HOJA DE CONTROL

|  |  |  |  |
| --- | --- | --- | --- |
| **Organismo** | Banco Estado | | |
| **Proyecto** | Data-Lake | | |
| **Entregable** | Planes de Pruebas Unitarias | | |
| **Revisado por** | Felipe Ávalos | | |
| **Versión/Edición** | 0300 | **Fecha Versión** | 05/06/2018 |
| **Nombre regla** | RGL\_GEN\_RN009 | **Fecha prueba** | 05/06/2018 |
|  |  | **Nº Total de Páginas** | 05 |

REGISTRO DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión doc** | **Causa del Cambio** | **Responsable del Cambio** | **Fecha del Cambio** |
| 0100 | Versión inicial |  | 30/05/2018 |
| 0200 | Actualización |  | 04/06/2018 |
| 0300 | Actualización |  | 05/06/2018 |

CONTROL DE DISTRIBUCIÓN

|  |
| --- |
| **Nombre y Apellidos** |
| Nicolás Llanos |
| Felipe Ávalos |
| Miguel Choque |
| Marcos Toro |
|  |

[1 DEFINICIÓN DE LOS CASOS DE PRUEBAS 3](#_Toc515963211)

[2 ANEXOS 5](#_Toc515963212)

# DEFINICIÓN DE LOS CASOS DE PRUEBAS

**Función RN009 válida para reglas: RN009, RN010, RN011, RN017, RN012, RN016, RN013, RN014, RN015, RN055**

|  |  |  |
| --- | --- | --- |
| Prueba de función | Código caso de prueba: | PF-RN009 |
| **Descripción:**  Se invoca la regla programada aplicándola sobre un archivo de texto (preparado por el equipo de desarrollo), este archivo contiene datos válidos y no válidos, la función debe identificar cada caso y generar una columna adicional con los valores esperados para éxito y error. | | |
| **Prerrequisitos:**   * Encontrarse en ambiente de desarrollo Python-Spark. * Archivo txt de prueba con datos validos e inválidos. | | |
| **Pasos:**   1. Actualizar el archivo de pruebas con los datos necesarios de acuerdo a la especificación de la regla. 2. Ejecutar la regla sobre el archivo de pruebas. 3. Verificar la creación de la columna adicional con los resultados por fila. | | |
| **Resultado esperado en la columna generada por la función**:   1. Dato válido: 1 2. Dato invalido: 0 | | |
| **Resultado obtenido:**   1. archivo\_prueba.txt; col24: 0 (No era numérico de largo 1) y 1 (Numérico de largo 1) 2. DV: 0 (No era numérico de largo 1) y 1 (Numérico de largo 1) 3. col4: 0 (No era numérico de largo 2) y 1 (Numérico de largo 2) 4. col6: 0 (No era numérico de largo 3) y 1 (Numérico de largo 3) 5. col17: 0 (No era numérico de largo 15) 6. col1: 0 (No era numérico de largo 4) 7. col11: 0 (No era numérico de largo 11) y 1 (Numérico de largo 11) 8. col8: 0 (No era numérico de largo 5) y 1 (Numérico de largo 5) 9. col9: 0 (No era numérico de largo 9) y 1 (Numérico de largo 9) 10. col10: 0 (No era numérico de largo 10) y 1 (Numérico de largo 10) 11. col21: 0 (No era numérico de largo 7) 12. No se encontraron excepciones | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| Prueba de ejecución | Código caso de prueba: | PE-RN009 |
| **Descripción:**  Se invoca la regla programada aplicándola sobre un archivo de texto (propio del aplicativo) en ambiente HDFS, este archivo contiene datos propios del dominio, la función debe identificar cada caso y generar una columna adicional con los valores esperados para éxito y error. | | |
| **Prerrequisitos:**   * Encontrarse en ambiente pre productivo Python-Spark. * Archivo txt en ambiente HDFS con datos validos e inválidos. | | |
| **Pasos:**   1. Ejecutar la regla sobre el archivo de texto. 2. Verificar la creación de la columna adicional con los resultados por fila. | | |
| **Resultado esperado en la columna generada por la función**:   1. Dato válido: 1 2. Dato invalido: 0 | | |
| **Resultado obtenido:**   1. <Indicar el set de datos imputados>: <resultado obtenido> 2. <Excepciones> | | |

# ANEXOS

Este punto contendrá toda aquella información de interés para la elaboración y validación del Plan de Pruebas Funcionales.