

Esquema de los documentos necesarios para la validación de los datos recibidos desde un formulario de RIT

Introducción

Desde la Dirección de Inclusión Productiva, se desarrolló un Script en Python capaz de validar la información que se recolecta a través de los formularios diligenciados a través del RIT. Esto con el objetivo de poder optimizar el proceso de validación de la información recogida.

Con esto en cuenta, el script desarrollado desde Python, es capaz de validar la información de cualquier formulario diligenciado desde RIT a partir de tres archivos:

- Malla de validación de la encuesta en formato JSON
- Malla de validación de la encuesta en formato Excel
- Archivo de configuración de acceso al API de RIT

A partir de estos archivos, el Script es capaz de acceder a la información en tiempo real y validar la consistencia de la información recogida. A continuación se va a detallar el contenido de estos archivos junto a la estructura requerida para su funcionamiento.

Malla de Validación en formato JSON

El Script de Python de Malla de Validación, valida la información recogida desde RIT a partir de los parámetros establecidos en la encuesta, así pues, en un archivo de tipo [JSON](#) se almacenan las características del formulario, el Script de Python toma la información de este archivo y valida los datos según los criterios definidos en este archivo.

Esquema de la malla de validación en formato JSON

Para entender la estructura del archivo de Malla de Validación, es importante identificar varios conceptos:

- **Pregunta Condicionada:** Se refiere a una pregunta que debe ser contestada únicamente si una pregunta anterior contiene una respuesta específica. Por ejemplo, la pregunta Dirección Rural del Hogar debe ser contestada únicamente si en la pregunta ZONA de ubicación del Hogar es "Rural Disperso".
- **Valores de la Pregunta:** En el caso que una pregunta se deba contestar a partir de una serie de valores predefinidos, estos son las posibles respuestas que puede tener la pregunta.
- **Tipos de entrada:** El Aplicativo RIT permite múltiples alternativas de generación de una pregunta, existen los métodos de entrada de texto, selección de variables, selección múltiple de variables, múltiples entradas de texto, entre otras.

La estructura del Archivo JSON es el siguiente:

- Nombre de la variable: Cómo su nombre lo indica, se refiere al nombre asociado en el conjunto de datos a la pregunta del formulario RIT.
- **"condicion"**: Es un objeto de tipo JSON que contiene la información acerca de las variables que condicionan la pregunta, si no hay condiciones el valor será **None**, caso contrario el objeto contendrá la siguiente información:

- Variable Dependiente: Se refiere a la variable cuya respuesta es necesaria para habilitar la pregunta. Es un objeto JSON cuyo atributo es el nombre de la variable y el valor será una lista donde se almacenan las respuestas que deben ocurrir para habilitar la pregunta.
- "valores": Es un objeto de tipo JSON que contiene la información acerca de las posibles valores que debe tomar la pregunta, si no hay valores predefinidos, el objeto será **None**, caso contrario el objeto contendrá la siguiente información:
 - "valor": Es un objeto de tipo lista donde se almacenan las respuestas que puede tomar la pregunta.
 - "Tipo": Es una cadena de texto que indica el formato de la respuesta según el tipo de valor que puede tomar la pregunta.
- "iand": contiene una respuesta de tipo booleano (False, True) en donde, si una variable depende de dos o más variables para su habilitación, se determina si se debe cumplir todas las condiciones al tiempo, o solamente se debe cumplir al menos una de las condiciones (valor lógico de tipo **and** o de tipo **or**).
- "opcional": contiene una respuesta de tipo booleano donde se determina si la respuesta de la pregunta es obligatoria u opcional.
- "excluida_PTA": Según los criterios de algunos programas de la Dirección de Inclusión Productiva, es necesario que el participante disponga de acceso a agua y disponga de un espacio de tierra para poder participar en el respectivo programa, con esto en cuenta, este campo contiene un booleano que determina si la pregunta debe ser excluida del análisis de estas variables o si por el contrario solo es contestada si se cumplen las condiciones definidas.

Ejemplo del Esquema de la malla de validación de tipo JSON

```
{
  "idencuesta": {
    "condicion": null,
    "valores": null,
    "iand": false,
    "opcional": false,
    "excluida_PTA": true
  },
  "ZONA": {
    "condicion": null,
    "valores": {
      "valor": [1,2,3],
      "Tipo": "int"
    },
    "iand": false,
    "opcional": false,
    "excluida_PTA": true
  },
  "Direccion_Rural":{
    "condicion": {
      "ZONA": [3],
      "TipoEstructura": [3]
    },
    "valores": {
      "valor": "^.{3,100}$",
      "Tipo": "regex"
    }
  }
}
```

```
    },  
    "iand": true,  
    "opcional": false,  
    "excluida_PTA": true  
  }  
}
```

Comentarios sobre la malla de validación tipo JSON

El archivo de tipo JSON es el archivo principal de validación de los formularios de RIT, en el caso que el proceso de creación de este archivo sea muy complejo, se ha dispuesto un archivo excel donde se puede definir la estructura de la malla, el Script de Python se encargará de crear el archivo de tipo JSON.

Ruta de exportación del archivo

Dentro de la carpeta del proyecto, este archivo debe ser almacenado en la carpeta `data/validation_json/`, esta carpeta está diseñada para almacenar todos los archivos de malla de validación de formato json, el script de Python va leer la malla de validación de los datos que se encuentran en esta carpeta.

Malla de Validación en formato Excel

El archivo principal que toma la estructura de la encuesta y su validación es la malla de tipo JSON, sin embargo, para facilitar este proceso se diseñó un archivo de tipo excel donde se pueden almacenar las diferentes validaciones que se deben realizar de una forma más sencilla y amigable con el usuario.

Este archivo Excel está compuesto por dos hojas:

- **Validaciones:** La hoja validaciones contiene la información relacionada con la estructura de la pregunta, es decir, contiene todas las variables del formulario, contiene las variables de las que depende la pregunta, contiene las respuestas que deben tener las preguntas de condición, el tipo de valor lógico para múltiples condiciones, la información sobre el estado obligatorio u opcional de la pregunta y la información sobre su exclusión o inclusión en la validación de la condición general.
- **Valores:** Como su nombre lo indica, esta hoja contiene la información únicamente de las variables sobre las que se debe validar el tipo de respuesta recibida y contiene además, el tipo de valor que se debe validar.

Esquema de la malla de validación en formato Excel

Hoja Validación

En la hoja validación se encuentran las siguientes columnas:

- **variable:** Contiene el nombre de la variable relacionada con la pregunta, en el caso que la variable contenga dos o más condiciones, se debe duplicar el registro de la variable por cada condición.
- **dependiente:** Nombre de la variable que condiciona a la variable a verificar.
- **condicion:** Contiene los valores que debe tener la variable que condiciona para habilitar la pregunta, en el caso que haya dos o más valores, estos irán concatenados en una cadena de texto sin espacio y separados por el carácter "|". Por ejemplo, si para una pregunta es necesario que hayan contestado "SI", "NO " ó " NO SE", el valor en la celda debe ser **SI|NO | NO SE**.

- **tipo_validacion**: Se refiere al tipo de valor que debe tomar los valores de la columna **condicion**, este puede ser de tipo str (Cadena de Texto) o int (Número Entero), debido a la dualidad de la variable, solo se debe marcar en la variable si es de tipo entero, en el caso que el valor de la celda este vacío, por defecto el script de python entenderá que es una cadena de texto.
- **iand**: Se refiere al valor lógico que une las multiples condiciones para la variable, al igual que el caso anterior, si se deben cumplir todas las condiciones (and) el valor de la celda en el primer registro de la variable se debe marcar como "SI", en el caso que se debe cumplir al menos una de las condiciones (or) el valor de la celda para el primer registro de la variable debe estar vacío y el script de Python lo reconocerá como tipo or.
- **excluye_pta**: Se refiere al valor booleano que indica si la variable se excluye o no de las condiciones generales de participación, al igual que la columna anterior, en el caso que sea exluida, el valor de la celda en el primer registro de la variable se debe marcar como "SI", caso contrario se debe dejar vacía y Python la entenderá como no exluida.
- **variable_opcional**: Se refiere al valor booleano que indica si la respuesta de la pregunta es obligatoria o no, de manera analoga a la columna anterior, si es opcional se debe marcar como "SI" en el primer registro de la variable, en el caso que sea obligatoria, este se debe dejar vacío y Python la entenderá como una variable obligatoria.

Ejemplo de la estructura de la hoja validaciones

Tomando los mismos valores de ejemplo de la malla de validación de tipo JSON, la hoja de validación tendría la siguiente estructura

variable	dependiente	condicion	tipo_validacion	iand	excluye_pta	variable_opcional
idencuesta					SI	
ZONA					SI	
Direccion_Rural	ZONA	2 3	int	SI	SI	
Direccion_Rural	TipoEstructura	3	int		SI	

Hoja Valores

En la hoja validación se encuentran las siguientes columnas:

- **variable**: Contiene el nombre de la variable relacionada con la pregunta, en el caso que la variable no contenga valores a validar, esta no debe ser almacenada en esta hoja.
- **valores**: Contiene los valores que debe tener la variable que condiciona para habilitar la pregunta, en el caso que haya dos o más valores, estos irán concatenados en una cadena de texto sin espacio y separados por el caracter "|". Por ejemplo, si para una pregunta es necesario que hayan contestado "SI", "NO " ó " NO SE", el valor en la celda debe ser **SI|NO | NO SE**, en el caso que los valores que puede tomar la variable son de respuesta abierta pero con una condición definida, se va a utilizar una cadena de texto con una **expresion regular** que determina los valores que puede tomar la pregunta.
- **tipo_valor**: Se refiere al tipo de valor que debe tomar los valores, se pueden seleccionar entre 4 opciones int, str, list (En el caso que la pregunta sea de selección multiple y por ende su resultado será una lista con las respuestas seleccionadas) o listlist (Se refiere al caso que la pregunta permita realizar una selección múltiple para varias respuestas, por ende el resultado será una lista que contiene las listas de respuestas seleccionadas).

Ejemplo de la estructura de la hoja valores

Tomando los mismos valores de ejemplo de la malla de validación de tipo JSON, la hoja de valores tendría la siguiente estructura

variable	valores	tipo_valor
ZONA	1 2 3	int
Direccion_Rural	"^{3,100}\$"	regex

Comentarios sobre la malla de validación tipo Excel

Una vez se ha definido la estructura de la malla de validación en el libro Excel, es necesario correr la función `create_json_malla` qué será la encargada de crear el archivo de malla de validación de tipo JSON.

Ruta de exportación del archivo

Dentro de la carpeta del proyecto, este archivo debe ser almacenado en la carpeta `data/validation_excel/`, esta carpeta está diseñada para almacenar todos los archivos de malla de validación de formato excel, el script de Python va leer la malla de validación de tipo excel de los datos que se encuentran en esta carpeta para crear el archivo de tipo json.

Esquema para la configuración del archivo de acceso al API del RIT

Los datos que van a ser analizados y validados, están siendo consumidos en tiempo real desde el API habilitado desde RIT. En ese sentido, es necesario definir los parámetros de acceso a la información que se encuentra en el API, o de lo contrario, el script no será capaz de poder acceder a ningún tipo de información y a una validación de datos erronea.

Con esto en cuenta, se requiere la creación de un archivo de tipo JSON que contenga los parámetros requeridos para el acceso al API y su respectivo consumo de información.

Esquema del archivo de acceso

El archivo de acceso debe ser de formato `JSON` y debe contener los siguientes campos:

- `"url"`: Cadena de texto que representa la URL de la API. Debe ser una URL válida y completa.
- `"method"`: Una cadena que representa el método HTTP utilizado para la solicitud. El valor de la solicitud debe ser "GET"
- `"headers"`: Un objeto JSON que contiene encabezados de la solicitud. Dentro de este objeto se debe añadir lo siguiente:
 - `"Authorization"`: Una cadena de texto que representa la clave de acceso al API, esta debe tener el formato "Bearer token-acceso".

Ejemplo

La estructura y valores del archivo de acceso debe ser de la siguiente manera

```
{
  "url": "https://api.example.com/data",
```

```
"method": "GET",  
"headers": {  
  "Authorization": "Bearer token-12345"  
}
```

Ubicación del Token

Dentro de la carpeta del proyecto, este archivo debe ser almacenado en la carpeta **config/**, esta carpeta está diseñada únicamente para almacenar las rutas de acceso a los diferentes formularios que se van a validar.