**UNIVERSIDAD DE TARAPACÁ**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA CIVIL EN COMPUTACIÓN E INFORMÁTICA**

**ARICA – CHILE**



**Informe Final**

**“Herramienta de automatización de validación de consistencia e integridad de campos (AVCIC)”**

**Equipo de Desarrollo:**

**- Sebastian Lukich Aste**

**Empresa o Unidad:**

**Digital Social Change SPA**

**Curso: Proyecto IV ICCI**

**Profesor: Diego Aracena Pizarro**

Arica, 18 de noviembre del 2022

Contenido

[**1.** **Definición del proyecto** 5](#_Toc122254540)

[1.1. Contexto 5](#_Toc122254541)

[1.2. Problema 5](#_Toc122254542)

[1.3. Solución 6](#_Toc122254543)

[**2.** **Requisitos del sistema** 6](#_Toc122254544)

[2.1. Requisitos funcionales 6](#_Toc122254545)

[2.2. Requisitos no funcionales 7](#_Toc122254546)

[**3.** **Alcances del ambiente de software** 8](#_Toc122254547)

[**4.** **Subsistemas** 8](#_Toc122254548)

[**5.** **Objetivos** 9](#_Toc122254549)

[5.1. Objetivo general 9](#_Toc122254550)

[5.2. Objetivos específicos 9](#_Toc122254551)

[**6.** **Alcance** 9](#_Toc122254552)

[**7.** **Lista de actividades** 10](#_Toc122254553)

[7.1. Etapas y actividades dentro del desarrollo del proyecto 10](#_Toc122254554)

[7.2. Carta Gantt 10](#_Toc122254555)

[**8.** **Diagramas del proyecto** 11](#_Toc122254556)

[8.1. Diagrama de sistema del proyecto (alto nivel) 11](#_Toc122254557)

[8.2. Diagrama de contexto: sistema y subsistemas 11](#_Toc122254558)

[**9.** **Herramientas a utilizar** 12](#_Toc122254559)

[9.1. Pentaho Data Integration (PDI) 12](#_Toc122254560)

[9.2. Saiku Analytics 12](#_Toc122254561)

[**10.** **Alcance de acuerdo a las herramientas** 12](#_Toc122254562)

[**11.** **Aspectos iniciales del front-end** 13](#_Toc122254563)

[11.1. Ejemplo de gráfico de torta mostrando el porcentaje de error de cada campo respecto al total de errores en toda la “[Tabla 1]” 13](#_Toc122254564)

[11.2. Ejemplo de gráfico de barras mostrando el total de errores encontrados por sistema en decremento con el aumento de iteraciones 13](#_Toc122254565)

[11.3. Ejemplo de gráficos de barras donde la cantidad de errores disminuye con las iteraciones por tabla dentro de un “[Sistema 1]” 14](#_Toc122254566)

[**12.** **Business Process Management (BPM)** 14](#_Toc122254567)

[**13.** **Repositorio de Github** 16](#_Toc122254568)

[**14.** **Almacenamiento y modelamiento de los datos** 17](#_Toc122254569)

[**15.** **Implementación del sistema** 18](#_Toc122254570)

[15.1. Módulo con Pentaho 18](#_Toc122254571)

[15.2. Módulo con Saiku Analytics 23](#_Toc122254572)

[**16.** **Análisis de resultados** 28](#_Toc122254573)

[**17.** **Sugerencias de mejoramiento** 29](#_Toc122254574)

[**18.** **Conclusiones** 29](#_Toc122254575)

[**19.** **Bibliografía** 30](#_Toc122254576)

**Introducción**

La empresa Digital Social Change Spa se dedica a la transformación digital de empresas principalmente en la asesoría e implementación de automatización relacionadas al procesamiento masivo de datos.

Digital Social Change Spa requiere implementar una herramienta de apoyo/complementaria para el proceso de validación de migración de datos. Esta herramienta es requerida puesto que al realizar una migración la cantidad de campos que se deben validar es muy alta para poder realizar una validación manual. Dado lo anterior se requiere diseñar y construir 2 modulos:

1. Proceso de Captura de Información (Back-End 20%).

Capturar el detalle de los errores.

1. Proceso de Seguimiento (Front-End 80%)

Visualización por iteraciones de los errores.

Para lo anterior se utilizará tecnologías de inteligencia de negocios como los cubos multidimensionales y los visores OLAP.

## **Definición del proyecto**

### Contexto

La captura de información es el proceso en el cual se sacan los datos de error de una tabla que ha sido migrada de un sistema a otro. Por ejemplo, si así fuere que la tabla tiene un campo “Sexo”, que después de la migración terminó con todos sus registros en ese campo con los valores “M”, “F” y “5”, entonces el detalle de ese error sería que en el campo “Sexo” de la tabla en cuestión hay un valor “5” fuera del conjunto válido de valores. Este detalle se registraría en otro conjunto de datos que suele estar actualizándose periódicamente por iteración.

Cuando el proceso no está automatizado, todos los recursos humanos disponibles en el área deben dedicarse a hacer consultas SQL a mano sobre cada campo de la tabla migrada. Lo que asciende a varios días de trabajo en el proyecto y más el hecho de que hay varias tablas por sistema y que se produce un informe de los errores que se encontraron por tabla puede derivar en que la tarea neta se haga inviable.

El proceso de seguimiento de la información es el que se encarga de dar seguimiento a estadísticas de errores que dejó como resultado el proceso distinto de captura de información. Para esto se presta de gráficos, tablas, diagramas y ayudas visuales. Las estadísticas evolucionan con el avance de cada iteración que tiene el proyecto, y las visualizaciones están hechas para conectar la información de una dimensión del modelo de datos del proyecto con las demás. Todo dentro de un control de mandos, conocido también como Dashboard.

### Problema

La empresa Digital Social Change SPA no puede hacerse cargo, simplemente con recursos humanos, del trabajo de captura de la información para todos los campos del total de tablas de cada sistema migrado. Necesita de herramientas automatizadas, que apoyen en los procesos tanto de captura de información de error, en lo que respecta a consistencia e integridad de los campos de sus tablas migradas, así como de seguimiento de la información de error de estos procesos de captura.

### Solución

Implementar a través de software de inteligencia de negocios una herramienta con dos módulos. Uno que automatice el proceso de captura de información de error, en lo que respecta a consistencia e integridad de los campos de las tablas migradas, y otro que, como complemento al anterior, haga un seguimiento que se pueda visualizar de la evolución por iteraciones de la información capturada por el proceso anterior.

## **Requisitos del sistema**

### 2.1. Requisitos funcionales

|  |  |
| --- | --- |
| Requisitos funcionales | |
| Código | Descripción |
| RF-01 | El sistema deberá mostrar un diagrama con los nombres de los sistemas seleccionados para la importación incluyendo para cada uno la visualización de los datos: cantidad de campos migrados correctamente, cantidad de campos con error en la migración, cantidad de campos totales, porcentaje de campos migrados correctamente, porcentaje de campos con error. |
| RF-02 | El sistema deberá mostrar un diagrama con los nombres de los subsistemas seleccionados para la importación incluyendo para cada uno la visualización de los datos: cantidad de campos migrados correctamente, cantidad de campos con error en la migración, cantidad de campos totales, porcentaje de campos migrados correctamente, porcentaje de campos con error. |
| RF-03 | El sistema deberá mostrar un diagrama con los nombres de las tablas seleccionadas para la importación incluyendo para cada una la visualización de los datos: cantidad de campos migrados correctamente, cantidad de campos con error en la migración, cantidad de campos totales, porcentaje de campos migrados correctamente, porcentaje de campos con error, cantidad de registros migrados correctamente, cantidad de registro con error en la migración, cantidad de registros totales, porcentaje de registros migrados correctamente, porcentaje de registros con error. |
| RF-04 | El sistema deberá mostrar un diagrama con los nombres de los campos seleccionados para la importación incluyendo para cada una la visualización de los datos: cantidad de registros migrados correctamente, cantidad de registro con error en la migración, cantidad de registros totales, porcentaje de registros migrados correctamente, porcentaje de registros con error. |
| RF-05 | El sistema deberá mostrar un diagrama con los identificadores de los tipos de error genérico que aparecen a lo largo de los registros de los sistemas seleccionados para la importación incluyendo para cada uno la visualización de los datos: cantidad de registro con error en la migración. |
| RF-06 | El sistema deberá mostrar un diagrama con las descripciones de los tipos de error detallado que aparecen a lo largo de los registros de los sistemas seleccionados para la importación incluyendo para cada una la visualización de los datos: cantidad de registro con error en la migración. |
| RF-07 | El sistema deberá mostrar un diagrama con las fechas de las iteraciones en las que se efectúa una mejora en el algoritmo de migración de campos de algún sistema, incluyendo para cada una la visualización de los datos: cantidad de campos migrados correctamente, cantidad de campos con error en la migración, cantidad de campos totales, porcentaje de campos migrados correctamente, porcentaje de campos con error, cantidad de registros migrados correctamente, cantidad de registro con error en la migración, cantidad de registros totales, porcentaje de registros migrados correctamente, porcentaje de registros con error. |
| RF-08 | El sistema deberá mostrar un diagrama con las horas de las iteraciones que ocurrieron en una fecha en las que se efectúa una mejora en el algoritmo de migración de campos de algún sistema, incluyendo para cada una la visualización de los datos: cantidad de campos migrados correctamente, cantidad de campos con error en la migración, cantidad de campos totales, porcentaje de campos migrados correctamente, porcentaje de campos con error, cantidad de registros migrados correctamente, cantidad de registro con error en la migración, cantidad de registros totales, porcentaje de registros migrados correctamente, porcentaje de registros con error. |
| RF-09 | El sistema deberá mostrar un diagrama con los números de orden de las iteraciones en las que se efectúa una mejora en el algoritmo de migración de campos de algún sistema, incluyendo para cada una la visualización de los datos: cantidad de campos migrados correctamente, cantidad de campos con error en la migración, cantidad de campos totales, porcentaje de campos migrados correctamente, porcentaje de campos con error, cantidad de registros migrados correctamente, cantidad de registro con error en la migración, cantidad de registros totales, porcentaje de registros migrados correctamente, porcentaje de registros con error. |
| RF-10 | El sistema permitirá traslados de un diagrama a otro, siempre que cumplan con la relación de estar en un nivel de abstracción inferior a la categoría principal del primer diagrama. Como pasar del diagrama sobre “sistemas” al diagrama de la categoría inferior “subsistemas”, a partir de la elección del usuario de qué sistema del conjunto se desea observar en más detalle. |
| RF-11 | El sistema permitirá filtros en los diagramas de los requisitos funcionales de 1 al 10 que permitirán la personalización en la navegación del usuario por las visualizaciones de los datos que dispone el sistema. |
| RF-12 | El sistema automatizará la captura de la información de errores contenida en las tablas que pasaron por un proceso de migración. |

### 2.2. Requisitos no funcionales

|  |  |
| --- | --- |
| Requisitos no funcionales | |
| Código | Descripción |
| RNF-01 | El sistema deberá utilizar una base de datos para almacenar los datos sobre los errores de la validación de registros de sistemas destinados a migración. |
| RNF-02 | El sistema deberá utilizar una herramienta de visor OLAP del grupo Pentaho Saiku o STPivot de Stratebi. |

## **Alcances del ambiente de software**

* Automatización del proceso de captura del detalle de los errores.
* Visibilidad de estadísticas de errores de migración en distinto nivel de detalle.
* Visibilidad de estadísticas de errores de migración en distinta representación gráfica.

## **Subsistemas**

* Subsistema de captura de información
* Subsistema de seguimiento
* Subsistema de base de datos de errores capturados

## **Objetivos**

### Objetivo general

Producir un módulo para un proceso automático de captura de información de errores de migración y otro que sirva de tablero para la visualización del seguimiento de los errores capturados en las migraciones de sistemas que ejecuta la empresa Digital Social Change.

### Objetivos específicos

1. Desarrollar un software de captura automatizada de datos de errores producidos en las migraciones de sistemas migrados por Digital Social Change Spa.
2. Estudiar la estructura de almacenamiento de los datos de los errores registrados por el proceso automático de validación de campos de tablas migradas con la que funcionará el tablero de estadísticas de evolución de estos mismos errores.
3. Desarrollar el diseño de las vistas con las que se visualizará la evolución de los errores en el tablero de estadísticas de evolución de errores generados por las validaciones de los campos de las tablas de los distintos sistemas y subsistemas sometidas a proceso de migración.
4. Desarrollar por software un tablero de estadísticas que permita ver la evolución de los errores generados por las validaciones de los campos de las tablas de los distintos sistemas y subsistemas que fueron sometidas a proceso de migración.
5. Realizar las pruebas de funcionamiento y análisis de resultados sobre el software.

## **Alcance**

Proveer a la empresa Digital Social Change Spa de un software que automatice las tareas de hacer las validaciones a mano campo por campo para las tablas de los sistemas que pasan por migración en la empresa, y de un software que muestre el avance gráfica y dinámicamente de estas validaciones conforme va mejorando el método de migración de sistemas.

## **Lista de actividades**

### Etapas y actividades dentro del desarrollo del proyecto

El proyecto constará de las siguientes etapas que a su vez se subdividen en actividades dentro de la planificación:

**Inauguración del proyecto:**

“Establecer la metodología de trabajo (acuerdo con cliente)”, “Elegir horarios de reuniones (acuerdo con cliente)“, “Entender el trasfondo de la empresa y su necesidad”.

**Diseño del proyecto:**

“Buscar trabajos relacionados y reportar experiencias anteriores (de la empresa)”, “Establecer listado de requisitos (acuerdo firmado)”, “Hacer diagrama en general del sistema”, “Hacer modelo estrella”, “Hacer diagrama de contexto”.

**Implementación:**

“Hacer pantallazos de primer prototipo”, “Llenado de tablas (estructura reportada por la empresa)”, “Boceto de concepto con Excel”, “Levantar servidor con Pentaho más base de datos”, “Desarrollar primer módulo con Pentaho”, “Desarrollar segundo módulo con Saiku”.

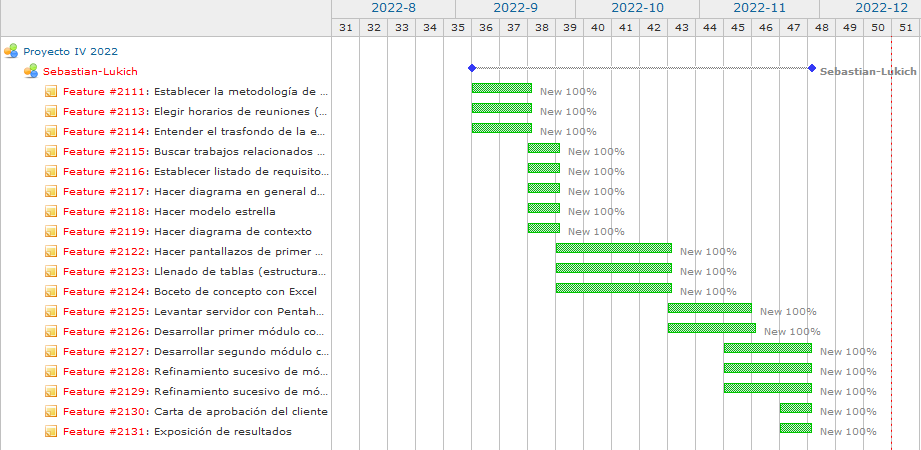
**Pruebas:**

“Refinamiento sucesivo de módulo de Pentaho”, “Refinamiento sucesivo de módulo de Saiku”.

**Cierre del proyecto:**

“Carta de aprobación del cliente”, “Exposición de resultados”.

### Carta Gantt



*Figura 1: Carta Gantt*

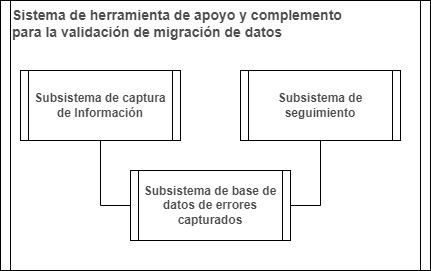
## **Diagramas del proyecto**

### Diagrama de sistema del proyecto (alto nivel)



*Figura 2: Diagrama de alto nivel*

### Diagrama de contexto: sistema y subsistemas



*Figura 3: Diagrama de contexto*

## **Herramientas a utilizar**

### Pentaho Data Integration (PDI)

Es una de las herramientas o componentes de Pentaho Suite que permite que se utilicen técnicas ETL, es decir, poder implementar procesos deextracción, transformación y carga de datos. PDI ofrece también datos analíticos muy precisos, eliminando las complejidades involucradas en la codificación al proporcionar bibliotecas en profundidad para el mismo.

### Saiku Analytics

Saiku permite a los usuarios comerciales explorar fuentes de datos complejas, utilizando una interfaz familiar de arrastrar y soltar y una terminología comercial fácil de entender, todo dentro de un navegador.

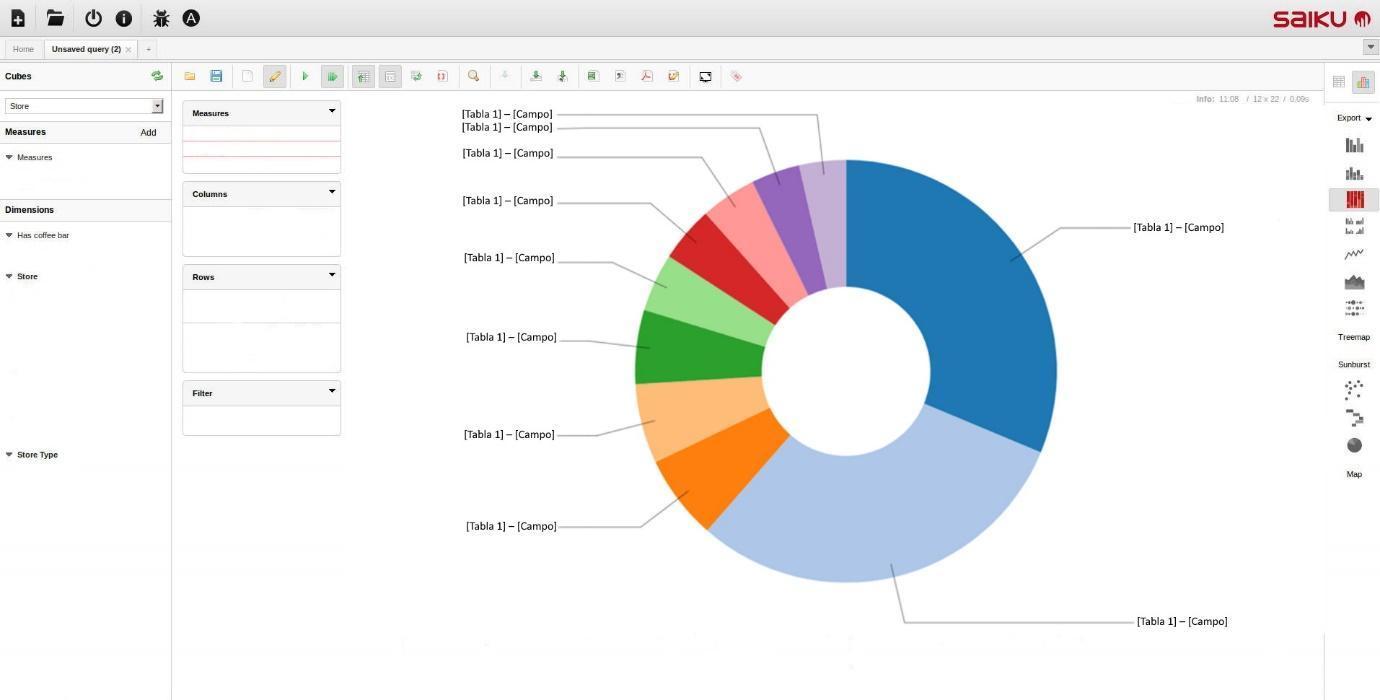
## **Alcance de acuerdo a las herramientas**

Automatizar la validación de campos de los sistemas de las migraciones utilizando Pentaho Data Integration. Crear un dashboard hecho en Saiku Analytics para mostrar gráfica y dinámicamente la evolución de las validaciones.

## **Aspectos iniciales del front-end**

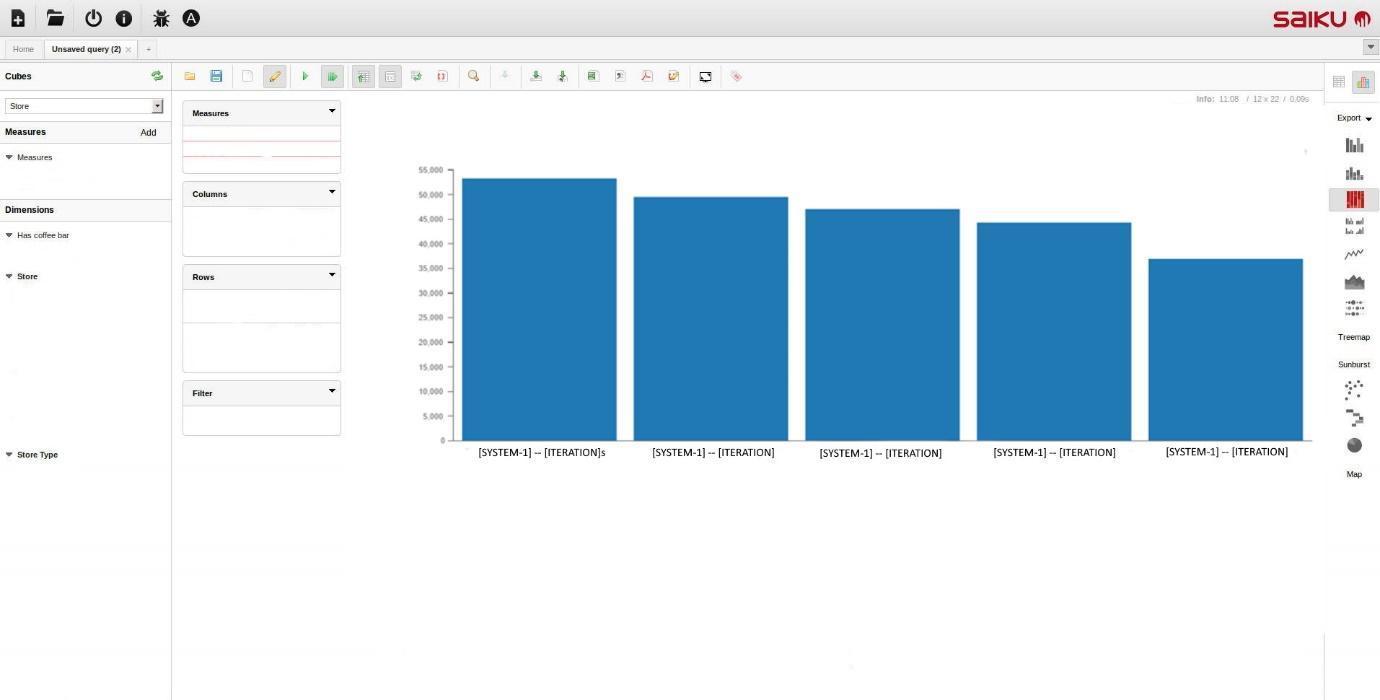
Los pantallazos que siguen corresponden a ejemplos tipo mockup y no con los resultados finales que aparecen en el apartado de “Implementación del sistema”.

### Ejemplo de gráfico de torta mostrando el porcentaje de error de cada campo respecto al total de errores en toda la “[Tabla 1]”



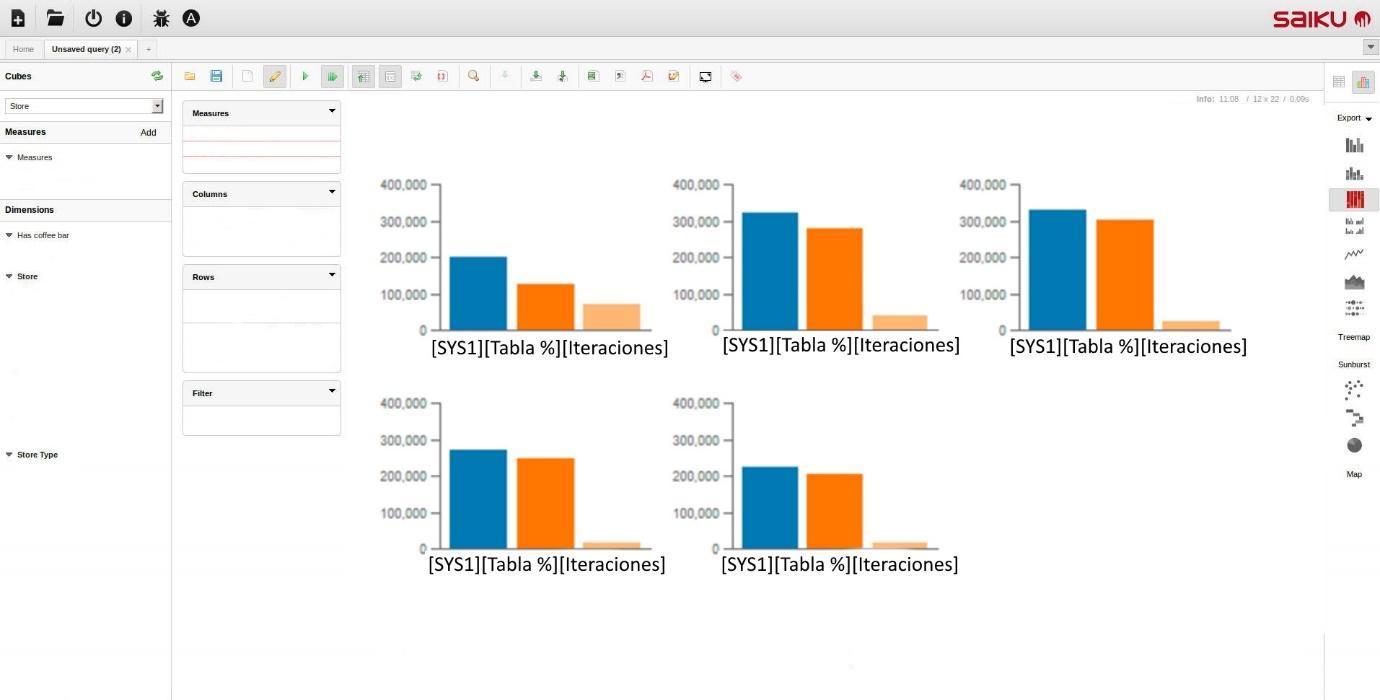
*Figura 4: Ejemplo Front - end con diagrama de torta.*

### Ejemplo de gráfico de barras mostrando el total de errores encontrados por sistema en decremento con el aumento de iteraciones



*Figura 5: Ejemplo Front - end con gráfico de barras*

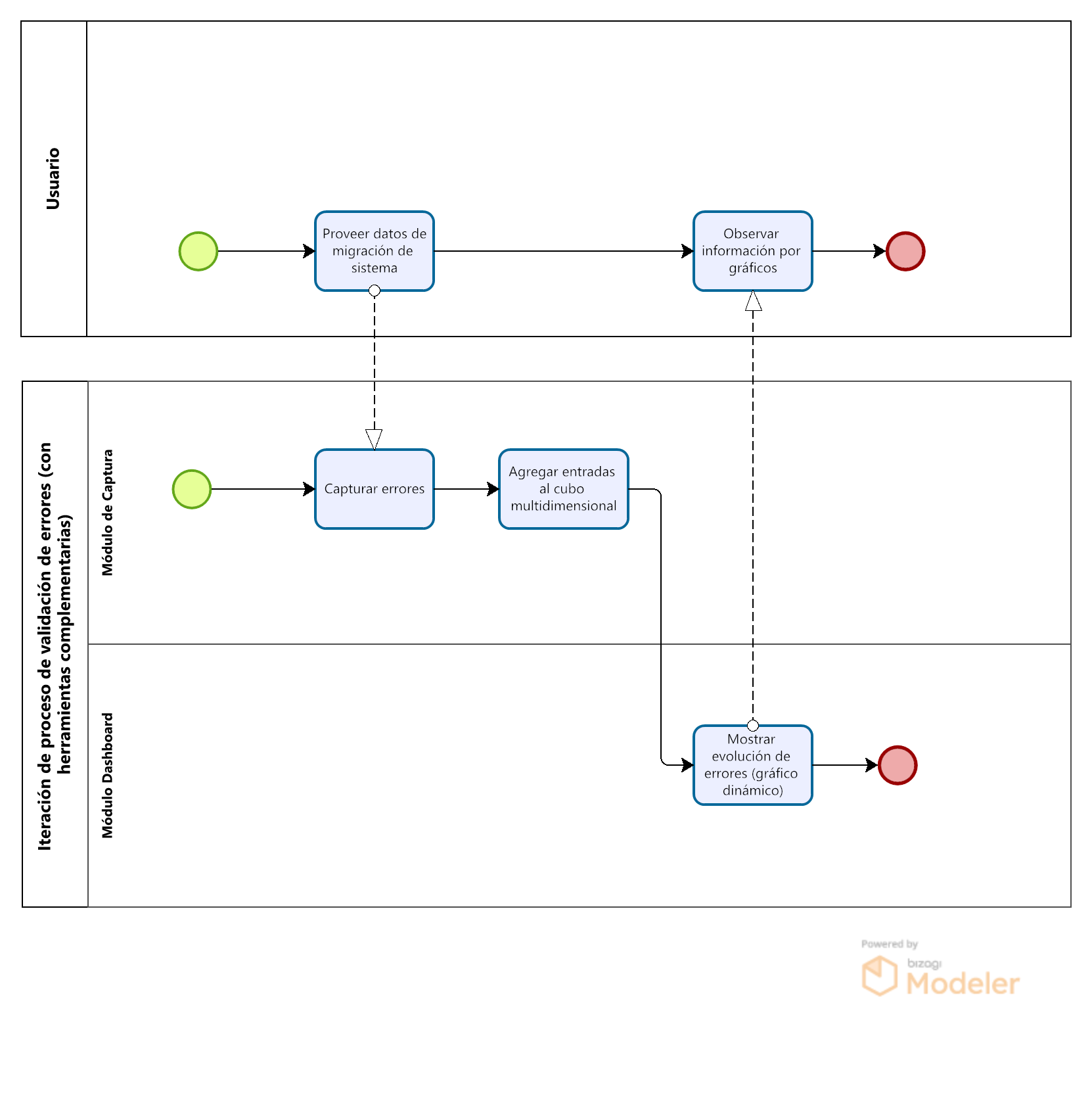
### Ejemplo de gráficos de barras donde la cantidad de errores disminuye con las iteraciones por tabla dentro de un “[Sistema 1]”



*Figura 6: Ejemplo Front - end con gráficos de barras por categoría*

## **Business Process Management (BPM)**

El diagrama describe el flujo donde el Módulo de Captura de la herramienta recolecta los errores de las tablas de migración una vez que el usuario los provee, añade las entradas respectivas al cubo multidimensional de datamart y luego muestra la evolución respectiva de los errores de migración a través del gráfico dinámico con el que interactúa el usuario y obtiene la información que buscaba.

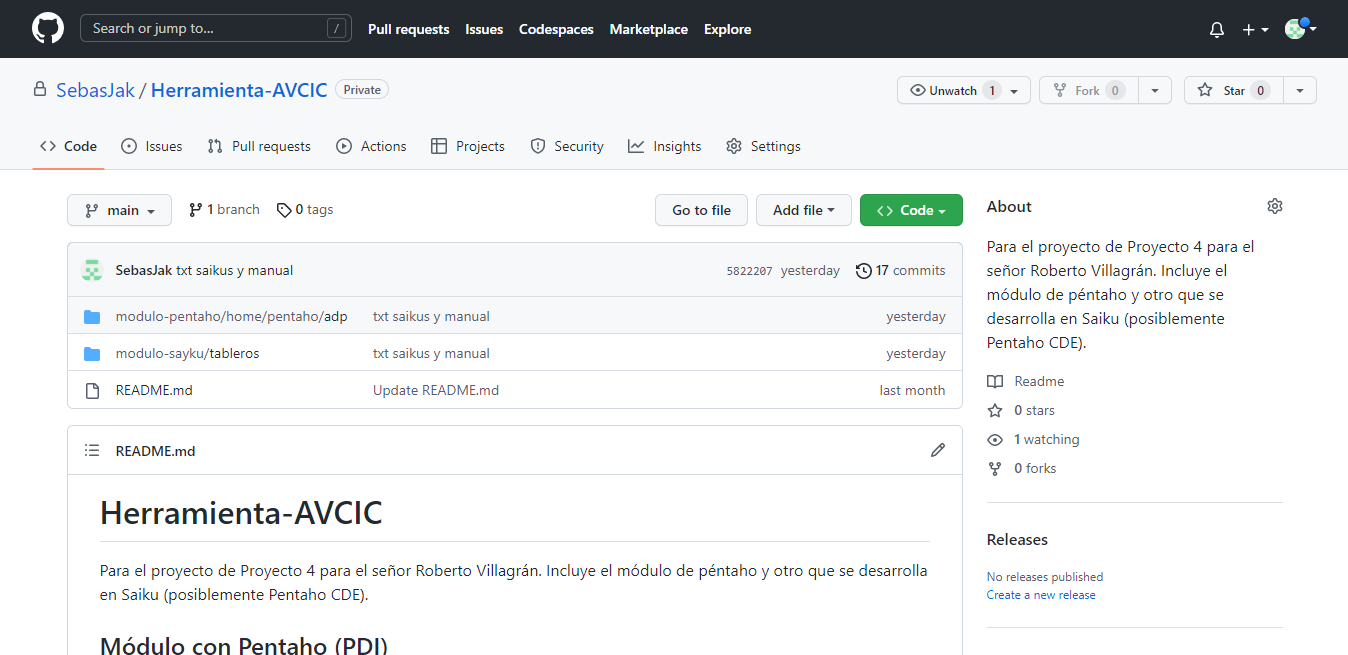


*Figura 7: BPM de la aplicación.*

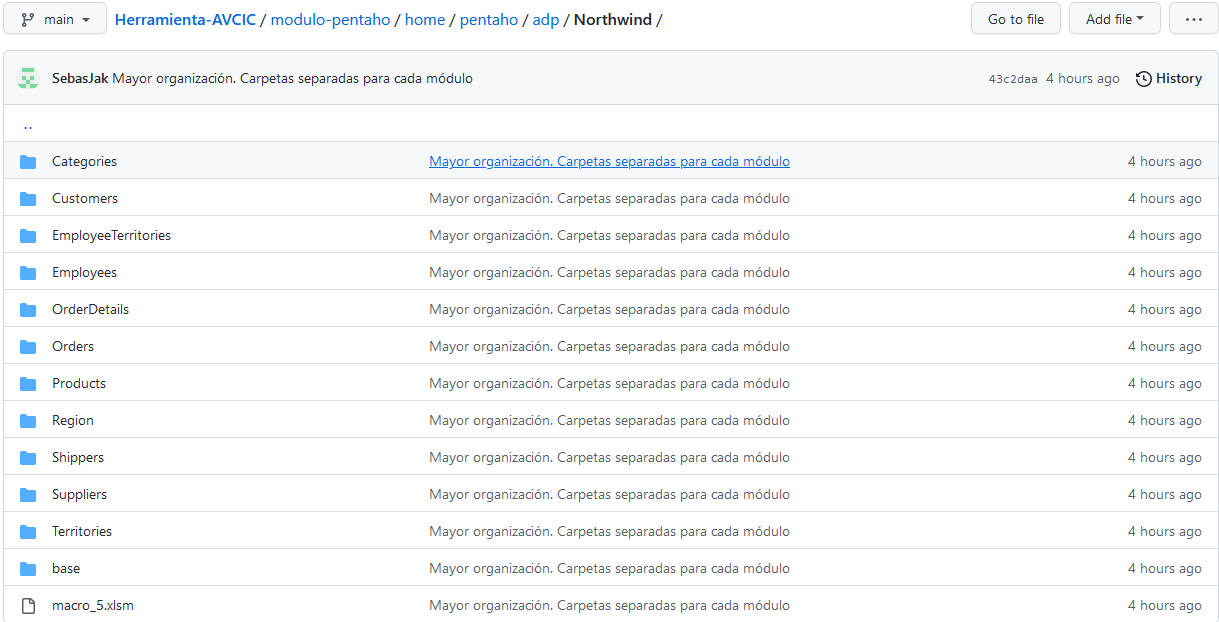
## **Repositorio de Github**

El avance de este proyecto se puede encontrar en el repositorio remoto de Github con la siguiente URL: <https://github.com/SebasJak/Herramienta-AVCIC>

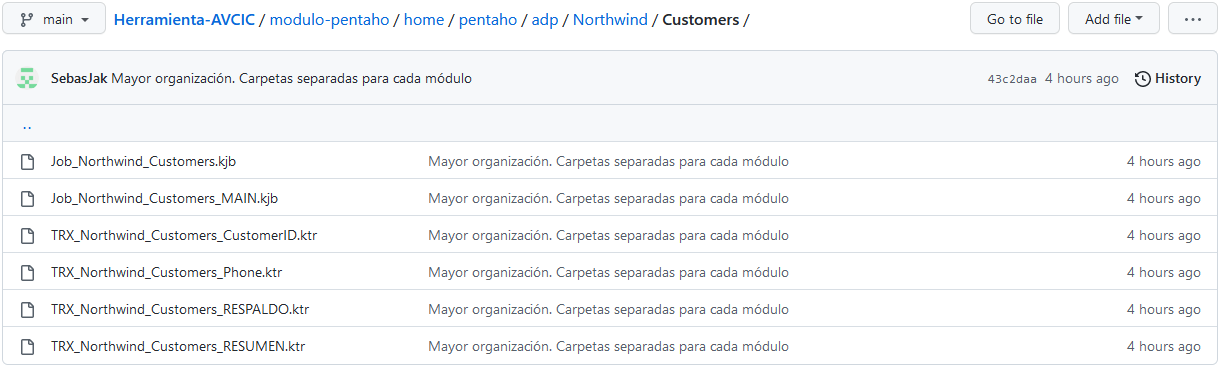
Las capturas de pantalla muestran la organización de carpetas que contiene el repositorio más su descripción.



*Figura 8: Página principal del repositorio en Github.*



*Figura 9: Ruta en el interior de la carpeta modulo-pentaho.*



*Figura 10: Sistema de archivos dentro de una ruta del módulo de Pentaho.*

## **Entregables anexos**

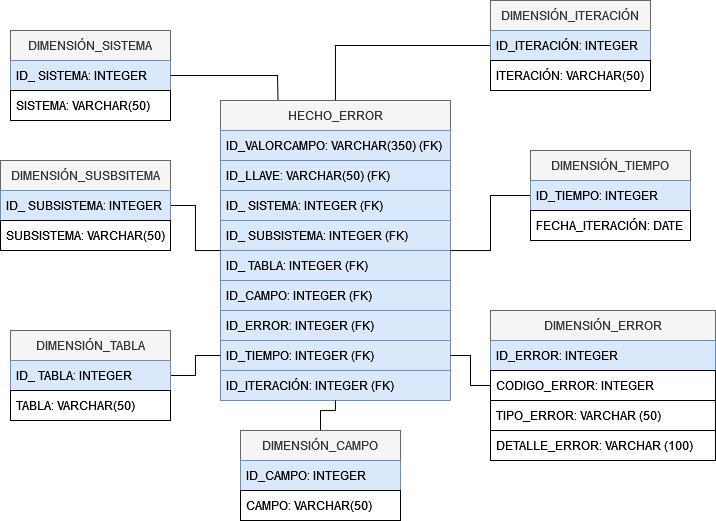
En el repositorio también se incluyen los documentos de orientación de uso:

* Manual Preparar conexión y cubo en Saiku
* Manual Usar los tableros en Saiku

Que vienen a ser los manuales de uso del producto.

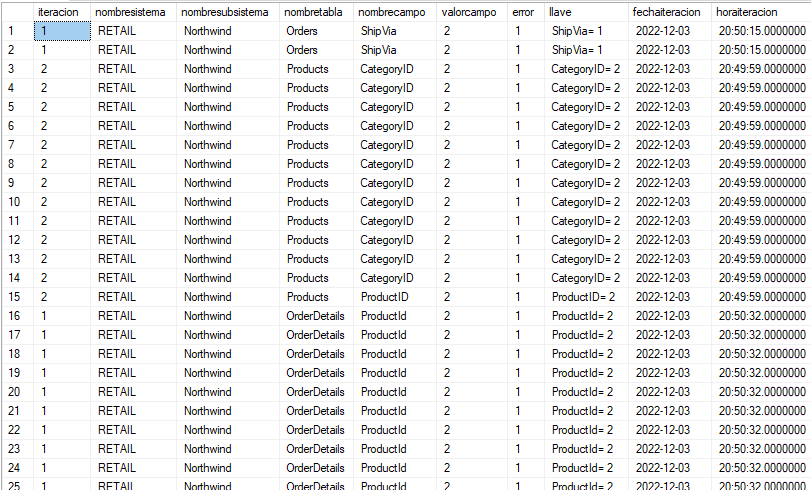
## **Almacenamiento y modelamiento de los datos**

Como lo que se está desarrollando es este proyecto en cuanto a los datos que se van a utilizar para el dashboard y se irán actualizando es un datamart, entonces el almacén para estos datos es naturalmente un cubo multidimensional, que a continuación se describe utilizando un modelo estrella.



*Figura 11: Modelo estrella del cubo multidimensional.*

Aunque cabe agregar que los datos de este cubo son extraídos y cargados principalmente de dos tablas de la base de datos que utiliza Digital Social Change SPA para almacenar datos de migración sin el uso de un modelo estrella: “his\_error” y “det\_error”



*Figura 12: Datos en “his\_error”.*

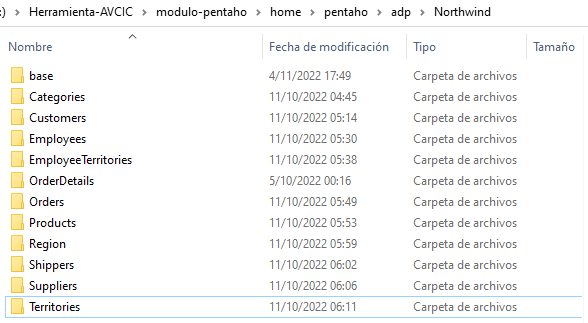


*Figura 13: Datos en “desc\_error”.*

## **Implementación del sistema**

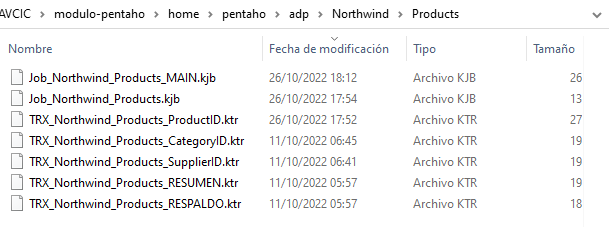
### Módulo con Pentaho

Para el caso del módulo con Pentaho se tiene la siguiente estructura para los archivos (base de datos utilizada de ejemplo: Northwind):



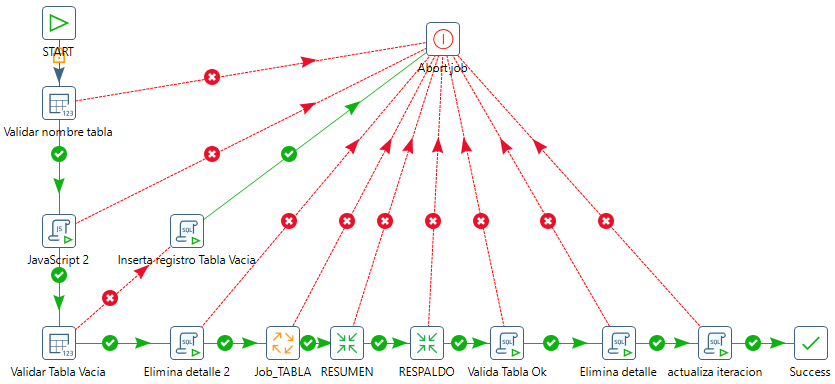
*Figura 14: Sistema de carpetas para la validación de las tablas de Northwind.*

El interior de cada carpeta que hace referencia a una tabla del sistema se ve de la siguiente manera:



*Figura 15: Archivos para la validación de la tabla Products*

El siguiente archivo de Pentaho Spoon de sufijo main es el que, por tabla, controla y pone en secuencia las transformaciones y jobs que utiliza Pentaho para la validación de campos respectiva de cada tabla:



*Figura 16: Archivo Job\_Northwind\_Products\_MAIN.kjb abierto con Pentaho.*

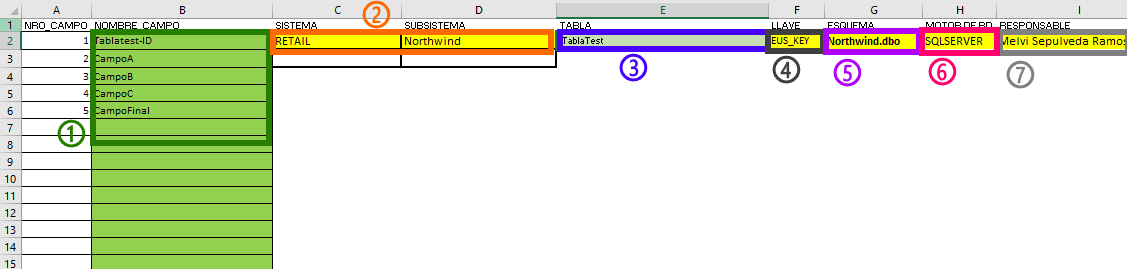
La explicación de cada paso dentro de la secuencia se explica en la siguiente tabla. No se considera el step de abortar trabajo en la secuencia por fines prácticos dado que es la interrupción del flujo esperado por el desarrollador:

|  |  |  |
| --- | --- | --- |
| **N° del paso** | **Nombre** | **Descripción** |
| 1 | START | Inicio de la secuencia de trabajo. |
| 2 | Validar nombre tabla | Antes de hacer operaciones en la base de datos con el nombre de la tabla se verifica que este esté registrado en la tabla ‘tablasmigracion’. |
| 3 | JavaScript 2 | Seteo de la fecha y hora de la iteración con base a funciones integradas en Pentaho. |
| 4 | Validar Tabla Vacía | Verifica que de hecho hayan registros en la tabla que se desea validar. Luego, si se detecta que la tabla está vacía se inserta un registro en la tabla ‘estadistica’ con la marca de tiempo, iteración, nombre de sistema, subsistema y tabla. |
| 5 | Elimina detalle 2 | Elimina de la tabla ‘det\_error’ todos los registros que puedan haber con el nombre de esa tabla, sistema y subsistema a la vez. |
| 6 | Job\_TABLA | Se invoca al job que hace las validaciones personalizadas para cada campo de la tabla a validar. |
| 7 | RESUMEN | Hace un resumen de los errores registrados en ‘det\_error’ y los inserta en la tabla ‘estadistica’. |
| 8 | RESPALDO | Hace una copia de los errores registrados en ‘det\_error’ en esa instancia y la guarda en ‘his\_error’. |
| 9 | Valida Tabla Ok | Si la tabla actualmente ya no presenta ningún error, se la registra como ‘OK’ en la tabla ‘estadistica’. |
| 10 | Elimina detalle | Limpia los cambios hechos en la tabla ‘det\_error’ para esa tabla, sistema y subsistema que se está validando. |
| 11 | actualiza iteración | Inserta en la tabla iteración la iteración actual en la que se trabaja con la tabla, el sistema y el subsistema. |
| 12 | Success | Fin planificado de la secuencia de trabajo. |

Steps donde la secuencia planificada se rompe:

|  |  |  |
| --- | --- | --- |
| **N° del paso** | **Nombre** | **Descripción** |
| 1 | Abort Job | Fin abrupto de la secuencia por algún fallo en los pasos. Indica el paso donde se aborta el trabajo ayudando en la depuración. |
| 2 | Inserta registro Tabla Vacia | La tabla vacía detectada queda registrada en la tabla ‘estadistica’ como ‘SR’ (sin registros) y ya no se prosigue a hacer la validación de integridad de sus campos. |

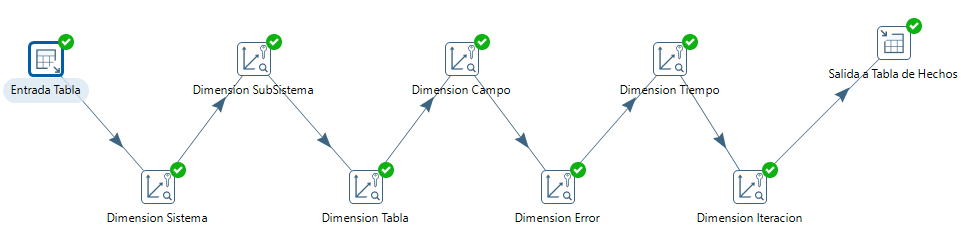
Como el módulo requiere de un conjunto de archivos que comparten la misma estructura pero aplicados específicamente a cada tabla del sistema en migración, se implementó un macro de Microsoft Excel (tomando un archivo base y complementándolo). El macro genera automáticamente los archivos de Pentaho que necesita cada tabla para el sistema que se configuró previamente tras cierta configuración inicial usando un archivo de Excel. Las funciones de los grupos de celdas en el archivo Excel son las que se explican en la tabla después de la figura siguiente:



*Figura 17: Archivo de configuración de Excel para ejecutar el macro, con grupos funcionales de celdas marcados*

|  |  |  |
| --- | --- | --- |
| **Grupo de celdas** | **Descripción** | **Valores que se esperan** |
| 1 | Nombres de los campos que se van a validar de la tabla. | Strings alfanuméricos sin espacios |
| 2 | Nombres del sistema y subsistema en migración. El nombre que se utiliza hasta el momento para la generación automática del nombre de los archivos es sólo el del subsistema. | String alfanumérico sin espacios |
| 3 | Nombre de la tabla a la que generar los archivos para su validación. | String alfanumérico sin espacios |
| 4 | Este dato no se usa. Era parte del archivo base. | - |
| 5 | Nombre del esquema de la base de datos que quedó como resultado de la migración. | String alfanumérico sin espacios. Acepta también punto. |
| 6 | Nombre del motor de base de datos en el que se hacen las operaciones en las tablas de la base de datos. Hasta el momento el macro está implementado sólo para SQLSERVER. | ‘DB2’, ‘SQLSERVER’, ‘SYBASE’ |
| 7 | Este dato no se usa. Era parte del archivo base. | - |

De la misma forma, el siguiente archivo TRX de Pentaho Spoon es el que toma los registros de la tabla “his\_error” y los dispone a lo largo de las tablas de dimensiones del cubo multidimensional y en su tabla de hechos.



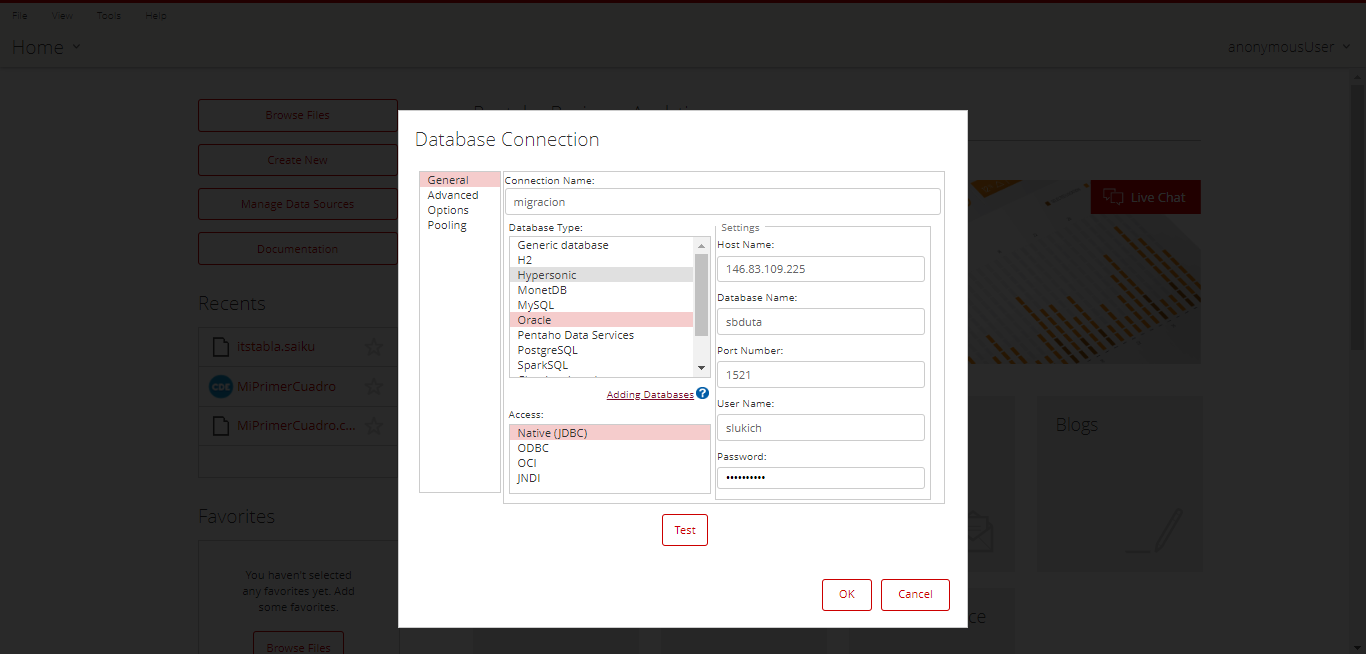
*Figura 18: Llenado de tablas de dimensiones*

Y sus Steps:

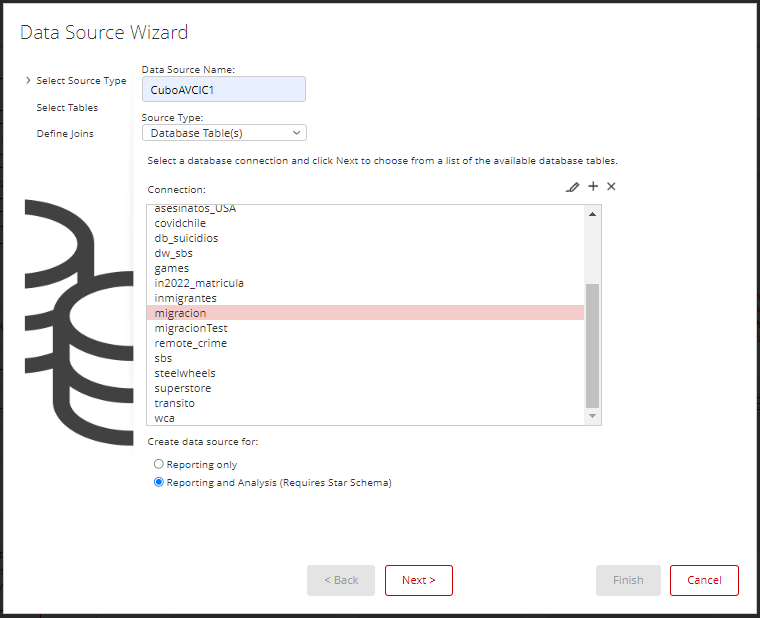
|  |  |  |
| --- | --- | --- |
| **N° del paso** | **Nombre** | **Descripción** |
| 1 | Entrada tabla | Selecciona la tabla his\_error más desc\_error con un JOIN. |
| 2 | Dimensión sistema | Llena la tabla dim\_sistema del cubo multidimensional. |
| 3 | Dimensión subsistema | Llena la tabla dim\_subsistema del cubo multidimensional. |
| 4 | Dimensión tabla | Llena la tabla dim\_tabla del cubo multidimensional. |
| 5 | Dimensión campo | Llena la tabla dim\_campo del cubo multidimensional. |
| 6 | Dimensión error | Llena la tabla dim\_error del cubo multidimensional. |
| 7 | Dimensión tiempo | Llena la tabla dim\_tiempo del cubo multidimensional. |
| 8 | Dimensión iteración | Llena la tabla dim\_iteracion del cubo multidimensional. |
| 9 | Salida a tabla de hechos | Llena la tabla fact\_errores del cubo multidimensional. |

### Módulo con Saiku Analytics

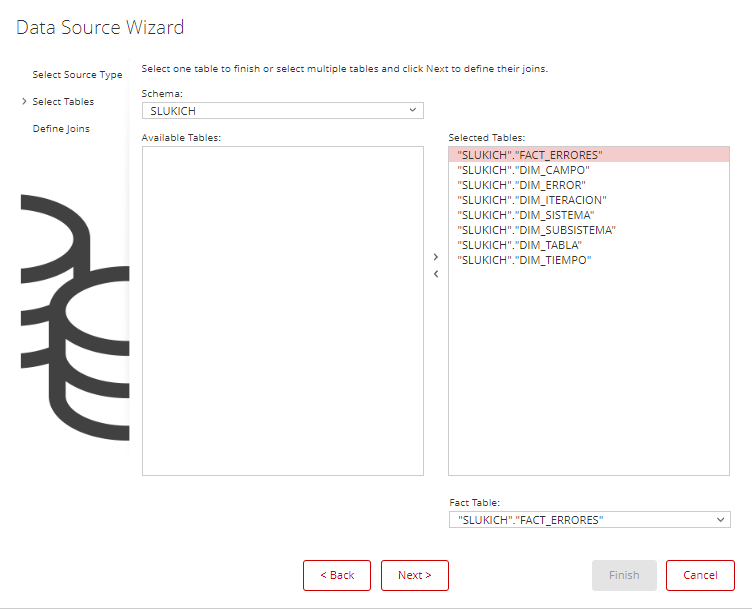
En el caso del módulo con Saiku Analytics, la versión final requirió de hacer antes de los tableros una conexión a la base de datos y una fuente de datos que contuviera el cubo multidimensional. En este ejemplo la base de datos con la que se hace la conexión es una proporcionada por la Universidad de Tarapacá por el departamento de ingeniería civil en computación e informática.



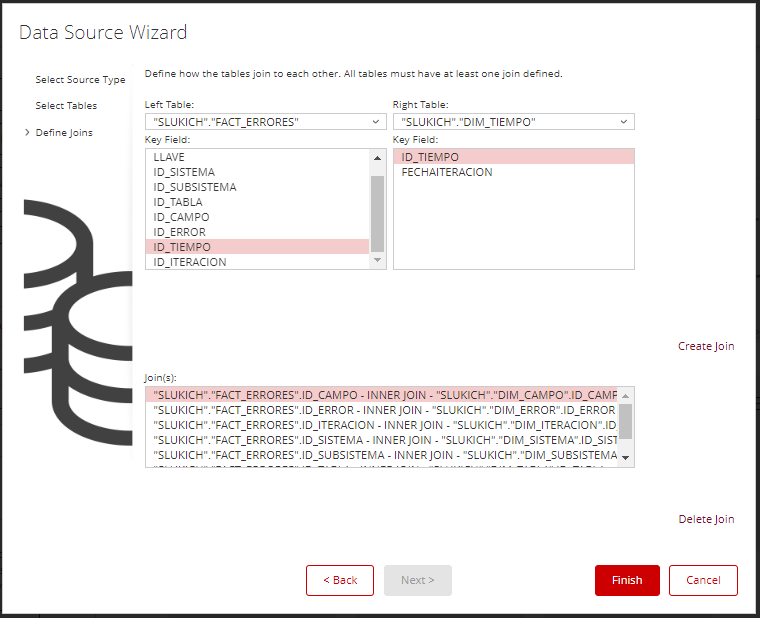
*Figura 19: Creando la conexión a la BD con la tabla de hechos.*



*Figura 20: Asistente de creación de fuente de datos - Paso 1.*

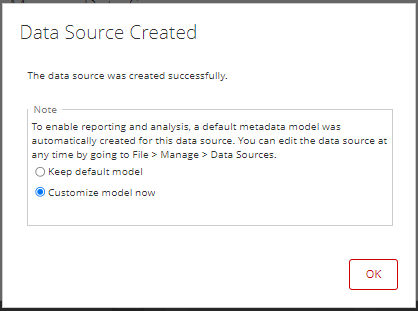


*Figura 21: Asistente de creación de fuente de datos - Paso 2.*

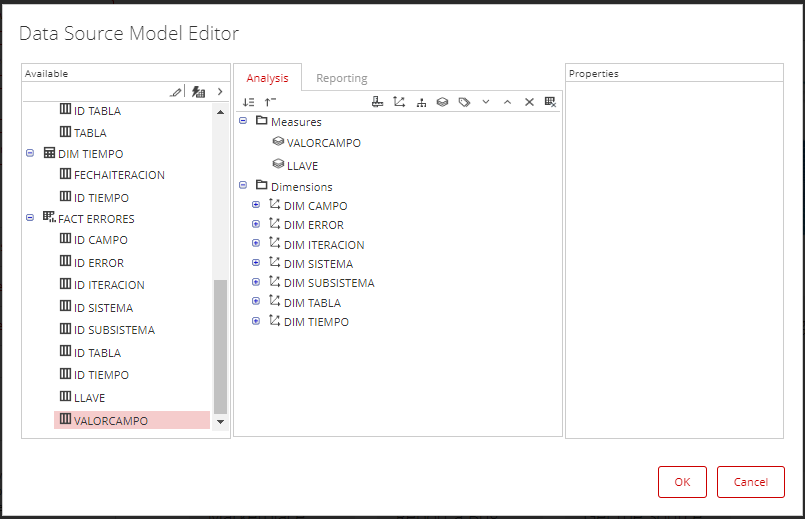


*Figura 22: Asistente de creación de fuente de datos - Paso 3.*

Como paso intermedio entre la creación de la fuente de datos y la creación del tablero queda hacer una configuración extra en el modelo de metadatos de la fuente de datos. Todos los elementos en la carpeta “Measures” deben ser eliminados y reemplazados por “valorcampo” y “llave”.

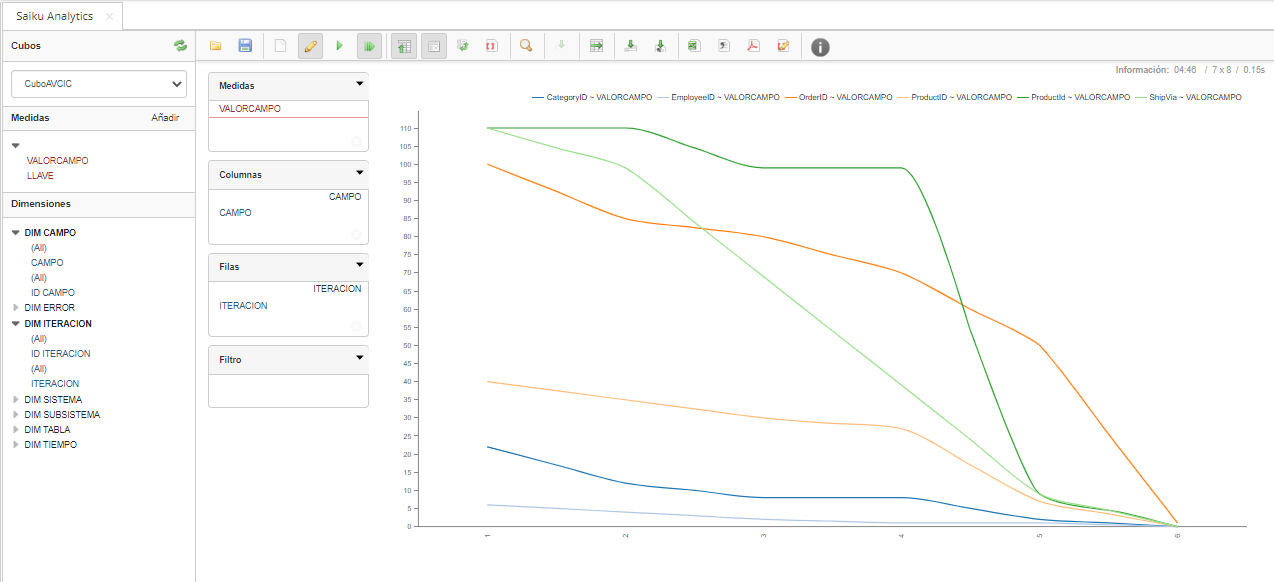


*Figura 23: Modelo de la fuente de datos - Parte 1.*

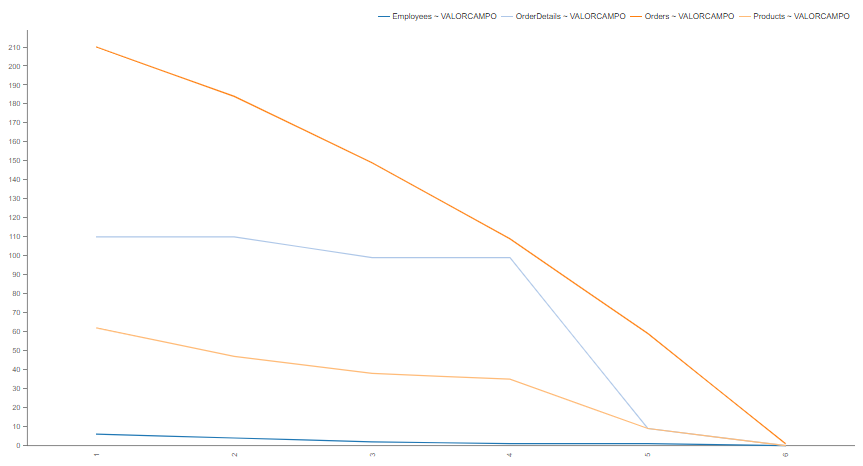


*Figura 24: Modelo de la fuente de datos - Parte 2.*

Así se tiene como resultado una fuente de datos disponible para seleccionar y mediante drag-and-drop en la herramienta Saiku Analytics se generaron los siguientes diagramas del dashboard.



*Figura 25: Diagrama de errores por campo a lo largo de las iteraciones.*



*Figura 26: Diagrama de errores por tabla a lo largo de las iteraciones.*



*Figura 27: Diagrama de errores por subsistema a lo largo de las iteraciones.*



*Figura 28: Diagrama de errores por sistema a lo largo de las iteraciones.*

**

*Figura 29: Parte del gráfico interactiva: Al pasar el cursor sobre un punto del gráfico.*

## **Análisis de resultados**

Finalmente se logró pasar por el ciclo que se tenía planteado, es decir, la carga de los datos de error por un proceso automatizado en Pentaho Spoon, la creación de un modelo estrella, su llenado, y representación a través de gráficos en un dashboard. Saiku Analytics tiene un sistema de vistas bastante completo y permitió visualizar la información con claridad en un diagrama de líneas hasta el punto en que se entendía que era una cantidad que disminuía en el tiempo sin agregar contexto a los gráficos del dashboard de que se trataba de la cantidad de errores. Los letreros color negro que incluyen los diagramas Saiku también son elementos de mucha utilidad y hasta funcionalidad necesaria, ya que, al pasar el cursor por encima contextualiza al usuario del dashboard de información sobre la iteración el campo/tabla/subsistema/sistema en cuestión además de brindar el número preciso de errores.

En el módulo de Pentaho Spoon en general se vio un buen desempeño. Esto por el hecho de que se simplificó la tarea de hacer las iteraciones además del trabajo de crear los archivos Pentaho Spoon uno por uno. La herramienta de inteligencia de negocios terminó demostrando tener mucho control y especialización sobre su dominio por la cantidad de Steps diferentes pero correspondientes entre sí que se usaron durante todo el proyecto.

## **Sugerencias de mejoramiento**

Una vía para mejorar el proyecto podría ser pasarse a una plataforma de pago (como por ejemplo: SQL server integration services) para rehacer los procesos de extracción, transformación y carga que se vieron a lo largo del informe. Esto por la sencilla razón que daría la ventaja para paquetizar el programa final de la herramienta de automatización de consistencia de integridad de campos.

## **Conclusiones**

Se concluye al final de este proyecto que:

* Se consiguió una mejora satisfactoria en el proceso de validación de la consistencia e integridad de campos. Esto debido a que no se contaba antes con un tablero de estadísticas, pero este se alcanzó a desarrollar con la herramienta Saiku. Se llegó de esta forma a mostrar la relación “errores restantes / iteración” para cada entidad del grupo de interés “campo, tabla, subsistema, sistema”, dando así el alcance estadístico de la evolución de errores para cada miembro del grupo.
* Se consiguió un progreso satisfactorio en el proceso de validación de la consistencia e integridad de campos debido a la introducción de un proceso semi- automatizado para la captura de errores. Si bien el macro de Microsoft Excel llegó a generar todos los archivos KTR y KJB para cada tabla de sistema salvo uno, el proceso de creación de estos archivos para la validación alivianó considerablemente la carga de tenerlos que hacer uno a uno por la GUI de Spoon. No obstante, lo archivos de Spoon generados semi- automáticamente para cada tabla terminaron cumpliendo su labor.

* A través del uso de herramientas de BI disponibles desde hacía ya un tiempo en el mercado se ha creado una innovación tecnológica al hacer frente al problema de la automatización de la validación de campos sin el uso de otras alternativas, como las librerías web de visualización, técnicas de minería de datos, etc. Que pudieron hacer frente también pero sin las ventajas que tiene un proceso ETL (extracción, transformación y carga) de estar diseñado para tratar eficazmente con el tipo de fuentes de datos que se describen en el problema (con bastantes campos).

## **Bibliografía**

1. Apuntes del profesor Diego Aracena. Intranet UTA, Chile
2. Documentación de Pentaho. <https://help.hitachivantara.com/Documentation>
3. Documentación de Microsoft SQL Server. <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
4. Digital Social Change SPA. <https://www.digitalsch.cl/>