**Uniendo y Combinando Datos & Gráficos de Doble Eje**

## **Profundización a Joins: Múltiples Campos**

### **Uniendo tablas con Múltiples Campos**

Al grano. Realmente Tableau guarda los mismos principios que SQL a la hora de hacer *Joins,* por lo que se recomienda leer sobre *Joins* desde la documentación que ya hay de SQL. Ahora, una novedad particular. Si usted bien recuerda, en SQL sólo bastaba con que las tablas estuviesen lo suficientemente *normalizadas* como para garantizar que sólo fuera necesario el uso de un solo *campo,* de tipo *llave* y por cada tabla,para poder unir dos tablas relacionadas entre sí. Es decir, con solo un campo *llave* que estuviese en ambas tablas relacionadas, que no fuera *null* ni se repitiera (que fuera *unique* por cada registro), era posible hacer una unión correcta entre dos o más de dos tablas (y de hecho esto es lo ideal, no se conoce otra forma en SQL para hacer *Joins*).

Sin embargo, en Tableau, las tablas no siempre están lo suficientemente *normalizadas* y se corre el riesgo de que ninguna tabla tenga un campo con un identificador único por cada registro (al menos no lo tenga el campo *primario* con el que nos interese, propiamente, unir las tablas en sí), es decir, se corre el riesgo de que se repitan valores en los registros de los campos o columnas *llaves* de las tablas de interés (las que nos interesa unir); por lo que a veces es necesario no sólo tomar un campo como parámetro (como *llave* única) para hacer una unión correcta sin generar registros erróneos; sino, tomar **múltiples campos** para garantizar que los registros de cada una de las tablas unidas **sí** sean correspondidos entre sí, *los datos sí estén limpios*. **Observe el siguiente ejemplo para mayor comprensión. Se evidenciará porqué, para este caso, será necesario unir las tablas con múltiples campos.**

#### **Ejemplo**

Trataremos de unir dos tablas, A & B, considerando solamente al campo ***Order #*** como campo *llave* para lograr la unión.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Con las especificaciones previamente dadas, nuestra unión quedaría así:

Interfaz de usuario gráfica

Descripción generada automáticamente  
**Esto es incorrecto.** El cruce de datos está mal. Si se percata, **no coinciden los datos de cada uno de los registros al ser unidos, puntualmente son errados los registros 3, 5 & 6 (no hacen falta)**. Planteado lo anterior, lo que corresponde sería considerar un campo adicional, como parámetro, para lograr una unión correcta y *limpia* entre ambas tablas. El parámetro *llave* adicional será ***Store.*** Veamos. **Observe la nueva sentencia de la unión (que ya incluye al campo *Store*).**

**Captura de pantalla de un celular

Descripción generada automáticamente  
 *Resultado***

**Tabla

Descripción generada automáticamente**

## **Joins de datos VS Blend de datos**

Trataremos de explicar las diferencias que hay entre un *Joins de datos* vs un *Blend de datos* y, adicionalmente, en qué escenario uno podría ser más apropiado que otro. Sin embargo, dejemos clara la primera diferencia: **Los *Joins* ocurren en la capa física y los *Blends* ocurren en la capa lógica.**

Esta vez continuaremos trabajando con el *dataset* ‘P1-AmazingMartEU2’; en esta ocasión nos interesa saber, puntualmente, **cuál fue el rendimiento de cada uno de los departamentos para la misma tienda de Europa que estuvimos analizando en la sección anterior:** *‘Mapas, Gráficos de dispersión & Dashboard’*. El rendimiento, **según *departamento****,* pretende ser evaluado tomando en consideración el desempeño **mensual** para todos los años de estudio propios al *dataset*; es decir, **se pretende evaluar el rendimiento (beneficio) de cada departamento** **mes a mes y compararlo con el objetivo de ventas que tienen trazado para ese mismo marco temporal, es decir, para cada mes**.

Adelanto. Será necesaria una ***unión*** entre tablas debido a que los datos de cada una de las ventas por departamento lo posee una tabla, mientras que los datos relacionados a las fechas de estas ventas fueron registrados en otra. Dicho lo anterior, se requiere unir una tabla con otra para poder tener datos completos que responda a nuestra consulta: **evaluar el rendimiento de cada departamento mes a mes y compararlo con el objetivo de ventas que tienen trazado para ese mismo marco temporal, es decir, para cada mes**.

Concretamente, la tabla que posee la información de cada uno de los registros de venta, por cada artículo, para cada departamento es la tabla **‘OrderBreakDown’** y la tabla que posee la información de cada una de las fechas de las ordenes concretadas, que pueden o no poseer muchos artículos dentro de cada orden, es la tabla: **‘ListOfOrders’**. Por lo pronto, esta será nuestra unión. Veamos.



### **Joins con Registros Duplicados**

Esta unión resultante es de tipo: ***Joins con Registros Duplicados***. Explico. Si bien la tabla *ListOfOrders* no depende de campos múltiples para hacer una unión a partir de ella, sino que, cuenta con su propio campo, tipo *llave primaria,* que es **Order ID** donde no existe registro repetido alguno; si bien esto es así, la tabla *OrderBreakDown*, al contrario, como registra las ventas de cada uno de los artículos vendidos, uno por uno sin importar si pertenecen o no a la misma orden, sí es posible que encuentre que varios **Order ID** se repitan en sus registros (porque hay ordenes con más de un artículo y los registros se hacen en términos de *artículos,* no *ordenes*). Entonces, como el campo **Order ID**es quien puede unir ambas tablas (es el campo que tienen en común, el que los relaciona) para que haya una equivalencia entre el número de los registros de cada tabla y queden a la par, al ser unidas, se deben incrementar los registros de la tabla *ListOfOrders* para cada uno de los **Order ID** que se repitan en la tabla *OrderBreakDown*; así hasta que haya una equivalencia en el número de registros de ambas tablas y los registros, de cada tabla, se correspondan y sean consistentes entre sí. Esto es, efectivamente, un ejemplo perfecto del tipo de ***Join con Registros Duplicados***; pues, a la tabla *ListOfOrders* le toca duplicar sus registros al punto de lograr ser equivalente a los registros de la tabla *OrderBreakDown* al ser unidas.

### **Join de tablas con Múltiples Campos**

Ahora, necesitaremos unir nuestra tercera tabla ***SalesTargets*** a la unión de la tabla resultante entre ***ListOfOrders*** + ***OrderBreakDown***, pues es en *SalesTargets* donde se encuentra nuestro *Target por mes* para cada una de nuestras *Categorías*; en consecuencia, **necesitaremos esta última tabla para poder comparar los rendimientos de cada departamento, mes a mes, con el *target* que ellos se fijaron para cada uno de esos meses***.* Luego, es muy probable que esta tercera tabla necesite de *múltiples campos* para lograr una *unión* con las otras; es decir, podría ser esto un ejemplo perfecto de ***Unir tablas con Múltiples Campos.*** Lo explico.

Diferente a la tabla *ListOfOrders,* la tabla *SalesTargets* no cuenta con un campo en específico que haga de *llave primaria*, **esto es así porque ninguno de sus campos podría presumir de que NO posea ningún registro repetido**: en todos sus campos, si los evaluamos uno por uno, verá que hay registros repetidos.

He aquí donde nace la necesidad de valerse de **múltiples campos** de una tabla para hacer una unión con otra; esto para lograr una combinación de valores/datos provenientes de diferentes campos, por cada registro de la misma tabla, donde sí resulte ser único, o haga inigualables, cada registro de la tabla en cuestión. En este caso puntual, para cada registro por separado, se podrían unir los datos del campo ***Month of Order Date*** con los datos del campo ***Category***, convirtiendo así a cada registro de la tabla ***SalesTargets*** en un registro único que conserva, en una eventual unión (*Join*) con otra tabla, una unión consistente y limpia.

Explicado todo esto, si procedemos con la unión de la tercera tabla, sería así:



Sin embargo para este caso puntual, este tipo de unión, el *Join* convencional, haría que las demás tablas se rijan a la granularidad de la última, no por ser la última tabla unida; si no, por ser la tabla con la granularidad más alta de todas las tres; es decir, los datos de la unión de las tablas *ListOfOrders* & *OrderBreakDown* ahora se van a estructurar en términos de los datos de la tabla *SalesTargets,* más puntualmente en términos de los campos ***Month of Order Date & Category***… Esto no es conveniente, se suprimiría la información relacionada a los números de órdenes y, más allá de eso, la información de cada uno de los artículos.

De hecho, la unión se logra mediante el campo ***Category*** (ni siquiera considera a ***Month of Order Date***).



Vamos al grano. Para evitar el anterior conflicto, lo siguiente: toda vez que pretenda unir de manera correcta una tabla con otra, por medio de **múltiples campos**, haga mejor un ***Blending*** en vez de un ***Join*** propiamente***.***

### **Blend de datos**

Por lo pronto tenemos que nuestra unión, desde la actual fuente de datos, quedaría así:

Diagrama

Descripción generada automáticamente

El *Blend* o fundido de datos, para este caso puntual, consiste en ‘unir’ nuestro fichero ***SalesTargets*** a la unión ***ListOfOrders + OrderBreakdown*** sin realmente hacer un ***Join.*** No sería un ***Join*** porque la tabla ***SalesTargets*** necesita unirse **valiéndose de múltiples campos;** con lo cual, así, no sería recomendado hacer un ***Join*** para lograr la *unión* entre las tres tablas en cuestión.

**Cuando una tabla se vale de múltiplos campos para hacer una unión con otra u otras, lo ideal es hacer un Blend de datos.**

Entonces, esta tabla que se vale de múltiplos campos para unirse con otra, necesitamos abrirla desde una fuente de datos completamente nueva desde el mismo Tableau. Veamos.

Vamos a nuestra ***Worksheet*** y hacemos clic en el *cilindro* que hace alusión a las bases de datos ; dado el clic, se nos pide insertar una fuente de datos adicional completamente nueva. La fuente de datos que va a insertar, como se imagina, debe contener dentro de sí al fichero ***SalesTargets*** (para este caso puntual continuamos hablando de ‘P1-AmazingMartEU2’) para poder lograr una unión con las otras tablas por medio del *Blending.* Así quedaría nuestra segunda fuente de datos donde tan sólo se vería arrastrado el campo ***SalesTargets***:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El total de las diferentes fuentes de datos insertadas se verían así: Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En **conclusión**, y antes de abordar a profundidad el alcance de un recurso tan poderoso como lo es el *Blending* y su forma de trabajo*,* conciba a esta herramienta como un ***Join*** inteligente que puede ocurrir sobre la marcha, no sobre la capa física, sino en base a la lógica (al código); de lo cual, tomaremos ventaja: vamos a crear visualizaciones requeridas usando el *Blending* en lugar del *Joining* para no machacar la estructura física que tienen los datos*.*

En resumidas cuentas no se puede proceder con un ***Join***,en la unión de estas tres tablas, sino a un ***Blend***; para este caso particular no se pudiera hacer un ***Join*** debido a que hubiéramos tenido que *agregar*, o llevar la *Granularidad*,nuestro conjunto de datos resultante ***ListOfOrders + OrderBreakdown*** a la granularidad de la tabla ***SalesTargets***, ya queno puede ser correspondida (son *granularidades* diferentes); en consecuencia, se perdería una gran cantidad de información de ***ListOfOrders + OrderBreakdown*** (éstas en primer lugar sí se correspondieron, la granularidad de ambas sí compaginaron porque contaban con la información necesaria para hacerlo por medio del campo ***Order ID***).

Especialmente le tocará, sí o sí, hacer un ***Blend*** cuando los conjuntos de datos que pretenda unir ni siquiera provengan de una misma fuente de datos; es decir, cuando sepa de entrada que son diferentes y que no necesariamente deban relacionarse. Si los conjuntos de datos son diferentes probablemente también tendrán diferentes granularidades; por lo cual no se recomendaría nunca hacer un ***Join***,si pretende unirlos, sino un ***Blend***.

## **Blend de Datos en Tableau**

En esta clase trataremos de hacer un ***Blend de Datos***, pero a partir de un nuevo *dataset* ‘**Airline**-**Comparison’**. El *dataset* ‘Airline-Comparison’ es de formato *.****xlsx***y se compone de dos hojas de cálculo (dos tablas para efectos prácticos): **‘Airline1’ & ‘Airline2’**; entonces, vamos a tratar a estas dos tablas como dos fuentes de datos diferentes para poder proceder con un ***Blend.*** Veamos:

***Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente***

Por sólo una finalidad académica, convengamos que de nuestra primera fuente de datos (***Airline1***) queremos ver una visualización que consista en un gráfico de barras que evalúe el ***Revenue*** por ***Region***, tal que así:

Gráfico, Gráfico de barras

Descripción generada automáticamente

Listo, esta es la visualización que tenemos para nuestra primera fuente de datos **‘*Airline1*’.** Vamos ahora a la fuente de datos ***‘Airline2’.***

Observe cómo varia un poco su interfaz, en la parte más izquierda de **Tableau**, al situarse sobre la base de datos ***‘Airline2’*:tenemos una barra de color naranja iluminándose al costado más izquierdo y una figura simbólica, tipo *cadena,* también de color naranja en el primer campo mostrado de su actual fuente de datos. Observe:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ambas son indicativos visuales que nos revelan que, ahora, usted está posibilitado para hacer un ***Blend de Datos***; sí, tal cual, un *Blend* o *fundido* de datos entre ambas fuentes de datos diferentes: **‘Airline1’** & **‘Airline2’.**

El icono de *cadena* o enlace, principalmente, es quien nos indica que se ha preparado la hoja de trabajo para hacer un *Blending* de datos por nosotros**; partiendo de, y sin dejar de tomar en cuenta, la visualización inicial de nuestra primera fuente de datos, es decir, usted seguirá trabajando sobre la *Worksheet* de la fuente de datos ‘Airline1’ y lo que se espera es que manipule los campos de ‘Airline2’ desde ahí.** Dicho lo anterior, efectivamente se hace una unión de datos sobre la *marcha*. Se habla de *marcha* debido a que, a diferencia de unir datos con ***Join,*** no necesitamos *unir* previamente los *datasets* para poder, luego, trabajar con ellos de manera conjunta.

Por ejemplo, ya estábamos evaluando inicialmente el *Revenue* por *Region* de **‘Airline1’***…* Lo que se espera es que pretenda evaluar, también, el *Revenue* de **‘Airline2’. Lo único que tendría que hacer, para evaluar visualmente el rendimiento de los *Revenue* de cada fuente de datos en el mismo gráfico, es arrastrar ahora el *Revenue* de ‘Airline2’ a la misma *Worksheet* de ‘Airline1’** (Arrastre el *Revenue* de ‘Airline2’ a la misma columna, o fila, donde se encuentre el *Revenue* de ‘Airline1’). **Así quedaría nuestra visualización ahora:**

**Gráfico, Gráfico de barras

Descripción generada automáticamente**

### **Manejo de la leyenda *Marcas* en un Blend**

Realizado nuestro primer ***Blend***, el manejo de algunas herramientas puede variar. Por ejemplo, se crea una leyenda de *Marcas* para cada una de las fuentes de datos unidas en el ***Blend,*** estopara poder personalizar la configuración de *marcas* para cada una de ellas por separado; sin embargo, se deja también una leyenda de ***Marcas*** tipo ‘madre’ en caso de que deseemos configurar, desde una misma leyenda*,* las *marcas* de todas las fuentes de datos unidas en el ***Blend.* Veamos esto:**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Las leyendas de *marcas* se ordenan en el mismo sentido en el que están enlistadas todas sus fuentes de datos, siendo la leyenda de *marcas* ‘Todo’ la leyenda de *Marcas* tipo ‘madre’; es decir, la que asigna distinciones visuales de forma global, sobre todas las fuentes de datos unidas en el *Blend*.**

Observe esta configuración de *marcas* a modo de *pedagogía*:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

No es lo más estético posible, pero es sólo una muestra de lo que podemos hacer teniendo una configuración de *marcas* global y una personalizada para cada fuente de datos del *Blend.* De hecho, partamos de la visualización que teníamos anteriormente:

Gráfico, Gráfico de barras

Descripción generada automáticamente

Ahora, vamos a tomar en consideración la variable del *tiempo* en nuestro gráfico; es decir, vamos a incorporar el campo *Period* de la fuente de datos ‘Airline1’ sobre nuestra *Worksheet*. Como nos interesa que los datos de las *Regiones* se estructuren en términos de los *Periodos,* y no lo contrario, vamos a situar de primero el campo *Period* en las filas y no situarlo después de *Region*; es decir, queremos guardarle un mayor grado de importancia a *Period* sobre *Region.*



*Queda nuestra visualización así:*

Gráfico, Escala de tiempo, Gráfico de barras

Descripción generada automáticamente

Ahora, para cada fuente de datos (2 en total), hay un gráfico de barras por periodo; en total son dos los periodos: 2015 & 2016. El marco temporal del campo *Period* es Anual. En general, tendríamos 4 gráficos de barras para la visualización de nuestro primer *Blend.*

### **Necesidad de Etiquetas**

Si observa detalladamente nuestra última visualización; se percataría que no se distingue visualmente, mediante sus barras, las cifras de los dos gráficos de barras para la fuente de datos ‘Airline2’; es decir, los dos gráficos de barras que se sitúan al costado más derecho de nuestra *Worksheet*: Da la sensación de que sus cifras fueran iguales para cada periodo de tiempo, es decir, 2015 & 2016, y NO es así.

Desafortunadamente, con una unión de datos por medio de ***Blend,*** se cuenta con el riesgo de que los gráficos de alguna de las fuentes de datos no sean precisos; por ello, si es preciso se usan *Marcas* de tipo *Etiquetas* para dejar claro, visualmente, las distinciones de nuestros datos*.* Vamos allá.

Gráfico de barras

Descripción generada automáticamente

Aquí fue necesario, por obvias razones, que no se aplicara una *Etiqueta* de representación global, sino que asignara sus *Etiquetas* de forma independiente para cada fuente de datos; de lo contrario, sus *Etiquetas* no serían consistentes con las cifras que tiene, de forma independiente, en cada fuente de datos.

**Ahora, si te fijas mejor, el problema persiste! Es decir, las cifras de los dos gráficos de barras para la fuente de datos ‘Airline2’ siguen iguales, ignorando el hecho de que son dos periodos de tiempo diferentes (*Period*) y que generaron, en consecuencia, *Revenue* diferentes.**

Por ejemplo, para la compañía área número 2 (*Airline2*) son 26M los ingresos generados en África si analizamos los datos de forma agregada, si no granulamos los *Revenue* por periodo; es decir, si no dividimos los ingresos entre los dos periodos de tiempo anuales (2015 & 2016). Observe.

Gráfico

Descripción generada automáticamente

Sin embargo, *Airline2* no se vería afectado si intento *granular* por *Period*; en consecuencia, como no le afecta, lo que sucedería es que su duplicaría exactamente el mismo gráfico para *Airline2* en el año 2016, nuestro 2do periodo de tiempo evaluado. Esto no es lo que queremos, esto sería un tipo de *Dirty* data: Duplicados. Observe de nuevo.

Gráfico de barras

Descripción generada automáticamente

En efecto, hay un duplicado en la visualización. Profundizaremos sobre esto.

### **Campos No correspondidos**

Retomando el caso anterior, lo que acontece es que, si bien el campo ***Period***(de la fuente de datos ***Airline1***)contiene el mismo tipo de dato que el campo *Year* (de la fuente de datos ***Airline2***)y hay total consistencia entre ambos datos; es decir, ambos contienen **fechas** en términos de **años**, **Tableau no interpreta que el campo *Period* de la fuente de datos *Airline1* se refiera, necesariamente, al campo *Year* de la fuente de datos *Airline2*; esto se debe, justamente, a que no hay una correspondencia en los nombres de los campos.**

**Solución: La conexión entre ambos campos tenemos que hacerla de forma manual** (Con el campo *Region* esto no fue necesario porque, como bien puede percatarse, ambas fuentes de datos conservan el mismo campo bajo el mismo nombre).

#### **Conexión manual entre campos**

**Opción 1.**

Váyase a la pestaña ***Datos*** que se sitúa sobre la barra de menú principal, haga clic sobre ella y, posteriormente, clic en ***Editar Relaciones de combinación…***

*Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente*

Ahí observará que campos realmente fueron unidos, de forma automática por Tableau, al realizar un *Blend* entre fuentes de datos. Por defecto, se establece una combinación automática sobre los múltiples campos que coincidan entre las fuentes de datos unidas, para este caso puntual hablamos de *Region*.

*Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente*

Para establecer una conexión manual, entre campos, debe hacer clic en *Personalizado* y luego en *Añadir.* Seleccione que campos, de cada fuente de datos, conectar. Como bien sabe, en nuestro caso queremos conectar a *Period* con *Year.* Observe.

*Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente*

Clic en *Aceptar.*

Observe de nuevo el recuadro de nuestras conexiones, se ha incluido ahora la conexión manual hecha.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Hacemos clic en *Aceptar*,si no nos interesa hacer más conexiones manuales, y listo!

Lo siguiente que verá es cómo ahora sí, definitivamente, se ha *granulado* correctamente el *Revenue* en términos de periodos de tiempo; es decir, en años, para la segunda aerolínea (*Airline2*): se han suprimido los ingresos duplicados para, ahora sí, dividir los ingresos totales entre los años 2015 & 2016. Observe.

Escala de tiempo

Descripción generada automáticamente

**Con esto extraemos una conclusión más**: Si requiere unir datos, de diferentes tablas, donde los campos que se corresponden entre sí no conservan exactamente el mismo nombre, debe recurrir al recurso de *Blend* en vez de *Join* para lograr esta unión de datos.

Nuestras conexiones también se reflejan en el interfaz, ahora hay dos campos y no solo uno con *Blend* (y ambos hacen parte de la visualización, debido a que el color del icono de *enlace* es naranja)*.*

Texto

Descripción generada automáticamente con confianza baja

Pudiera haber campos conectados con *Blend*, de diferentes fuentes de datos, y no estar visualizados en la *Worksheet*; para este caso puntual, el icono del *enlace*,para estos campos conectados NO visualizados, se vería así:



**Opción 2.**

Tan sencillo como hacer que el nombre de los campos correspondidos, al unir las tablas con *Blend,* sean iguales. **La solución número 2 consiste en cambiar los nombres de los campos correspondidos, si hace falta, para lograr una equivalencia entre ellos (se llamen igual) y automáticamente Tableau haga la conexión de los campos por ti; repito, al poseer el mismo nombre los campos correspondidos.**

### **Tablas: Primarias & Secundarias**

#### **Lógica detrás de la Asignación**

Observe que, entre *Airline1* & *Airline2,* la tabla o fuente de datos primaria es *Airline1* y la secundaria es *Airline2*.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

No es que sea así por defecto, no; sino que, **Tableau selecciona cuál será la tabla primaria a partir del primer campo que decida usted arrastrar a la *Worksheet*** (a partir de qué tabla o fuente de dato provenga este primer campo arrastrado); de hecho, el ***Blending*** como tal se lleva a cabo a nivel de hoja (a nivel de Worksheet); es decir, usted puede crear cuántos ***Blending*** desee desde 0 por hoja, en consecuencia, puede personalizar la configuración de un ***Blending*** como más le plazca a nivel de Worksheet.

#### **Distinción visual y Blend como un Left Join desde la capa lógica**

El *Checkmark* de color azul significa que está sobre su tabla primaria mientras que el *Checkmark* de color naranja significa que está sobre su tabla secundaria. En cuanto a grados de importancia, los campos de la tabla primaria desplazan a los campos de la tabla secundaria (en caso de que no haya correspondencia entre todos los campos al hacer el *Blend*). En resumen, se subordinan los campos de la tabla secundaria a la primaria. **Debe tener en cuenta que, sólo se mostrarán los campos y los valores de campos de su tabla primaria y la tabla secundaria tratará de ligarse a eso; sin embargo, más allá de eso, se ignorarán por defecto los campos y valores de campos extras que estén contenidos en su tabla secundaria, pero que no estén sobre su tabla primaria. Repito. Si hay campos o valores de campos que tiene su tabla secundaria, pero no su tabla primaria, estos campos y valores serán ignorados por defecto. Tenga cuidado.**

Por ejemplo, si considera que *Airline2* debiera ser su tabla primaria y *Airline1* su tabla secundaria; esto sería una mala decisión porque en el *Blend* sólo se vería la evaluación de los campos de *Airline2* y, adicionalmente, los que *Airline1* pudiera corresponder a partir de los campos de *Airline2*: *Airline1* toma como referencia los campos de *Airline2* para poder mostrar los suyos. Usted ya sabe, o ya debería saber, que *Airline2* no evalúa la misma cantidad de campos que *Airline1*;por lo que esos datos extras, que no aborda, se perderían a la hora de visualizar el *Blend*. Observe.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Si observa de cerca, notará que se quedó por fuera de la evaluación los datos de las regiones *North America* & *South America* de *Airline1* debido a que la tabla primaria, *Airline2*, no los contempla; generando así un tipo de *dirty data*: Datos faltantes (*missing data*). **Explicado todo lo anterior, podrá discernir que un *Blend*, por defecto, sería interpretado como un *Left Join* hecho desde la capa lógica: sobre la marcha.**