**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 (*category*) mes a mes y compararlo con el objetivo de ventas que tiene trazado para ese mismo marco temporal (mes a 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.**

### **Caso del Apartado 2. resuelto con *Blend***

¿Recuerda el requerimiento de la consulta inicialmente planteada en el caso del apartado *‘2. Joins de datos Vs Blend de datos’*? Recordemos.

Consulta: **Evaluar el rendimiento de cada departamento (*category*) mes a mes y compararlo con el objetivo de ventas que tiene trazado para ese mismo marco temporal, es decir, para cada mes.**

Pues bien, con el conocimiento que tiene sobre cómo poner en práctica correctamente el *Blending,* yapodría abordar la resolución a esta consulta haciendo un *Blend*. Observemos la resolución de la consulta sobre el departamento ***Furniture*** en la visualización***.***

***Gráfico, Gráfico de barras

Descripción generada automáticamente***

**IMPORTANTE**: Fue necesario agregar dos relaciones de combinación, entre el *Join* ‘ListOfOrders+OrderBreakdown’+ el *Blend* con **‘SalesTargets’**, para granular el ***target***(mensual) de cada departamento **mes a mes**. Observe.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Anotación**: La relación de combinación que anteriormente ya existía, por defecto, era: **Category + Category**.

#### **Introducción a Gráficos de Doble Eje**

Ahora bien, lo que nos interesa puntualmente de comparar los rendimientos reales obtenidos en cada departamento, mes a mes, Vs los *targets* inicialmente trazados, es **combinar ambos gráficos**; de tal manera que, el proceso de evaluar rendimientos reales vs *targets* sea más fácil. Es decir, la comparación visual se haga más fácil, rápida y efectiva. Si los gráficos están separados, como actualmente se encuentra la evaluación del departamento ***Furniture***, se hace difícil comparar si los departamentos están cumpliendo o no con sus objetivos: **he aquí la importancia de los gráficos de doble eje**, lo trataremos en el próximo apartado.

## **Gráficos de Doble Eje**

En este apartado aprenderemos a crear un gráfico de doble eje, habilidad tan simple como valorada que nos permite realizar *dashboards* en Tableau con visualizaciones que incorporen dos o más de dos variables: una a cada lado de los ejes del gráfico (Es preciso decir que las variables son, o deben ser, comparables entre sí porque guardan alguna relación; especialmente para algún requerimiento de alguna consulta). Para nuestro caso puntual, las variables a tratar son: Rendimientos reales (*Sales*) vs *Targets****.***

### **Añadiendo el Doble Eje**

***Hacer un doble eje*** es tan sencillo como hacer clic derecho sobre el eje de la segunda variable, que se encuentra relacionada con una primera, y estando ambas variables subordinadas a un campo; por ejemplo, si hacemos un símil con la resolución de la consulta sobre el departamento ***Furniture***, del apartado anterior, estaríamos hablando del campo ***Category*** que contiene a una primera variable ***Sales*** y a una segunda variable ***Target***; entonces, hecho clic sobre nuestra segunda variable (***Target***) seleccionamos la opción ***Eje Doble***, tal que así:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Resultado***:

Gráfico, Gráfico de barras, Gráfico de dispersión

Descripción generada automáticamente

### **Procure una visualización limpia**

Efectivamente, ya tenemos nuestros gráficos de doble eje; sin embargo, debe recordar que la utilidad de una visualización radica en facilitar la interpretación y el entendimiento de nuestros datos, no lo contario; dicho eso, nuestra visualización no es ni estética ni fácil de digerir. Consideremos usar un tipo de gráfico diferente para cada variable y, también, suprimir una de las dos etiquetas añadidas para cada variable en cuestión **(Idealmente deje las etiquetas para el gráfico que está siendo *pordebajeado* por otro, generalmente corresponde al gráfico de la primera variable; para este caso, *Sales*).**

**Dejemos algo claro. *Sales***es la ‘primera variable’, que está también contenida por el campo ***Category****,* debido a que ***Sales***está situada inmediatamente después de ***Category***en las filas; digamos, después de ***Category***se arrastró ***Sales***y, posteriormente, ***Target****.* Obsérvelo usted mismo:

  
En todo caso, observe como dos campos, unidos por un doble eje, se *unen* en las filas (o columnas).

Retomando la búsqueda de una visualización más limpia… ¿Qué tal si le asignamos un ***gráfico de*** ***área*** a nuestra variable ***Target***y ***suprimimos***, de esta misma variable, sus ***etiquetas*** asignadas? Veamos.

Gráfico

Descripción generada automáticamente

***Definitivamente mejor, más comprensible e interpretable.***

### **Sincronice siempre sus ejes**

De nuevo haga clic derecho sobre el eje de su segunda variable (ahora recostada sobre la parte más derecha de su visualización), ***Target,*** y seleccione la opción ***Sincronizar Eje.*** Con la *sincronización de ejes* usted podrá ver que habrá una correspondencia exacta, proporcionalmente, entre las medidas de los ejes de cada variable: los ejes de cada variable se alinean, se corresponden; lo cual, brinda una visualización más consistente y precisa de las dimensiones de cada gráfico para cada variable, pues, se corresponden las medidas de sus ejes. Veamos.

Gráfico, Gráfico de barras

Descripción generada automáticamente

## **Crear campos calculados en un Blend**

Ahora nos interesa saber puntualmente cuál es la diferencia entre los rendimientos reales por venta, de cada departamento (category), mes a mes, con relación a los objetivos trazados; es decir, restar ***Sales***con ***Target***directamente; con lo cual, tendríamos que crear un campo calculado entre ambas variables. Se habla de un campo calculado en un Blend porque, si bien recuerda, las variables en discusión se logran comparar visualmente entre sí a raíz de un ***Blend*** resuelto***.***

Entonces, se crea un nuevo gráfico a partir de cómo el campo calculado de nuestro interés toma a cada barra de ***Sales*** y le ***sustrae*** su área correspondiente del ***Target***. A partir de esta ***sustracción***, ya podemos visualizar mejor el comportamiento de cada departamento, mes a mes, y ver explícitamente si se superaron o no nuestras expectativas de ventas.

Como ahora tendríamos tres gráficos acumulados sobre cada categoría, lo cual ya empezaría a *saturar* exorbitantemente nuestra visualización, lo ideal sería entonces ***filtrar*** los gráficos por ***Categoría*** (Category). Hagamos eso primeramente. Arrastre ***Category*** de su fila, **sin suprimir ésta**, y pásela también a la casilla de filtros.

Ahora, usted podría suprimir ***Category*** de su fila; sin embargo, como queremos conservar ese mismo nivel de granularidad (**Category**) para no generar sesgos en la evaluación visual de los rendimientos de ventas Vs sus objetivos, que se evalúan justamente por ***Categoría***, usted tendría que arrastrar dicho campo a la *marca* ***Detalle***.En consecuencia, los datos de ***Sales*** vs ***Target*** se mantendrían consistentes y alineados a cada ***Categoría***.

**IMPORTANTE**: Ahora, si usted considera arrastrar ***Category*** a la *marca* ***Detalle,*** en vez de dejarla en filas, sus posibles campos calculados dentro del ***Blend*** también tendrían que estar granulados haciendo uso de la *marca* ***Detalle.***

Caso aparte, acá nuestra configuración de ***Filtro.***

Interfaz de usuario gráfica, Aplicación, Tabla, Word

Descripción generada automáticamente

### **Creando nuestro campo calculado**

Hagamos clic derecho sobre el campo ***Sales*** y seleccionemos la opción *Campo calculado…* después de hacer clic en la opción *Crear*. En el recuadro para la creación de su *Campo calculado* debe tener inicialmente lo siguiente:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Ahora, como le interesa restar su ***Target*** a ***Sales***, tiene que dirigirse a su otra fuente de datos y, desde ahí, arrastrar ***Target*** al recuadro de ***Campo calculado.*** Verá lo siguiente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En principio, la anterior ecuación marcaría *error* debido a que estamos intentando operar matemáticamente un campo agregado (***Target***) con otro que no (***Sales***); entonces, para que el cálculo sea legítimo y correcto (al menos en un ***Blend***), debemos *agregar* todos los campos. Tal que así:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**nota**: Al campo calculado le pasamos el nombre ***‘Excess of Target’***.

Ahora arrastre su nuevo campo calculado ***‘Excess of Target’*** a la fila,



Quedando su visualización así (*agregando también un poco de etiqueta y formato a* ***‘Excess of Target’***):

Gráfico

Descripción generada automáticamente

En nuestro nuevo gráfico de barras se nos muestra la diferencia visual exacta entre ***Sales*** & ***Target***; es decir, se muestra la representación gráfica del campo calculado ***‘Excess of Target’***.

**Adicionalmente,** si arrastrara ***Excess of Target*** a la *marca* ***Color*** sería conveniente para jugar con colores y lograr una mayor distinción visual entre los valores negativos (**Sales** < **Target**) Vs los valores positivos (**Sales** > **Target**).

## **El modelo de datos**

Diagrama

Descripción generada automáticamente

La anterior imagen refleja un muy buen ejemplo de lo que es un ***Modelo de datos.*** Básicamente se trata de una representación visual, sencilla, de una base de datos o esquema (*schema*) donde se evidencian los campos de tipo *Primary Key* que conectan entre sí cada uno de los *datasets* de la base de datos en cuestión. En Tableau estaremos capacitados para creer ***Modelos de datos,*** tal como éstos. Es muy probable que, dependiendo de la complejidad de la consulta, usted requiera hacer más *uniones* o, en general, cruces de datos (incluidos *cruces dobles)* para poder responder a sus preguntas.

### **Ni *Joins* ni *Blends:* Relaciones**

El modelo de datos cruza sus datos de forma diferente: no hace *Joins* ni *Blends*; si no, *Relaciones.* No hace *Joins* debido a que, como son muchos *datasets* diversos tratando de unirse entre sí, es casi que imposible encontrar que todos los *datasets* guarden el mismo nivel de *granularidad* (o en su defecto de *agregación*) que se rija a un mismo campo o columna *clave* en específico; de hecho, como puede observar en el ejemplo, hay muchos campos de tipo *Primary Key* uniendo diferentes *datasets* del mismo *schema* o base de datos; en consecuencia, son diversos los niveles de *granularidad …* hacer un *Join* sería un completo disparate.

¿Podría entonces considerar un *Blend*? Usted podría, sí, pero este recurso puntual sería poco práctico por la poca o nula eficiencia en el tratamiento de sus datos como un todo; pues, si bien se recuerda, tan solo un *Blend* entre dos *datasets* (o más) relacionados por un mismo campo, de tipo *Primary Key*,se ocuparía toda una *Worksheet*; entonces, como son muchas relaciones, muchas *Primary Key*, serían muchas las *Worksheets* que ocuparía para tratar cada una de las relaciones que hayan para las diversas conexiones que hay entre los *datasets* que posee.

Por todo lo anterior, es preciso que las *uniones* que pretenda hacer en un ***modelo de datos*** se den en la capa lógica; es decir, por medio de ***relaciones***. **En todo caso, si precisa más bien un *Join* entre 2 tablas de su mismo Modelo de datos, no una relación entre ellas, bien proceda.**

### **Su dataset central**

Ahora, algo puntual: El primer *dataset* que arrastrará al panel, para empezar a trazar sus relaciones, es el *dataset* que logre acumular el mayor número de relaciones con otros *datasets* de su misma base de datos, a este *dataset* se le llama: ***‘dataset central’***. Según nuestro ejemplo, este *dataset* sería: **‘olist\_orders\_dataset’. ‘olist\_orders\_dataset’** sería, en efecto, la tabla más central: la tabla que tiene más conexiones con elementos al exterior; es decir, con otras tablas.A partir del *dataset central* es donde se establecen el resto de las relaciones con las otras tablas. Todo esto nos facilita más la tarea de hacer nuestras conexiones totales.

### **Inconsistencias & Propiedades de archivos CSV**

#### **Inconsistencia en el Separador de Campos**

La tabla **‘olist\_orders\_dataset’,** como las demás tablas de su *database,* vienen en formato ‘.csv’. A veces, por alguna razón, Tableau no interpreta de manera correcta la configuración del archivo; en consecuencia, no lo estructura correctamente al ser importado (el formato *tabular* no se logra). Observe lo que nos acontece con nuestro primer archivo o *tabla* **‘olist\_orders\_dataset’** al arrastrarlo al panel.

Si se percata Tableau ha estructurado todos los datos del fichero en una sola columna; es decir, no está tomando en cuenta el ***delimitador*** con el que se está separando los datos (*comas*); por lo cual, presenta todos los datos de manera conjunta en una sola columna. **Para arreglar esto**, hagamos clic derecho sobre el *dataset* arrastrado en cuestión, revisemos las **propiedades del archivo de texto…**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

*Se nos abre el siguiente pop-up*

Texto

Descripción generada automáticamente con confianza baja

Si se da cuenta, el ***delimitador*** o separador de campo configurado por defecto es de *puntos y comas* y no *comas*; dicho eso, la solución correcta sería cambiar la configuración en la que se pueden separar los campos según Tableau: de *puntos y comas* a *comas,* y listo! Su tabla o *dataset* **‘olist\_orders\_dataset’** queda estructurada en formato tabular de manera correcta.

**Nota**: Es probable que este mismo error persista en sus demás tablas; por lo cual, a la hora de hacer una *relación* entre ellas, probablemente se verá así:



La solución es la misma.La novedad sería que luego le toque, manualmente, relacionar las *llaves* de cada tabla*.* **De hecho, habrá ocasiones en que el *bug* sólo consista en que le toque manualmente unir las *llaves*.**

#### **Inconsistencia en el Tipo de datos (Campos)**

En ocasiones sucede que el tipo de dato, de alguno o algunos campos de alguna tabla con la que pretenda trabajar, no es consistente; es decir, Tableau no interpreta de manera correcta el tipo de dato original, o pretendido originalmente, de la columna o campo en cuestión; haciendo que la columna presente sus datos con un *tipo de dato* incorrecto. Por ejemplo, cuando una columna que contiene números, y que debiera conservar su tipo de dato como *números*¸ conserva un tipo de dato de *textos* (strings)… Tableau está creyendo que los números representan cadenas de textos cuando, originalmente, por el origen de nuestra fuente de datos, se sabe que los números representan exactamente eso: *números.*

Pues bien, la anterior inconsistencia se debe a que, probablemente, **la configuración regional de nuestra maquina (computador) no corresponde a la configuración regional del archivo propiamente**. **Para arreglar esto…** Si usted por ejemplo toma una tabla o *dataset*,con la que pretende trabajar que está presentando este tipo de inconsistencia; digamos, el *dataset* **‘olist\_geolocation\_dataset’,** donde casualmente Tableau está mal interpretando el tipo de dato de sus columnas**Geolocation Lat & Geolocation Lng** (interpreta sus datos como tipo *textos*, en vez de conservar los datos como tipo *geográficos*),

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

y hace clic derecho sobre dicha tabla para acceder a sus **propiedades del archivo de texto** (el caso puntual que estamos abordando)…

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Notará que la **configuración regional** es ‘español (Colombia)’; sin embargo, la fuente de datos originalmente no proviene de Colombia; sino, para este caso puntual que estamos trabajando con el *schema* ‘P1-Brazilian-E-Commerce-Dataset’, la fuente de datos tiene un origen *inglés*; más puntualmente, se ha guardado en un formato *inglés.* **Entonces, para corregir la inconsistencia sobre el tipo de dato de las columnas de una tabla, lo ideal es cambiar la configuración regional de la tabla en cuestión;** es decir, que sea equivalente a la configuración regional de la fuente de datos importada.

Con una configuración regional en ‘inglés (Estados Unidos)’ sobre su *dataset* **‘olist\_geolocation\_dataset’,** verá que ahora el tipo de dato de sus columnas **Geolocation Lat & Geolocation Lng** sí serán consistentes:

Tabla

Descripción generada automáticamente con confianza media

Ahora, si bien **Geolocation Lng** dejó de ser de un tipo de dato *textos*,todavía su dato no es exactamente de tipo *geográfico*; entonces, para ser más puntuales a este punto, podemos irnos directamente a la *Worksheet* y personalizar el formato numérico a un formato que sea estrictamente de tipo *geográfico.* Observe cómo.

Ubique la tabla de su interés, donde contiene su columna **Geolocation Lng,** y siga los pasos a continuación:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Se ha seleccionado ***Función geográfica* +** ***Longitud*** debido a que la columna **Geolocation Lng** conserva un tipo de dato **geográfico** que evalúa especialmente **longitudes.** Con este clic derecho sobre una columna, podemos tanto formatear como convertir nuestros datos a otros tipos de datos.

### **Creando nuestras *Relaciones***

La diferencia principal entre un ***Join*** y una ***Relación*** es que, si bien ambas se *relacionan* por medio de un campo de tipo ***llave***; es decir, un *campo* que ambos poseen, una ***Relación*** mantiene las tablas independientes (separadas) y no se constituyen en una *macrotabla* por medio de la unión tangible (física) de los datos, como sí acontece en un ***Join***. Por lo anterior, es que se habla de una *unión* en la capa física (en el caso de un ***Join***).

Ahora, para crear una relación entre dos tablas en específico simplemente debe arrastrarlas al panel de manera conjunta. Para relacionar una tabla con otra debe fijarse de que, visualmente, esté relacionando una tabla A de su interés con una tabla B; de tal manera que, A y B estén explícitamente conectados: guíese de su Modelo de datos para observar qué tablas tienen un vinculo directo entre sí y qué no.

#### **Resultado final**

Diagrama

Descripción generada automáticamente

Cuando se dirige a su ***Worksheet***, luego de hacer todas sus ***Relaciones***, verá por separado y de forma independiente cada una de las *medidas* y *dimensiones* de cada tabla de su **Modelo de datos**. Lo dicho, en las ***Relaciones*** las tablas no dejan de mantenerse separadas; sin embargo, todas las columnas o campos los vería enlistados en el costado más izquierdo de Tableau, con lo cual podrá hacer ***drag & drop***: arrastrar y crear las visualizaciones con todos los datos importados que necesite.

## **Desafío: Trabajando con Relaciones**

Este apartado consistirá en poner en práctica las uniones de tipo *relación* mediante un caso de estudio por resolver, o *query* requerido, a partir del Modelo de datos previamente abordado en el apartado ‘6. El modelo de datos’.

### **Los datasets requeridos para nuestra consulta**

El planteamiento a la consulta que se requiere resolver es el siguiente:

* Eres un científico de datos en la tienda brasileña e-commerce. Tu encargado te ha pedido crear un mapa mostrando los **vendedores**, en sus **localizaciones**, con burbujas para ilustrar la cantidad de dinero en **pagos totales** que han recibido de los clientes. En consecuencia, se pretende saber qué vendedores son los que mayormente reciben ingresos operacionales por sus ventas concretadas.

Observe que para responder a esta consulta se requieren los datos de los **vendedores**, contenidos en el *dataset* o tabla **‘olist\_sellers\_dataset’**; los datos de las **localizaciones**, contenidos en el *dataset* o tabla **‘olist\_geolocation\_dataset’** & se requiere de los datos de **pagos totales**,contenidos en el *dataset* o tabla **‘olist\_order\_payments\_dataset’**; digamos, son los *datasets* que puntualmente precisamos visualizar en un gráfico de tipo *mapa* que responda a nuestra consulta.

Sin embargo, si se remite a su modelo de datos, tenemos que; si bien los *datasets* **‘olist\_sellers\_dataset’ & ‘olist\_geolocation\_dataset’** se encuentran relacionados directamente, por medio de un campo de tipo *llave,* no es el caso de **‘olist\_order\_payments\_dataset’** que, prácticamente, se encuentra al otro extremo de nuestro modelo de datos.

Entonces, para poder visualizar a **‘olist\_sellers\_dataset’ & ‘olist\_geolocation\_dataset’** + **‘olist\_order\_payments\_dataset’,** debemos pasar por todos los *datasets* que sirven de ‘puente’ para llegar a **‘olist\_order\_payments\_dataset’** justamente; es decir, debemos pasar por las *llaves* de **‘olist\_order\_orders\_dataset’** y **‘olist\_orders\_dataset’**; lo cual, en práctica, significa dejar establecidas, también, las relaciones de esas tablas en la capa lógica de Tableau hasta conectar con **‘olist\_order\_payments\_dataset’**…para nuestra suerte, eso ya lo hemos hecho en el apartado anterior. De hecho, hemos relacionado todas las tablas del *schema* a excepción del *dataset* **‘olist\_order\_reviews\_dataset’.**Observe nuevamente:

Diagrama

Descripción generada automáticamente

### **Tratamiento de datasets duplicados**

Ahora, aclaremos algo. Si se percata el *dataset* **‘olist\_geolocation\_dataset’** se encuentra duplicado en nuestras relaciones, esto debido a que hay un **‘olist\_geolocation\_dataset’** tanto para **‘olist\_sellers\_dataset’** como para **‘olist\_order\_customer\_dataset’**; es decir, en **‘olist\_geolocation\_dataset’** se conservan las localizaciones tanto de los clientes como de los vendedores (por decirlo de alguna manera); sin embargo, nosotros sólo precisamos el *dataset* **‘olist\_geolocation\_dataset’** que se relaciona con los ***vendedores***; es decir, el *dataset* **‘olist\_geolocation\_dataset’** conectado o relacionado a **‘olist\_sellers\_dataset’.**

**Para no confundirse,** y evitar que trabaje con el *dataset* **‘olist\_geolocation\_dataset’** relacionado a **‘olist\_order\_customer\_dataset’, lo ideal es que asigne un nombre personalizado a cada ‘olist\_geolocation\_dataset’.** Por ejemplo, para los vendedores, **‘olist\_geosellers\_dataset’** & para los compradores, **‘olist\_geocustomers\_dataset’.** Quedarían así nuestras tablas relacionadas en capa lógica:

Interfaz de usuario gráfica

Descripción generada automáticamente

### **Manos a la obra: Visualizando nuestro mapa**

Si ya tiene todos sus campos de interés listos para ser arrastrados a su *Worksheet*,y empezar a visualizar su mapa geográfico, lo ideal es que conserve tanto un campo de tipo geográfico que mida *longitudes* como un campo de tipo geográfico que mida *latitudes*. Posteriormente, para dejar visualizado el mapa inicial, lo único que tiene que hacer es un doble clic sobre cada campo de tipo *geográfico* mencionado previamente; exactamentelos que miden *longitudes* y *latitudes*, en este caso, los campos **Geolocation Lat & Geolocation Lng.** Su visualización ya tomaría forma, tal que así:

Mapa

Descripción generada automáticamente

#### **Visualizando relaciones entre varias tablas**

Al principio pueda que note que sus datos de tipo *geográfico* se *agreguen* como un todo al calcular un *promedio* con ellos; pues bien, es preciso que los *granule* por medio del campo *llave* **seller\_id** haciendo uso de la *marca* **Detalle** (Tal como lo evidenciamos en la anterior imagen); es por medio del campo *llave* **seller**\_**id** que se *granulan* todos los datos de los campos **Geolocation Lat & Geolocation Lng**, porque sólo nos interesa localizar por medio de la *longitud* y la *latitud* a cada uno de los vendedores (tal como lo requiere nuestra consulta); y, como bien se imagina, cada vendedor se identifica con un único **id**, es decir, con un valor único del campo o columna **seller\_id**

**seller\_id** nos garantiza también que no se duplicará información a la hora de localizar a cada vendedor; sin embargo, la columna que realmente conecta a ambas tablas es **zip\_code\_prefix**; pues, ambas la poseen.

También decir que esto es un gran primer ejemplo de cómo se ven *relacionados*, visualmente, diferentes *datasets*; especialmente, cómo se relacionan campos de diferentes *datasets* vinculados por medio de un Modelo de datos. Observe cómo hemos trabajado con columnas provenientes de dos *datasets* diferentes previamente relacionados: tabla **olist\_sellers\_dataset,** quien contiene la columna **seller\_id** & tabla **olist\_geosellers\_dataset,** quien contiene las columnas **Geolocation Lat & Geolocation Lng.**

Recordemos nuestra consulta:

* Eres un científico de datos en la tienda brasileña e-commerce. Tu encargado te ha pedido crear un mapa mostrando los **vendedores**, en sus **localizaciones**, con burbujas para ilustrar la cantidad de dinero en **pagos totales** que han recibido de los clientes. En consecuencia, se pretende saber qué vendedores son los que mayormente reciben ingresos operacionales por sus ventas concretadas.

Ahora, a este punto, necesitamos distinguir visualmente el valor de los pagos hechos (**Payment Value**) para cada vendedor (que ya fue previamente localizado) por medio de burbujas que varían de *tamaño* según el nivel de los pagos totales; **a mayor valor en el pago, mayor el tamaño de la burbuja.** Las burbujas, o círculos, son el tipo de figura dentro del *mapa* con el que se identificó, también, la localización de cada vendedor.

**Nota**: Si bien recuerdas, la columna **Payment Value** proviene de la tabla **olist\_order\_payments\_dataset,** una tabla que no tiene vínculo directo alguno, mediante *llaves* en común, con las tablas **olist\_geosellers\_dataset** u **olist\_sellers\_dataset.**

Sin embargo, como ya establecimos todas las *relaciones* de nuestro Modelo de datos, en la cual se incluye la tabla **olist\_order\_payments\_dataset**, no es necesario que haya un vínculo directo de **olist\_order\_payments\_dataset** con **olist\_geosellers\_dataset** o **olist\_sellers\_dataset**; de hecho, esto ya lo explicamos al inicio en ‘***Los datasets requeridos para nuestra consulta’.***

#### **Uso de marcas**

Dicho lo anterior, simplemente arrastremos la columna **Payment Value** a la *marca* **Tamaño.**

**Payment Value** es un campo de tipo *medida*, por lo cual debe asegurarse que Tableau y su *Worksheet* lo interpreten como tal (campo de color *verde*, generalmente *agregado*). Para la resolución de nuestra consulta, la función de agregación es de tipo ***suma*** debido a que nos interesa saber los pagos **totales** que recibió cada vendedor en todas sus negociaciones concretadas.Por lo pronto, su visualización se vería así:

Mapa

Descripción generada automáticamente

Ahora, distingamos visualmente a cada Estado de Brasil por medio de ***colores.*** Arrastre la columna ***Seller State***, de la tabla **olist\_sellers\_dataset,** a la *marca* **Colores** con una **Opacidad del 70%** para que se logren ver un poco más las *burbujas* que están por debajo de otras; es decir, para poder tener una visualización de mayor alcance sobre todos los vendedores (que las *burbujas* más grandes no se superpongan sobre las más pequeñas, sino que las más pequeñas se puedan ver través de las más grandes). Ahora, su visualización así:

Mapa

Descripción generada automáticamente

#### **Pestaña Mapas**

Usted podría personalizar la capa o el segundo plano, *background*, de su mapa; es decir, darle a su mapa un *formato* visual personalizado. Diríjase al menú principal de Tableau y haga clic en la pestaña ***Mapa***, posteriormente haga clic en ***Mapas en segundo plano.*** Observe:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Por defecto, el *background* del mapa es **Claro.** Para que me entienda mejor con lo que quiero decir con ‘background’, probemos el *background* de **Calles.**

Antes de observar nuestra visualización con un *background* de **Calles**, definamos un **límite** (*borde*) de color **negro** para nuestras burbujas, … para que resalten más (será necesario). Observe a continuación.

Mapa

Descripción generada automáticamente

Con ***Calles*** se obtiene un mapa tipo GPS y las burbujas, ahora, tienen ***limites*** (bordes) de color negro.