello se utilizan dos variables que contienen la fecha de inicio y final del mes anterior al actual. Además de esto, se utilizan dos argumentos de tipo DataTable para almacenar la información y pasarla al workflow principal.

Ubicación: \Filtrar.xaml

Parámetros:

Nombre	Tipo	Descripción
io_MenoresDt	InOutArgument(sd:DataTable)	Información proveniente del archivo de seguimiento
io_MayoresDt	InOutArgument(sd:DataTable)	Información proveniente del archivo de seguimiento
in_Config	InArgument(scg:Dictionary(x:String, x:Object))	Rutas de archivos y carpetas

5.4 Seccionamiento_de_Columnas.xaml

El propósito de este workflow es establecer las tablas de datos a utilizar a lo largo de la ejecución del proyecto y la información que estas contienen.

Este seccionamiento es imprescindible para poder procesar y tratar cada sección de información de manera independiente según sea necesario, ya que no todas van a recibir el mismo procesamiento. Además, el mismo tipo de información se obtiene de columnas diferentes del archivo de seguimiento según su clasificación en “Menores a 100” o “Mayores a 100”.

Este workflow almacena toda la información del mismo tipo y clasificada por secciones en cada uno de los argumentos listados, para posteriormente ser transmitidos a los diferentes workflows de todo el proyecto.

Ubicación: \Seccionamiento de Columnas.xaml

Parámetros:

Nombre	Tipo	Descripción
in_MayoresDt	InArgument(sd:DataTable)	Información proveniente del archivo de seguimiento.
in_MenoresDt	InArgument(sd:DataTable)	Información proveniente del archivo de seguimiento.
in_HayMayores	InArgument(x:Boolean)	Utilizado para definir el seccionamiento.
out_PreTT2Dt	OutArgument(sd:DataTable)	Valores a escribir antes de los valores del archivo TT2
out_PostTT2Dt	OutArgument(sd:DataTable)	Valores a escribir después de los valores del archivo TT2
out_CodDaneDt	OutArgument(sd:DataTable)	Valores del municipio de cada solicitud
out_FechaPDt	OutArgument(sd:DataTable)	Valores de fecha de puesta

		en operación de cada solicitud
out_NivTensionDt	OutArgument(sd:DataTable)	Valores de nivel de tensión de cada solicitud

5.5 Procesar Datos Pre TT2.xaml

El propósito de este workflow es tratar la información extraída del archivo de seguimiento, concretamente las columnas “Tipo de Generación”, “Cuenta existente” y “NOMBRE PROYECTO / CLIENTE”.

Para el caso del tipo de generación, es necesario convertir este valor en su equivalente numérico establecido por el Área de Gestión Comercial. Mientras que para el caso de las otras dos se realiza un proceso de limpieza en los valores, eliminando la información innecesaria para el reporte.

Ubicación: \Procesar Datos Pre TT2.xaml

Parámetros:

Nombre	Tipo	Descripción
io_PreTT2Dt	InOutArgument(sd:DataTable)	Valores a escribir antes de los valores del archivo TT2

5.6 Procesar Nivel de Tensión.xaml

El propósito de este workflow es convertir la información extraída del archivo de seguimiento, concretamente la columna “Nivel de tensión” al equivalente numérico natural, ya que estos valores se encuentran en números romanos.

Ubicación: \Procesar Nivel de Tensión.xaml

Parámetros:

Nombre	Tipo	Descripción
io_NivTensionDt	InOutArgument(sd:DataTable)	Valores de nivel de tensión de cada solicitud

5.7 Codigo Dane.xaml

El propósito de este workflow es convertir la información extraída del archivo de seguimiento, concretamente de la columna “Municipio” al equivalente según el código establecido por el DANE. Para ello, se utiliza el archivo de Excel “Encargados por Zona” que se encuentra dentro de la carpeta “Recursos” del proyecto.

Para realizar correctamente este proceso fue necesario estandarizar la información de cada municipio, es decir, antes de compararlo con el valor de municipio encontrado en el archivo de encargados se eliminaban todas las tildes y caracteres especiales que la palabra pudiera contener. También se transformaba todo el municipio a mayúsculas. Una vez realizado este proceso, se lee la información del archivo de encargados que contiene el código correspondiente a cada municipio y se le asigna este nuevo valor a cada campo del DataTable “in_CodDaneDT”

Ubicación: \Procesar Código DANE.xaml

Parámetros:

Nombre	Tipo	Descripción
io_CodigoDaneDT	InOutArgument(sd:DataTable)	Valores del municipio de cada solicitud

5.8 Procesar Datos Post TT2.xaml

El propósito de este workflow es convertir la información extraída del archivo de seguimiento, concretamente la columna “Estado Solicitud” al equivalente numérico establecido por el Área de Gestión Comercial.

En este workflow también se añaden dos columnas nuevas al DataTable, estas columnas son requeridas para el reporte TT8 y es posible diligenciarlas utilizando la información almacenada hasta este punto del proceso.

Ubicación: \Procesar Datos Post TT2.xaml

Parámetros:

Nombre	Tipo	Descripción
io_PostTT2Dt	InOutArgument(sd:DataTable)	Valores a escribir después de los valores del archivo TT2

5.9 Generar Reporte TT8.xaml

Este es el workflow que finaliza el proceso, consiste en escribir toda la información almacenada en las diferentes variables a lo largo de la ejecución del proyecto en el archivo de plantilla TT8. Primero se escriben todas las variables por orden de las columnas requeridas por el documento, después, se invoca el workflow 5.10. Y finalmente se verifica si existen proyectos clase 1, en cuyo caso se escribirá en la plantilla TT8, caso contrario finalizará el proceso.

Ubicación: \Generar Reporte TT8.xaml

Parámetros:

Nombre	Tipo	Descripción
in_PreTT2Dt	InArgument(sd:DataTable)	Valores a escribir antes de los valores del archivo TT2. Previamente procesados.
in_FechaPDt	InArgument(sd:DataTable)	Valores de fecha de puesta en operación de cada solicitud
in_CodDaneDt	InArgument(sd:DataTable)	Valores del código DANE de cada solicitud. Previamente procesados.
in_PostTT2Dt	InArgument(sd:DataTable)	Valores a escribir después de los valores del archivo TT2. Previamente procesados.
in_NivTensionDt	InArgument(sd:DataTable)	Valores de nivel de tensión de cada solicitud. Previamente Procesados.
in_Config	InArgument(scg:Dictionary(x:String, x:Object))	Rutas de archivos y carpetas

5.10 Escribir Información TT2.xaml

El propósito de este workflow es unir la información proveniente del archivo TT2 con los registros de solicitudes correspondientes. Para ello, se utiliza el código NIU que se encuentra en los datos extraídos del archivo de seguimiento, este código también se encuentra en el archivo TT2.

Con este dato en común es posible realizar un proceso de fusión de las dos tablas de datos para poder juntar la información correctamente, una vez fusionadas, es necesario dividir esta tabla de datos en dos nuevamente, para facilitar el proceso de escritura en el documento Excel (ya que los datos se encuentran interrumpidos por la columna “Nivel de Tensión”). Esto se hace creando dos variables “InicioTT2” y “FinTT2” que posteriormente son escritas en la Plantilla TT8.

Ubicación: \Escribir Información TT2.xaml

Parámetros:

Nombre	Tipo	Descripción
in_CodigosCuentas	InArgument(sd:DataTable)	Valores a escribir antes de los valores del archivo TT2. Se realiza un procesamiento durante el workflow.
in_Config	InArgument(scg:Dictionary(x:String, x:Object))	Rutas de archivos y carpetas

6 Árbol de Invocación de Workflows

Nombre	Invoca	Es invocado en
\Main.xaml	<ul style="list-style-type: none"> •Iniciación.xaml •Filtrar.xaml •Seccionamiento de Columnas.xaml •Procesar Datos Pre TT2.xaml •Procesar Nivel de Tensión.xaml •Procesar Código DANE.xaml •Procesar Datos Post TT2.xaml •Generar Reporte TT8.xaml 	
\Iniciación.xaml		•Main.xaml
\Filtrar.xaml		<ul style="list-style-type: none"> •Main.xaml •Pruebas Unitarias\Caso de Prueba Filtrar.xaml
\Seccionamiento de Columnas.xaml		<ul style="list-style-type: none"> •Main.xaml •Pruebas Unitarias\Caso de Prueba Seccionamiento de Columnas.xaml
\Procesar Datos Pre TT2.xaml		<ul style="list-style-type: none"> •Main.xaml •Pruebas Unitarias\Caso de Prueba Procesar Datos Pre TT2.xaml
\Procesar Nivel de Tensión.xaml		<ul style="list-style-type: none"> •Main.xaml •Pruebas Unitarias\Caso de Prueba Procesar Nivel de Tensión.xaml
\Procesar Código DANE.xaml		<ul style="list-style-type: none"> •Main.xaml •Pruebas Unitarias\Caso de Prueba Código DANE.xaml
\Procesar Datos Post TT2.xaml		<ul style="list-style-type: none"> •Main.xaml •Pruebas Unitarias\Caso de Prueba Procesar Datos Post TT2.xaml
\Generar Reporte TT8.xaml	•Escribir Información TT2.xaml	<ul style="list-style-type: none"> •Main.xaml •Pruebas Unitarias\Caso de Prueba Generar Reporte TT8.xaml
\Escribir Información TT2.xaml		<ul style="list-style-type: none"> •Generar Reporte TT8.xaml •Pruebas Unitarias\Caso de Prueba Escribir Información TT2.xaml
\Pruebas Unitarias\Caso de Prueba Filtrar.xaml	•Filtrar.xaml	
\Pruebas Unitarias\Caso de Prueba Seccionamiento de Columnas.xaml	•Seccionamiento de Columnas.xaml	

\Pruebas Unitarias\Caso de Prueba Procesar Datos Pre TT2.xaml	•Procesar Datos Pre TT2.xaml	
\Pruebas Unitarias\Caso de Prueba Procesar Nivel de Tensión.xaml	•Procesar Nivel de Tensión.xaml	
\Pruebas Unitarias\Caso de Prueba Código DANE.xaml	•Procesar Código DANE.xaml	
\Pruebas Unitarias\Caso de Prueba Procesar Datos Post TT2.xaml	•Procesar Datos Post TT2.xaml	
\Pruebas Unitarias\Caso de Prueba Generar Reporte TT8.xaml	•Generar Reporte TT8.xaml	
\Pruebas Unitarias\Caso de Prueba Escribir Información TT2.xaml	•Escribir Información TT2.xaml	

7 Dependencias del Proyecto

Estos son los paquetes y las versiones utilizadas en UiPath Studio para el desarrollo del proyecto:

- UiPath.Excel.Activities: 2.16.2
- UiPath.Mail.Activities: 1.18.2
- UiPath.System.Activities: 22.10.4
- UiPath.Testing.Activities: 22.10.3
- UiPath.UIAutomation.Activities: 22.10.5

8 Riesgos y Puntos de Atención.

Aquí se listan los errores y puntos de atención identificados durante el periodo de desarrollo y pruebas del proyecto:

- **Excel Process Scope: Exception from HRESULT: 0x800AC472:** Este error surgió al momento de interactuar con diferentes archivos de Excel, para solucionarlo fue necesario modificar las propiedades de diferentes actividades para que al momento de abrir un archivo nuevo se cerraran todos los demás procesos de Excel que se encontrasen corriendo en ese momento. Es posible encontrar información acerca del error en los [Foros de UiPath](#)
- **Errores de compatibilidad NET (Windows Legacy-Windows):** Este error surgió al momento de desplazar el proyecto de un equipo a otro debido a que su estructura había sido modificada de Windows-Legacy a Windows. Si bien es necesario empezar a desarrollar los nuevos proyectos con compatibilidad Windows, para el caso de este proyecto se mantuvo la versión Legacy. Para convertir un proyecto de compatibilidad Windows a Legacy se utilizó la información encontrada en [este foro](#). En caso de querer convertir un proyecto de Legacy a Windows es posible hallar más información [aquí](#).