



# FileMaker

## Convenciones de Desarrollo

©2007 de FileMaker, Inc. Reservados todos los derechos. FileMaker es una marca comercial de FileMaker, Inc., registrada en los Estados Unidos de América y en otros países. El logotipo de la carpeta de archivos y ScriptMaker son marcas comerciales de FileMaker, Inc. Todas las demás marcas comerciales son propiedad de sus respectivos propietarios. Las especificaciones y disponibilidad del producto están sujetas a modificaciones sin aviso previo.

La documentación de FileMaker tiene su copyright. Al descargarla del sitio Web de FileMaker usted se compromete a no efectuar copias adicionales ni distribuir esta documentación sin el permiso por escrito de FileMaker.

ESTE DOCUMENTO SE SUMINISTRA "TAL CUAL» SIN GARANTÍA DE NINGUNA ÍNDOLE, Y FILEMAKER, INC. RECHAZA TODA GARANTÍA, YA SEA EXPRESA O IMPLÍCITA, INCLUYENDO SIN CARÁCTER LIMITATIVO LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZABILIDAD O APTITUD PARA UN FIN EN PARTICULAR O LA GARANTÍA DE NO INFRACCIÓN. EN NINGÚN CASO FILEMAKER INC. O SUS PROVEEDORES SERÁN RESPONSABLES POR DAÑOS Y PERJUICIOS, INCLUYENDO DAÑOS DIRECTOS, INDIRECTOS, INCIDENTALS, EMERGENTES, NI POR LUCRO CESANTE, DAÑOS PUNITIVOS O ESPECÍFICOS, AÚN EN EL CASO DE QUE FILEMAKER INC. O SUS PROVEEDORES HAYAN SIDO NOTIFICADOS RESPECTO DE LA POSIBILIDAD DE DICHS DAÑOS. EN ALGUNOS ESTADOS NO SE PERMITE LA EXCLUSIÓN O LIMITACIÓN DE RESPONSABILIDAD. FILEMAKER PUEDE EFECTUAR MODIFICACIONES A ESTE DOCUMENTO EN CUALQUIER MOMENTO SIN PREVIO AVISO. ESTE DOCUMENTO PUEDE CONVERTIRSE EN OBSOLETO Y FILEMAKER NO SE COMPROMETE A MANTENER ESTA INFORMACIÓN ACTUALIZADA.

Este documento técnico es una traducción y adaptación del documento técnico creado el 17 de Abril del 2007 en inglés: "FileMaker Development Conventions" (doc v1).

## Introducción

Programar con FileMaker® Pro es muy diferente a programar con otros entornos de desarrollo. Los desarrolladores de FileMaker tienen la libertad de crear y modificar rápidamente aplicaciones sin tener que ocuparse de muchas de las restricciones y preocupaciones a las que se enfrentan otros desarrolladores.

Al emplear un criterio interactivo para el diseño, una herramienta como FileMaker permite modificar y ampliar funcionalidades casi sin considerar las dependencias que existen en otras partes de las soluciones. Sin embargo, cuanto más complejas son las soluciones, tanto más difícil es mantenerlas.

Por ejemplo, sin apartarse del contexto de desarrollo ni navegar a través de numerosos cuadros de diálogo, es imposible comprender la información esencial acerca del principal objeto programable de FileMaker Pro: el campo. ¿Cuál es su tipo de datos? ¿Qué depende de él? ¿Cómo almacena los datos? ¿Cuál es su ubicación o participación en el gráfico de relaciones? ¿Cuál será su comportamiento en un guión, en una presentación, en un cálculo o en una lista de valores?

La flexibilidad de FileMaker Pro ofrece muchas formas únicas de abordar el desarrollo. Esto ha dado origen a innumerables normas redundantes aunque diferentes. Muchos desarrolladores de soluciones utilizan normas, pero cada uno utiliza las propias, ligadas a su metodología específica de desarrollo y a sus preferencias singulares. Algunos desarrolladores han difundido públicamente sus normas, pero la adopción de éstas ha sido limitada. Ninguna convención de nombres puede aplicarse a todas las metodologías de desarrollo y diseños de solución concebibles. Además, sería contraproducente intentar establecer semejante conjunto de reglas rígidas.

Lo que se necesita son recomendaciones de referencia para establecer un conjunto de criterios que deben tenerse en cuenta para cualquier tipo de convención de nombres. Estas recomendaciones de referencia constituirían el punto de partida que podrá ampliarse hasta lograr una convención más rica que satisfaga la necesidad de contar con soluciones y métodos de diseño más exigentes.

En respuesta a esta necesidad, FileMaker Inc. estableció un comité que reúne a expertos de la comunidad de desarrollo de FileMaker para crear consciencia y establecer pautas para las convenciones de nombres. Este grupo especializado, conocido como FileMaker Development Conventions Advisory Committee (FDCAC) [Comité Consultivo sobre Convenciones de Desarrollo de FileMaker], representa una muestra representativa y miembros asociados de FileMaker Solutions Alliance (FSA) del mundo entero que son expertos en materia de desarrollo. En conjunto, estos expertos representan cientos de años de experiencia de trabajo en proyectos individuales y empresariales, y en el desarrollo de paquetes de soluciones. Este comité ofrece una extensa variedad de opiniones y prácticas y representa un amplio espectro de criterios aplicables a las convenciones más que una visión específica.



El objetivo del FDCAC era crear el libro blanco de FileMaker Development Conventions (FDC) (Convenciones de Desarrollo FileMaker), cuyo objetivo es promover la coherencia y el profesionalismo entre las soluciones FileMaker.

La base de los desarrolladores de FileMaker es sumamente variada. La accesibilidad del producto hace de éste una clara elección para aquellos desarrolladores de aplicaciones sin experiencia, y la capacidad de rápido desarrollo de aplicaciones que ofrece esta herramienta es atractiva también para desarrolladores más experimentados. El nivel de esfuerzo que un desarrollador debe dedicarle a las convenciones de nombres es diferente en cada extremo del espectro.

Las FDC están diseñadas para brindar a todos los niveles cierto grado de interés, pero están especialmente diseñadas para el desarrollador de nivel intermedio. Las FDC no están dirigidas al usuario sin experiencia en FileMaker Pro. Sin embargo, tampoco están dirigidas a desarrolladores sumamente avanzados porque la orientación en convenciones podría no ser suficiente. De igual modo, los desarrolladores comerciales o en paquete podrían considerar que las FDC no son directamente aplicables dado que la complejidad y arquitectura de que se trate podrían requerir convenciones más estrictas y específicas. Pero estos desarrolladores pueden aprovechar algo de los conocimientos que aporta el documento para afinar aún más sus convenciones específicas.

Las recomendaciones generales de las FDC brindan a los desarrolladores una plantilla con la cual trabajar. Las recomendaciones "en cumplimiento de las normas" no están dirigidas a todas las arquitecturas o metodologías concebibles.

Sin embargo, sí aportan una orientación bien fundada acerca de dónde comenzar, lo cual posiblemente evolucione con el paso del tiempo.

El método seguido por el FDCAC consistió en identificar y posteriormente definir los problemas específicos para hacer frente a los cuales están diseñadas las convenciones de nombres. Una vez identificados los problemas, el comité buscó los puntos en común entre las numerosas variantes. En muchos casos, la simple definición de un conjunto de pautas coherentes permite abordar el punto conflictivo. Sin embargo, existen otras áreas temáticas en las cuales un estilo particular o metodología de desarrollo determina la convención que se debe aplicar. En estos casos, se ha decidido que las FDC aporten recomendaciones para normas generales y consideraciones adicionales que ofrezcan una mejor perspectiva acerca de la forma en que algunos desarrolladores han contemplado esas áreas más avanzadas.

Las FDC están constituidas por múltiples temas que cubren casi todo aspecto dentro del entorno de desarrollo FileMaker en el cual desarrollador podría aplicar convenciones de nombres. Los temas incluyen:

- Asignación de nombres de archivos
- Asignación de nombres de tablas
- Asignación de nombres de campos
- Asignación de nombres de Ocurrencia de tablas
- Asignación de nombres de presentaciones
- Documentación y formato de cálculos
- Asignación de nombres de listas de valores
- Documentación y formato de funciones personalizadas
- Asignación de nombres de cuentas
- Documentación de adhesión a las FDC

Cada sección está compuesta por tres subsecciones, la primera de las cuales es la sección de Definición de Problemas. Esta sección identifica las áreas clave a las cuales el desarrollador podría aplicar una convención:

- Abordar una limitación del entorno de desarrollo
- Evitar todas las consabidas dificultades que presenta una convención, o
- Aplicar coherencia, comprensión y documentación.



En la segunda sección de las FDC se analizan las recomendaciones para constatar su “cumplimiento de las normas” según lo designa el FDCAC. Esto provee una plantilla, un intérprete de comandos, si se comprende mejor, que los desarrolladores pueden utilizar cuando aplican convenciones a sus proyectos.

Por último, el documento trata algunos ejemplos de la forma en que los miembros del FDCAC han dado tratamiento a las convenciones en cada una de las áreas temáticas. El objetivo es brindar una muestra de los conceptos y ampliaciones que superen lo que prevé el “cumplimiento de las normas” de las FDC.

Aunque FileMaker cree que todos los niveles de desarrolladores se pueden beneficiar con la información dentro de las FDC, el mayor benefactor es la comunidad FileMaker. Se trata del primer paso en un proceso evolutivo.

Obviamente, este documento no es para todos. Si el desarrollador no experimentado está comenzando con FileMaker lo alentamos a continuar haciendo lo que está haciendo, pero también le recomendamos leer atentamente las FDC para que se familiarice con sus conceptos. Si está diseñando la próxima aplicación de éxito seguro (“killer application”) y ha desarrollado sus propias convenciones para poder aplicar metodologías más avanzadas, esto es una especie de revisión.

De todos modos, esperamos que tenga en cuenta la información y que trate de utilizarla siempre que sea posible.

En el caso de los desarrolladores de nivel intermedio, tienen a su disposición el conocimiento que poseen los socios y asociados expertos en FileMaker, por lo tanto debe aprovecharlo. La guía que ofrecen los documentos FDC puede ayudar a acelerar la utilización de soluciones profesionales y duraderas.

## **Prefacio**

En modo alguno el documento de las Convenciones de Desarrollo FileMaker (FDC) tiene por objeto que se lo considere como la forma “correcta” u obligatoria en que el desarrollador debe desarrollar sus soluciones FileMaker. Sería contraproducente intentar establecer semejante conjunto de reglas rígidas. Después de todo, uno de los hitos de FileMaker es su flexibilidad. Ni FileMaker ni ninguna otra institución pueden ordenar qué metodología se utilizará en una solución o convención de nombres determinada. Tampoco creemos que alguna institución o que una única institución tenga la perspectiva suficiente y necesaria para establecer una convención. El esfuerzo invertido aquí apunta a establecer un primer paso en aquello que seguramente constituirá un proceso evolutivo. Su éxito no depende de una sola persona. La comunidad FileMaker, en conjunto, debe colaborar con este esfuerzo y nutrirlo colectivamente a través de su evolución. Por supuesto que existe motivación dentro de la comunidad de desarrollo FileMaker para utilizar este marco de normas, dado que todos conocemos sus beneficios en materia de comunicación y durabilidad.

De este esfuerzo se desprenden ventajas considerables. Los equipos de desarrollo en cooperación pueden trabajar más fácilmente unos con otros. Los desarrolladores de instituciones diferentes pueden comprender con rapidez el código de los demás, y aún su propio código años después. Este esfuerzo permite un enfoque más coherente y comprensible en la creación de plantillas y soluciones abiertas que se comparten. Permite establecer una documentación clara y coherente. Y algunas de las pautas más simples pueden impedir la ocurrencia de problemas de evolución de las soluciones comunes así como de conflictos de compatibilidad.

La versión 1.0 de las FDC fue desarrollada con la línea de productos FileMaker 7, como base de su contenido. Se tuvo en cuenta también FileMaker 8 en los casos en que era fácilmente aplicable; sin embargo, se decidió que las FDC no se centraran, en detalle, en las nuevas funciones disponibles de la línea de productos FileMaker 8. Esta decisión era coherente con una de las metas primarias, que era aportar orientación basada en el “mundo real”; esto no existía antes de la elaboración de este documento. Los futuros documentos de las FDC darán orientación adicional si fuera necesario.

Los desarrolladores nuevos y experimentados se beneficiarán por igual con la ocurrencia de las FDC. A medida que el producto FileMaker evoluciona, las innumerables formas en que enfrentamos los problemas también cambiarán. La comunidad necesita de este esfuerzo sostenido y con el apoyo de la comunidad FileMaker al igual que la línea de productos FileMaker, posiblemente se perfeccione con cada versión



## Agradecimientos

Deseamos agradecer especialmente a cada una de las siguientes empresas su generoso aporte de tiempo y profundos conocimientos a este proyecto.

Beezwax	<a href="http://www.beezwax.net">http://www.beezwax.net</a> Vince Menanno Rick Aguire
Core Solutions	<a href="http://www.coresolutions.ca">http://www.coresolutions.ca</a> Steve Hearn
Dataworks	<a href="http://www.dataworks.ca">http://www.dataworks.ca</a> James Hea
DataWaves International	<a href="http://www.data—waves.com">http://www.data—waves.com</a> Peter Makin Colleen Hammersley
FileMaker	<a href="http://www.filemaker.com">http://www.filemaker.com</a>
Geist Interactive	<a href="http://www.geistinteractive.com">http://www.geistinteractive.com</a> Todd Geist
InResonance	<a href="http://www.inresonance.com">http://www.inresonance.com</a> Corn Walker
iSolutions	<a href="http://www.isolutions.com">http://www.isolutions.com</a> Cris Ippolite
Management Counselling Services	<a href="http://www.fmp—power.com">http://www.fmp—power.com</a> Steven Blackwell
The Moyer Group	<a href="http://www.moyergroup.com">http://www.moyergroup.com</a> Chris Moyer
New Millenium Communications	<a href="http://www.nmci.com">http://www.nmci.com</a> Danny Mack
Soliant Consulting	<a href="http://www.soliantconsulting.com">http://www.soliantconsulting.com</a> Steve Lane Scott Love Rodger Jacques
The Support Group	<a href="http://www.supportgroup.com">http://www.supportgroup.com</a> Chad Novotny



# Índice

<b>1</b>	<b>▷ ARCHIVOS.....</b>	<b>9</b>
1.1	Objetivos:.....	9
1.2	Definición de problemas:.....	9
1.3	Pautas “en cumplimiento de las normas” (Archivos):.....	10
1.4	Consideraciones complementarias:.....	12
<b>2</b>	<b>▷ ASIGNACIÓN DE NOMBRES DE TABLAS .....</b>	<b>13</b>
2.1	Objetivos:.....	13
2.2	Definición de problemas:.....	13
2.3	Pautas “en cumplimiento de las normas” (Tablas):.....	14
2.4	Consideraciones complementarias:.....	15
<b>3</b>	<b>▷ ASIGNACIÓN DE NOMBRES DE CAMPOS .....</b>	<b>16</b>
3.1	Objetivos:.....	16
3.2	Definición de problemas:.....	17
3.3	Pautas “en cumplimiento de las normas” (Campos):.....	17
3.3.1	Campos generales.....	18
3.3.2	Campos clave/de coincidencia.....	19
3.3.3	Campos auxiliares.....	20
3.3.4	Referencia a la notación del campo.....	22
3.4	Consideraciones complementarias:.....	23
<b>4</b>	<b>▷ OCURRENCIAS DE TABLAS .....</b>	<b>24</b>
4.1	Objetivos:.....	24
4.2	Descripción de problemas:.....	25
4.3	Pautas “en cumplimiento de las normas” (Ocurrencias de Tablas):.....	26
4.3.1	Método de Agrupamiento Funcional tipo Araña (FSG) Functional Spider Grouping.....	26
4.3.1.1	Ventajas del Agrupamiento Funcional tipo Araña (Functional Spider Grouping FSG).....	28
4.3.1.2	Desventajas del Agrupamiento Funcional tipo Araña (Functional Spider Grouping).....	28
4.3.1.3	Agrupamiento Funcional tipo Araña (Functional Spider Grouping): Pautas “en cumplimiento de las normas” ...	28
4.3.2	Método de los Grupos Funcionales de Ocurrencia de Tablas (Functional Table Occurrence Group FTOG).....	28
4.3.2.1	Ventajas del Agrupamiento Funcional de Ocurrencia de Tablas (FTOG).....	29
4.3.2.2	Ventajas del Agrupamiento Funcional de Ocurrencia de Tablas (FTOG).....	30
4.3.2.3	Grupos Funcionales de Ocurrencia de Tablas: Pautas “en cumplimiento de las normas”.....	30
4.3.3	Método de Agrupamiento de Ocurrencias de Tablas Base y Flotantes (Anchor Buoy) / Agrupamiento Jerárquico (Hierarchical Table Occurrence).....	32
4.3.3.1	Ventajas del método de agrupamiento de Ocurrencias de tablas base y flotantes (Anchor Buoy) / (HTOG).....	33
4.3.3.2	Desventajas del método de agrupamiento de Ocurrencias de tablas base y flotantes (Anchor Buoy) / (HTOG).....	33
4.3.3.3	Pautas de agrupamiento de Ocurrencias de tablas base y flotantes (Anchor Buoy) / HTOG “en cumplimiento de las normas”.....	34
4.4	Consideraciones complementarias.....	35



<b>5</b>	<b>▷ PRESENTACIONES .....</b>	<b>37</b>
5.1	Objetivos: .....	37
5.2	Definición de problemas:.....	38
5.3	Pautas "en cumplimiento de las normas" (Presentaciones):.....	39
<b>6</b>	<b>▷ FUNCIONES PERSONALIZADAS .....</b>	<b>40</b>
6.1	Objetivos: .....	40
6.2	Definición de problemas:.....	41
6.3	Pautas "en cumplimiento de las normas" (Funciones personalizadas):.....	41
6.3.1	Funciones personalizadas públicas .....	41
6.3.2	Funciones personalizadas privadas .....	42
6.3.3	Parámetros de la función personalizada .....	43
6.3.4	Ejemplos de asignación de nombres a funciones personalizadas.....	43
6.3.5	Documentación de las funciones personalizadas.....	44
6.3.6	Formato de la función personalizada .....	44
6.4	Consideraciones complementarias .....	44
<b>7</b>	<b>▷ GUIONES .....</b>	<b>45</b>
7.1	Objetivos: .....	45
7.2	Definición de problemas:.....	45
7.3	Pautas "en cumplimiento de las normas" (Guiones):.....	45
7.3.1	Nombres de guiones .....	45
7.3.2	Variables .....	46
7.3.3	Documentación de guiones .....	46
<b>8</b>	<b>▷ CÁLCULOS.....</b>	<b>48</b>
8.1	Objetivos: .....	48
8.2	Definición de problemas.....	48
8.3	Pautas "en cumplimiento de las normas" (Cálculos): .....	48
8.3.1	Variables .....	48
8.3.2	Comentarios de la cabecera del bloque de cálculo.....	49
8.3.3	Comentarios de cálculos en el renglón .....	49
8.3.4	Formato de cálculos.....	50
8.3.4.1	Ejemplo de formato #1 .....	50
8.3.4.2	Ejemplo de formato #2 .....	51
<b>9</b>	<b>▷ LISTAS DE VALORES .....</b>	<b>54</b>
9.1	Objetivos: .....	54
9.2	Definición de problemas:.....	54
9.3	Pautas "en cumplimiento de las normas" (Listas de valores): .....	55



<b>10</b>	<b>▷ CUENTAS Y SEGURIDAD.....</b>	<b>57</b>
10.1	Objetivos:.....	58
10.2	Definición de problemas .....	58
10.3	Pautas “en cumplimiento de las normas” (Seguridad): .....	59
10.3.1	Conjuntos de privilegios.....	59
10.3.2	Nombres de cuentas autenticadas internamente .....	59
10.3.3	Nombres de cuentas autenticadas externamente (Nombres de grupo).....	60
10.4	Consideraciones complementarias:.....	61
<b>11</b>	<b>▷ ADHESIÓN .....</b>	<b>63</b>
11.1	Objetivos de la convención.....	63
11.2	Definición de problemas .....	63
11.3	Pautas de convenciones en cumplimiento de las normas .....	63
<b>APÉNDICE A</b>	<b>▷ PALABRAS RESERVADAS DE SQL .....</b>	<b>65</b>
<b>APÉNDICE B</b>	<b>▷ TABLA DE USO DE CARACTERES.....</b>	<b>74</b>
<b>APÉNDICE C</b>	<b>▷ TERMINOLOGÍA Y DEFINICIONES .....</b>	<b>76</b>
<b>APÉNDICE D</b>	<b>▷ LEYENDA DE SINTAXIS .....</b>	<b>77</b>



# 1 ▷ Archivos

Las necesidades de diseño y la arquitectura de una solución incidirán en el hecho de que esté constituida por un archivo único o por múltiples archivos. Los diseños pueden incluir un único archivo que contenga todas las tablas, o arquitecturas de separación que puedan dividir los datos y la interfaz en diferentes archivos. Las soluciones pueden contener múltiples archivos utilitarios utilizados para módulos destinados a informes o a soluciones. En todo caso, una solución está formada ya sea por un archivo único o por múltiples archivos. Para cada uno de estos archivos deben considerarse varios factores durante el proceso de asignación de nombres.

Se debe utilizar un conjunto de caracteres que no obstaculice la conectividad a la solución desde tecnologías tales como ODBC, JDBC y XML. Elija entre mayúsculas y minúsculas y sea coherente al respecto. Además de las mayúsculas y/o minúsculas, debe pensarse en cómo separar las palabras del nombre. Por otra parte, debe pensarse la forma de separar sintácticamente el nombre del archivo en sí de los metadatos, prefijos o sufijos. En este documento, nos referimos a dicha separación como separación sintáctica. Elegir si se utilizarán nombres en singular o en plural en el nombre del archivo será otro paso a tener en cuenta para una presentación profesional. Debe determinarse cómo agrupar el conjunto de archivos en forma tal de dar una indicación visual de que están conectados entre sí. A partir del FileMaker 7, los desarrolladores pueden construir soluciones "todo en uno" en las cuales cada tabla está autocontenida dentro del mismo archivo. Esta última no siempre es la mejor elección, por razones que exceden el alcance de este documento. En el caso de soluciones formadas por archivos múltiples, desde una perspectiva administrativa y evolutiva, es importante que se proporcione alguna convención que lo especifique claramente. Finalmente, deben tenerse en cuenta las consideraciones referidas a multiplataformas así como también algunas recomendaciones generales sobre versiones.

## 1.1 Objetivos:

- ✓ Conjunto de caracteres recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada
- ✓ Separación sintáctica recomendada
- ✓ Longitud del nombre de archivo recomendada
- ✓ Agrupamiento recomendado de la solución de archivos múltiples
- ✓ Pautas para singular vs. plural
- ✓ Pautas para versiones
- ✓ Pautas para extensiones
- ✓ Cuestiones relacionadas con tecnologías de conectividad (XML, ODBC, JDBC), incluso palabras reservadas de SQL.

## 1.2 Definición de problemas:

**Conjunto de caracteres recomendado:** Existen algunas restricciones básicas respecto de los caracteres que admitirá FileMaker. Por otra parte, algunos caracteres pueden provocar conflictos con la conectividad externa o con el intercambio de datos en algunos Sistemas de Administración de Bases de Datos Relacionales (RDBMS). El desarrollador debe estar al tanto de las restricciones tanto de FileMaker como de cualquier sistema externo con el cual interactuarán sus soluciones.

**Convención de mayúsculas/minúsculas recomendada:** La recomendación, en este caso, no consiste en superar un problema específico sino en aplicar un criterio coherente para adoptar la manera de dar nombres a los archivos. El objetivo es seleccionar un método y ser coherente al respecto.

**Separación sintáctica recomendada:** La separación sintáctica para los nombres de archivo se refiere a la forma en que se diferencia cualquier prefijo y/o sufijo del nombre del archivo. Es habitual ver prefijos o sufijos utilizados para indicar que un archivo es parte de un grupo de archivos. También se utilizan para contener características de identificación. El objetivo es aportar una forma coherente y universalmente reconocida de diferenciar estos elementos de metadatos principales/continuas del nombre del archivo en sí.



**Pautas para versiones:** Las referencias de los archivos se mantienen mediante el nombre del archivo. Para evitar conflictos con las referencias de los archivos a medida que cambian las versiones, el desarrollador debe evitar el uso de nombres de archivos para representar la versión de una solución.

**Pautas para extensiones:** En Windows, al archivo siempre se le asignará la extensión 'fpX'. En Macintosh, la extensión utilizará por defecto la extensión pero esto es opcional. Lo anterior se transforma en un problema cuando se trata de alojar un archivo sin una extensión desde un servidor Windows, en cuyo caso al iniciarse la transacción el servicio busca la extensión .fpX para colocar el archivo. La convención debe aplicar un criterio coherente para una extensión que trabajará en todas las plataformas admitidas.

**Pautas para singular vs. Plural:** La decisión de utilizar nombres en singular o colectivos en lugar de la alternativa en plural se indica en el presente documento sólo con el fin de sugerir a los desarrolladores que tomen una decisión respecto de cuál utilizará y que sean coherentes con dicha decisión.

**Cuestiones de longitud del nombre de archivo:** El Sistema Operativo (OS) limita la longitud del nombre de archivo. Sin embargo, tener un nombre de archivo de 1000 caracteres no es práctico. No obstante, el desarrollador debe considerar una longitud razonable del nombre del archivo y debe ser consciente de que algunas aplicaciones que necesite utilizar presentarán diversas limitaciones. Por ejemplo, un nombre de archivo largo puede presentar dificultades al tratar de adjuntarlo a un correo electrónico. Si se necesita interactuar con otras aplicaciones, hay que tener presente que la longitud del nombre del archivo puede ser un problema.

**Agrupamiento recomendado de la solución de archivos múltiples:** En toda solución de archivos múltiples existe un archivo o archivos principales y archivos secundarios. Una solución de archivos múltiples tendrá como mínimo un archivo principal. El o los archivos principales son 'expuestos' al usuario final. En una solución hospedada este o estos archivos serían los archivos que el usuario debe seleccionar en el cuadro de diálogo anfitrión. En una solución de base local este o estos archivos son los que el usuario debe seleccionar directamente desde el Sistema Operativo o a través de un acceso directo / alias. Los archivos de soporte son los archivos subordinados o aquéllos no 'expuestos' al usuario para acceso directo a una solución. En una solución hospedada, éstos son los archivos que se utilizan para soportar la función del archivo principal. Además, existen arquitecturas que separan la interfaz del usuario de la capa de datos, en cuyo caso se trata generalmente de dos archivos separados. Algunos diseños colocan todos los archivos en el servidor, divididos entre cliente y servidor, y aún entre servidores múltiples. El objetivo en este caso es reconocer que esos archivos, independientemente del lugar en que estén ubicados, son parte de la misma solución o están conectados en alguna forma entre sí. Existen varias razones por las cuales este aspecto es importante. Desde una perspectiva administrativa, dicho aspecto facilita mucho la gestión y administración de soluciones para movimientos, seguridad y copias de seguridad. Desde el punto de vista del desarrollo, tal aspecto aporta un método coherente que brinda una indicación visual de un agrupamiento de soluciones.

**Conectividad externa:** Al interactuar con cualquier RDBMS externo, deberá tener en cuenta todo conflicto que pudiera surgir con el nombre del archivo, ya sea debido a caracteres o palabras reservadas. Además, cuando se accede a los archivos FileMaker desde otras tecnologías, como por ejemplo ODBC, JDBC o XML, debe especificarse el nombre de archivo.

### 1.3 Pautas "en cumplimiento de las normas" (Archivos):

1. Sólo deben utilizarse los siguientes caracteres:
  - aA - zZ mayúsculas y minúsculas
  - Número 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple o doble "\_ " " \_"
2. NO debe contener espacios.
3. NO debe comenzar con números.
4. NO debe utilizar el nombre de archivo para indicar la versión.
5. NO debe contener puntos además del punto único utilizado para la extensión.
6. Debe utilizar en forma coherente nombres en singular o en plural.
7. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:



## Métodos de separación de palabras recomendados (Nombres de archivo) :

Minúsculas y Camelcase	nombreDeMiArchivo
Minúsculas y Camelcase	NombreDeMiArchivo
Subrayado simple (minúsculas)	nombre_de_mi_archivo
Subrayado simple (Tipo Titulo)	nombre_De_Mi_ Archivo
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_ARCHIVO

8. Debe contener la extensión .fp7 (en minúsculas), independientemente de la plataforma. Sin embargo, las soluciones de ejecución pueden utilizar cualquier extensión, debe tenerse cuidado de no utilizar otras extensiones registradas.
9. Las soluciones de archivo único consisten en archivos que comprenden todas las tablas para la solución completa. No existen recomendaciones adicionales.
10. Soluciones de archivos múltiples: Son grupos de archivos que comprenden todas las tablas para la solución completa. En toda solución de archivos múltiples existen archivos primarios y archivos de soporte. El principal objetivo del agrupamiento de soluciones múltiples es identificar fácilmente los archivos que componen una solución. En muchos casos, se pueden utilizar nombres naturales. Por ejemplo, consideremos una solución con 4 archivos que se denominan de la siguiente forma:
  - Helpdesk.fp7
  - Tickets.fp7
  - Reports.fp7
  - Inventory.fp7

Los nombres no agrupan estos archivos en forma evidente. No obstante, con algunas modificaciones sería fácil hacerlo utilizando nombres naturales. Sin embargo, hay casos en que los nombres naturales no son útiles. O en todo caso puede decidirse dar un tratamiento más coherente a la cuestión de los nombres de archivo, independientemente del hecho de que un nombre natural pueda ser útil en algunos casos. Cabe analizar el mismo ejemplo utilizando la recomendación de cumplimiento de las normas. Esto ayuda en algunas áreas: En primer lugar, identifica claramente el archivo principal, en segundo lugar identifica claramente los archivos de soporte, y por último, presenta el conjunto de archivos como grupo.

- Helpdesk\_hd.fp7
- hd\_Tickets.fp7
- hd\_Reports.fp7
- hd\_Inventory.fp7

- 10.1. Archivo/s de soporte: Cada archivo de soporte debe contener un prefijo, definido aquí como Identificador Lógico de Solución (Logical Solution Identifier (LSI)). Para cada archivo de soporte presente en la solución debe utilizarse el mismo LSI. El LSI debe observar las convenciones generales y siempre debe estar separado del nombre con dos signos de subrayado "\_\_". El uso de dos subrayados ayuda a identificar el LSI del nombre del archivo, en el cual se puede utilizar un subrayado simple como separador de palabras. El LSI puede presentar cualquier longitud pero ésta debe ser coherente a través de toda la solución.

Sintaxis (Archivos de soporte)

<<LSI>>[ \_\_ ]NombreDeMiArchivo[ .fpX ]

Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.

- LSI (Identificador lógico de solución); obligatorio, definido por el desarrollador  
Denota archivos que están conectados lógicamente
- "\_\_" - obligatorio  
Separa el LSI del Nombre de Archivo
- NombreDeMiArchivo  
Nombre de archivo descriptivo



- .fpX – obligatorio  
Todos los archivos deben contener el "." (punto) seguido por la extensión fpX, donde X es la versión de formato de archivo de FileMaker. Ejemplo: .fp7
- 10.2. Archivo/s principal/es: Puede optarse por la inclusión del LSI en el o los archivos principales como prefijo o como sufijo. En uno u otro caso, el LSI siempre debe estar separado del nombre con dos subrayados "\_\_".
- Sintaxis (Archivos primarios)
- NombreDeMiArchivo[ \_\_ ] { LSI } [ . fpX ]  
{ LSI } [ \_\_ ] NombreDeMiArchivo [ . fpX ]
- Ver en el punto [Leyenda de sintaxis](#) (Apéndice D) la descripción de ésta.
- "Prefijo" LSI (Identificador lógico de solución); obligatorio, definido por el desarrollador  
Denota archivos que están conectados lógicamente
  - "\_\_" – obligatorio  
Separa el LSI del Nombre de Archivo
  - NombreDeMiArchivo  
Nombre de archivo descriptivo
  - "Sufijo" LSI (Identificador lógico de solución); obligatorio, definido por el desarrollador  
Denota archivos que están conectados lógicamente
  - .fpX – obligatorio  
Todos los archivos deben contener el "." (punto) seguido por la extensión fpX, donde X es la versión de formato de archivo de FileMaker. Ejemplo: .fp7
- 10.3. Ejemplos:
- nombreDeMiArchivo\_\_xxx.fp7; Archivo primario
  - xxx\_\_nombreDeMiArchivoDeSoporteA.fp7 – Archivo de soporte
  - xxx\_\_nombreDeMiArchivoDeSoporteB.fp7 – Archivo de soporte
  - xxxx\_\_nombreDeMiArchivo.fp7 – Archivo primario
  - xxxx\_\_nombreDeMiArchivoDeSoporteA.fp7 – Archivo de soporte
  - xxxx\_\_nombreDeMiArchivoDeSoporteB.fp7 – Archivo de soporte
  - Nombre\_De\_Mi\_Archivo.fp7; Archivo primario
  - ABC\_\_Nombre\_De\_Mi\_Archivo\_De\_Soporte\_a.fp7 – Archivo de soporte
  - ABC\_\_Nombre\_De\_Mi\_Archivo\_De\_Soporte\_b.fp7 – Archivo de soporte

## 1.4 Consideraciones complementarias:

- **Archivos prestados:** En algunos casos una solución puede incluir referencias a otro archivo que no está relacionado individualmente con la solución específica. Por ejemplo, un directorio de empleados podría utilizarse en muchas soluciones. En estos casos se recomienda tratar el archivo 'prestado' como si fuera su propia solución y no incorporarlo como parte de una solución específica.
- **Almacenamiento de archivos cliente:** Con fines organizativos y de mayor claridad, debe analizarse la posibilidad de colocar archivos de soporte que residan en una unidad local en una subcarpeta/ subdirectorio incluido en el lugar donde está ubicado el archivo principal. En aquellas soluciones que diseminan archivos tanto en el servidor como en el cliente, corresponde analizar la posibilidad de crear una carpeta de programas y de alojar los archivos principales del lado cliente en la raíz de esta carpeta, con todos los archivos de soporte ubicados en una subcarpeta..



## 2 ▷ Asignación de nombres de tablas

Los nombres de tablas no están mayormente expuestos dentro del entorno de desarrollo.

La interacción directa con las tablas generalmente se limita a relacionar una tabla con una Ocurrencia de la Tabla que se encuentra en el Gráfico de relaciones. Todas las interacciones entre datos se efectúan a través de la Ocurrencia de la Tabla. Es importante efectuar esta distinción. En tanto que muchos de los cuadros de diálogo utilizarán el término 'Tabla', el desarrollador en realidad utilizará la 'Ocurrencia de la Tabla'.

Al seleccionar nombres de tablas hay varios factores que deben tenerse en cuenta. Como con todas las áreas, debe seleccionarse un conjunto de caracteres que no obstaculice la conectividad a las soluciones desde tecnologías tales como ODBC, JDBC y XML. Para las tablas reales en sí esto no es importante en forma directa, debido al hecho de que no se hará referencia directa a las tablas. No obstante, si se decide utilizar el nombre de la Tabla de origen en su nombre de Ocurrencia de la Tabla, el cual se utiliza para llamadas de conectividad y al cual se hace referencia a través de todas las interacciones, habrá que ser más selectivo. Obviamente, no toda solución utilizará conectividad externa, pero dado que las soluciones evolucionan, es mejor estar preparado y prever esa posibilidad.

Por razones de coherencia y profesionalismo, deberá elegirse entre mayúsculas, minúsculas y otros tipos, y ser coherente con la elección. Deberá seleccionarse la forma en que se separarán las palabras individuales dentro del nombre de la tabla.

Debe tenerse presente cómo se separará sintácticamente el nombre de la tabla en sí de todo metadato, prefijo o sufijo. Se podría utilizar la separación sintáctica para nombres de tablas si el desarrollador decide indicar alguna función o categoría dentro del nombre. Por ejemplo, indicar dentro del nombre una tabla de 'unión' o 'sesión' utilizando un prefijo, por ejemplo: 'j\_\_TableNName' o 'ses\_\_table\_name'. En ambos casos, el separador sintáctico es un signo de subrayado doble "\_\_". Hay que ser coherente con el uso de nombres en singular o plural. Debe analizarse la forma en que el uso del 'espacio en blanco' puede ayudar al desarrollador a agregar una determinada organización a la tabulación de la Tabla. Esto puede ser muy útil cuando se trata de grandes cantidades de tablas. Por último, aunque un nombre de tabla puede tener 100 caracteres, por varias razones esto no resulta práctico.

### 2.1 Objetivos:

- ✓ Conjunto de caracteres recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada
- ✓ Recomendaciones para singular vs. plural
- ✓ Recomendaciones relacionadas con la longitud del nombre de la tabla
- ✓ Opciones de espacio en blanco para abordar la falta de organización
- ✓ Pautas para asignación de nombres de tipo de tabla (es decir, Unión, Utilitario, Sistema, Referencia)

### 2.2 Definición de problemas:

**Conjunto de caracteres recomendado:** Nombres de tablas que son protegidos de algún modo contra cualquier problema relacionado con los caracteres debido a su uso limitado en toda la aplicación. Sin embargo, se deben considerar de alguna forma los caracteres utilizados suponiendo que el desarrollador puede incluir el nombre de la tabla en otras áreas, como por ejemplo nombres de Ocurrencia de tablas, donde se haga referencia a éstos.

**Convención de mayúsculas y minúsculas recomendada:** La selección de una convención para utilizarla en Nombres de tablas debe realizarse en función de la forma en que se asignarán nombres a las Ocurrencias de tablas y Presentaciones. La decisión respecto del uso o no de UpperCamelCase (mayúsculas y CamelCase), lowerCamelCase (minúsculas y CamelCase) o MAYÚSCULAS es en la mayoría de los casos simplemente una preferencia. Sin embargo, el desarrollador debe analizar algunos factores. Primero, ¿se desea incluir el Nombre de Tabla en el nombre de Ocurrencia de tabla? Si es así, es necesario analizar la convención que se desea utilizar



para los nombres de las Ocurrencias de tabla. En el supuesto de que se desee utilizar todas MAYÚSCULAS, deberá determinarse la forma en que se separarán las palabras (espacio, subrayado, subrayado doble, etc.) Por otra parte, si se utilizan subrayados para separar las palabras, qué se utilizará para separar la sintaxis que será asignada a las Ocurrencias de tablas, Presentaciones y otras áreas donde sea necesario incluir el Nombre de tabla. Es importante que las elecciones para cada sección se realicen en función del lugar y la forma en que se hará referencia al Nombre de tabla en otras partes.

**Cuestiones de longitud del nombre de tabla:** Un Nombre de tabla puede tener hasta 100 caracteres. Sin embargo, en muchos casos éste no resulta un límite práctico. Refiriéndonos nuevamente a la Convención de mayúsculas y minúsculas recomendada, el Nombre de tabla se podría incluir en el Nombre de Ocurrencia de Tabla, de modo que esta referencia al Nombre de la Tabla de origen que se hace en el nombre de la Ocurrencia de tabla agregaría longitud al nombre de la Ocurrencia de tabla. Además, toda convención adicional utilizada en la Ocurrencia de Tabla puede crear un nombre de Ocurrencia de tabla de mayor longitud que lo que mostrarán algunos cuadros de diálogo de interfaces sin recurrir a alguna manipulación. Presentamos esta cuestión para enfatizar que debemos pensar un poco en la longitud de los nombres de nuestras tablas en función de las otras áreas en que se aplicarán las convenciones.

**Pautas para singular vs. plural:** La decisión de utilizar nombres en singular o colectivos en lugar de la alternativa en plural se incluye sólo con el fin de sugerir al desarrollador que adopte una decisión respecto de cuál utilizará y que sea coherente con dicha decisión.

**Categorización de tabla:** A algunos desarrolladores les gusta incluir metadatos en el nombre de tabla. En la mayoría de los casos esto da a la tabla una categoría que permite alojar un tipo específico de datos o indicar que realiza una función específica en la estructura de la solución. Algunos ejemplos son tablas construidas para sesiones, referencias, utilitarios y uniones. Dentro de FileMaker no existe la capacidad para denotar o comentar este tipo de categorización. Por ello, varios desarrolladores incluyen estos metadatos como parte del nombre de tabla.

**Tablas separadoras entre categorías de tablas (Spacing tables):** En la comunidad FileMaker existe una práctica que consiste en utilizar una técnica que crea tablas separadoras entre categorías de tablas; estas tablas no contienen campos sino que se utilizan para actuar como separadores entre categorías de tablas. Las soluciones menores no se verán beneficiadas con esta técnica, pero dado que las soluciones crecen e incluyen mayores cantidades de tablas, esto puede ser de suma utilidad.

## 2.3 Pautas “en cumplimiento de las normas” (Tablas):

1. Sólo deben contener los siguientes caracteres:
  - aA - zZ mayúsculas y minúsculas
  - Números 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple o doble “\_” “\_\_”
2. NO deben contener espacios.
3. Debe utilizar en forma coherente nombres de tablas en singular o en plural.
4. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

### Métodos de separación de palabras recomendados (Nombres de tablas) :

minúsculas y CamelCase	nombreDeMiTabla
mayúsculas y CamelCase	NombreDeMiTabla
Subrayado simple (minúsculas)	nombre_de_mi_tabla
Subrayado simple (Tipo Titulo)	Nombre_De_Mi_Tabla
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_TABLA

5. NO se debe utilizar ninguna de las Palabras Reservadas de FileMaker
6. NO se debe utilizar ninguna de las Palabras Reservadas de SQL Ver el Apéndice A. Verifique además la documentación del RDBMS.



7. Categorización de tabla: Como se menciona en la definición del problema, algunos desarrolladores prefieren indicar el 'tipo' de tabla o la función de ésta dentro del nombre de la tabla en sí. Si la práctica de desarrollo hace necesario incluir esto, recomendamos la siguiente sintaxis:
8. Sintaxis: {función}[\_\_]TableName  
Ver en el punto [Leyenda de sintaxis](#) (Apéndice D) la descripción de ésta.
  - función – opcional; definida por el desarrollador  
Le ofrece al desarrollador una categorización definida que puede ampliar en la forma necesaria. El desarrollador, como tal, puede definir las diferentes categorías. Tendrá en mente si usará mayúsculas y/o minúsculas, longitud y coherencia cuando la defina. Será necesario documentar las elecciones siguiendo las pautas de la Sección de [Adhesión](#) en este documento.
  - “\_\_” – obligatorio  
Los subrayados dobles deben utilizarse para indicar la separación de la función el Nombre de la tabla.
  - Nombre descriptivo  
El nombre descriptivo para la tabla.

## 2.4 Consideraciones complementarias:

**Uso de Tablas separadoras entre categorías de tablas (Spacing Tables):** En el caso de soluciones con gran cantidad de tablas, puede ser útil agregar cierta categorización a la lista de nombres de tablas. Dado que el entorno de desarrollo no aporta esta capacidad depende del desarrollador imaginar una forma para lograrlo. En el siguiente ejemplo tenemos una lista de tablas relativamente corta, sin embargo, servirá para nuestro ejemplo: Si se considera una lista más larga, el valor de esta función se hace más evidente. Se han creado tablas con alguna indicación de que actúan como cabeceras para una categoría. El ejemplo muestra Contabilidad, Comunicaciones y ‘-Registro-’. Se trata simplemente de tablas que no contienen campos. Es posible que se desee quitarlas del Gráfico de Relaciones, dado que nunca se utilizarán en la solución propuesta. Pero, como es de imaginarse, podría hacer el trabajo un poco más organizado en los supuestos de soluciones más grandes. En nuestro ejemplo se utiliza un signo “-” delante y detrás. El desarrollador puede utilizar aquella forma que le sirva. El punto es utilizar algo que se destaque. Se tendrá en cuenta que FileMaker emitirá una advertencia acerca de determinados caracteres, y que no se utilizarán esas tablas en forma alguna.

Nombre de tabla
◆ -----
◆ - Contabilidad -
◆ Clientes
◆ Matriculas
◆ Facturas
◆ LineaDeFactura
◆ Productos
◆ - Comunicaciones -
◆ Eventos
◆ Principales
◆ Reservas
◆ - Inscripción -
◆ Asistencia
◆ Empleados
◆ Estudiantes

Figura 1



## 3 ▷ Asignación de nombres de campos

Los nombres de campo constituyen un área en la cual el tema de las convenciones es sumamente variado. Generalmente, existen dos tendencias. Aquellos desarrolladores que deseen usar nombres “naturales” y los que utilicen alguna forma de notación para exponer algunos metadatos respecto del campo. Esta notación se utiliza para exponer información acerca del campo, la cual no se encuentra disponible aparte del cuadro de diálogo de Definir Base de Datos (Define Database). Existen muchas variantes del detalle que expone una notación. Pero la idea subyacente es la misma en todos los casos.

Al seleccionar nombres de campos existen varios factores que deben tenerse en cuenta. Como con todas las áreas, debe seleccionarse un conjunto de caracteres que no obstaculice la conectividad a las soluciones desde tecnologías tales como ODBC, JDBC y XML. Por razones de coherencia y profesionalismo, deberá elegirse entre mayúsculas, minúsculas y otros tipos, y ser coherente con la elección. Deberá seleccionarse la forma en que se separarán las palabras individuales dentro del nombre del campo. Una vez más, el objetivo es lograr coherencia y profesionalismo.

Debe tenerse presente cómo se separará sintácticamente el nombre del campo en sí de todo metadato, prefijo o sufijo. La separación sintáctica para nombres de campo se torna muy importante debido a la variedad de problemas que se intenta enfrentar. Es necesario tener en cuenta la categorización del campo, indicando campos clave/de coincidencia, campos utilitarios o campos del desarrollador. El objetivo es poder distinguir lo que es notación de aquello que no lo es, sin necesidad de descifrar códigos. Dicho esto, si se decide utilizar una notación es necesario que exista una comprensión universal de ésta. Por lo tanto, las FDC ofrecen dos componentes básicos que ayudan en este aspecto. En primer lugar, la convención ofrece una sintaxis común que indica el ‘lugar’ en el cual colocar las partes. Los componentes más comunes deben anotarse e indicarse. Y por último, dentro de la sintaxis ofrecer la capacidad de ‘extenderla’ o personalizarla de acuerdo con los requerimientos más específicos del desarrollador.

Se tendrá en cuenta dónde y porqué los componentes de la sintaxis se colocan dentro de un orden de recomendación específico. Se utilizarán nombres en singular o plural y su uso será coherente. En el caso de nombres de campos es importante buscar formas de agrupar los campos, como por ejemplo utilizando ‘NameFirst’, ‘NameLast’ opuestos a ‘FirstName’ y ‘LastName’. Uno de los problemas analizados era utilizar el tipo de orden por nombre de campo más que el tipo de orden personalizado. Aunque no se trata de una decisión básica en la forma de asignar nombre a los campos, utilizar esta capacidad incorporada puede reducir en tiempo invertido en actividades de organización. Debe analizarse la forma en que el uso del ‘espacio en blanco’ puede ayudar al desarrollador a agregar una determinada organización a las listas de campos. Esto es muy útil cuando se trata de una gran cantidad de campos, especialmente si se prefiere un método de clasificación manual para las listas de campos.

Las FDC se han elaborado comprendiendo que las prácticas de desarrollo pueden no ajustarse a lo ya descrito. En este punto realmente se trata de una elección. Si se utilizará una forma de notación, esta recomendación debe considerarse en profundidad.

### 3.1 Objetivos:

- ✓ Conjunto de caracteres recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada
- ✓ Cuestiones de longitud del nombre de campo:
- ✓ Opciones de asignación de nombre a los campos por falta de metadatos de campo en el entorno de desarrollo
- ✓ Limitaciones que impiden el ‘ocultamiento’ de los campos utilitarios de la interfaz del usuario que enfrentan la limitación del entorno de desarrollo
- ✓ Separación de campo y metadatos
- ✓ Opciones para uso de espacios en blanco
- ✓ Opciones para agrupamiento y legibilidad de los campos



## 3.2 Definición de problemas:

**Conjunto de caracteres recomendado:** El uso de caracteres inadecuados puede acarrear dificultades en diversas áreas. El uso de caracteres válidos dentro de los cálculos es muy importante. Los desarrolladores deben tener cuidado de utilizar caracteres que no entren en conflicto con los cálculos de FileMaker, de evitar problemas con las variables, y deben ser conscientes de todo carácter que pueda presentar problemas en el intercambio de datos con Sistemas de Administración de Bases de Datos Relacionales (RDBMS).

**Convención de mayúsculas/minúsculas recomendada:** La decisión de cómo separar las palabras dentro de los nombres de campos incidirá en la coherencia de conjunto en todas las convenciones. Por ejemplo, decidir que un signo “\_” (subrayado) separe las palabras individuales dificultará la diferenciación del nombre del campo de cualquier prefijo o sufijo que se haya elegido para la representación de metadatos del campo. Uno de los usos más comunes de esto es el uso de “g\_” para preceder el nombre del campo.

**Longitud del nombre de campo:** Un campo FileMaker puede tener hasta 100 caracteres. Sin embargo, en muchos casos éste no resulta un límite práctico. Con el lanzamiento de la versión 8, la mayoría de los cuadros de diálogo no presentarán dificultad alguna para exhibir el nombre completo. No obstante, las peticiones Ordenar (Sort), Exportar (Export) y Edición Buscar (Edit Find) son limitadas. Ordenar (Sort) se limita a 20 caracteres en OS X y a 26 caracteres en Windows. Exportar (Export) se limita a 30 caracteres en OS X y a 28 caracteres en Windows. Las peticiones Edición Buscar (Edit Find) se limitan a 26 caracteres en OS X y a 27 caracteres en Windows. Estas cantidades varían según la anchura de los caracteres. Por ejemplo “8” es más ancho que “1”. Las cantidades mencionadas se basan en una cadena de caracteres de anchura completa. Las FDC no contienen recomendaciones acerca de la longitud pero sí estos límites como referencia.

**Falta de metadatos de campo dentro del entorno de desarrollo:** El campo es uno de los principales objetos programables con los cuales trabaja el desarrollador. Dejando de lado una memoria notable, es difícil, si no imposible, conocer los metadatos de un campo sin abrir Definir Base de Datos (Define Database). ¿Cuál es su tipo de datos? ¿Qué depende de él? ¿Cómo almacena los datos? ¿Cuál será su comportamiento en un guión? No se dispone rápidamente de esta información. Ya sea que se esté de acuerdo con que éste es un desafío o no, sí determina la forma en que se le aplicará una convención de nombres. En el supuesto de que se desee exponer esta información a nivel del nombre del campo, las consideraciones respecto de la forma de lograrlo deben ser parte de la convención.

**Imposibilidad para ocultar campos auxiliares:** Los campos “auxiliares” son campos definidos para almacenar datos o información transitoria que generalmente no se expone al usuario final. En todo caso, el reto para el desarrollador es que no puede impedir que los usuarios finales vean estos campos en todos los campos que presentan cuadros de diálogo. Mientras FileMaker 8 ofrece la funcionalidad de enumerar sólo los campos en una presentación para exportarlos y clasificarlos en cierta forma, existen muchas otras áreas donde se enumerarán todos los campos. En este caso la imposibilidad de ocultar los campos deja a criterio del desarrollador la posibilidad de enumerar estos campos en forma poco común. En la mayoría de los casos, esto termina siendo la parte inferior de la lista de campos. A partir de la versión 7, el FileMaker utiliza el tipo de orden Unicode para los nombres de los campos. Diferente de clasificar manualmente los nombres de los campos, lo cual puede ser molesto y lento, la única opción es establecer una convención de nombres, que mueva los campos utilitarios a la parte inferior de la lista. Debido al tipo de clasificación Unicode, el carácter “z”, “zz”, o un carácter Unicode que ordena estos campos en la parte inferior generalmente se acuerda como la mejor forma de asegurar que un campo se liste el último. El desarrollador debe decidir si esto es importante o no para él.

**Sintaxis de separación de campo y metadatos:** Aquí es necesario un claro entendimiento entre los desarrolladores del momento y del futuro así como también el usuario final acerca de qué separa el nombre del campo de los metadatos / notación. Además, es necesario que haya una designación clara y coherente entre lo que es notación y lo que es nombre de campo.

## 3.3 Pautas “en cumplimiento de las normas” (Campos):

En las FDC se ha decidido reconocer tres (3) amplias categorías para los campos. Son los Campos Generales, Campos Clave y Campos Auxiliares.



### 3.3.1 Campos generales

Un Campo general está categorizado como esos campos que se utilizan para almacenamiento de datos en general.

1. Sólo deben contener los siguientes caracteres:
  - Mayúsculas y minúsculas: a–z o A–Z
  - Números: 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble " \_ " " \_\_ "
2. NO deben contener espacios.
3. NO deben comenzar con números.
4. Se debe ser coherente con el uso de nombres en singular o plural.
5. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Nombre de campo) :

Minúsculas y CamelCase	nombreDeMiCampo
Mayúsculas y CamelCase	NombreDeMiCampo
Subrayado simple (minúsculas)	nombre_de_mi_campo
Subrayado simple (Tipo Título)	Nombre_de_Mi_Campo
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_CAMPO

6. Se debe ordenar la lista de campos por nombre de campo
7. Según el carácter o caracteres que se utilice en el indicador de "Campo Utilitario", se debe reservar este carácter, o sea no se debe iniciar ningún campo con esa letra. En el área de campos utilitarios la letra "z" y "zz" así como también otros caracteres Unicode colocarán los campos utilitarios en la parte inferior de la lista de campos, en el supuesto de que los campos se ordenen por nombre.
8. Se debe utilizar el Agrupamiento de archivos cuando sea posible.  
Los ejemplos incluyen:
  - Nombre
  - Nombre Apellido
  - DireccionCiudad (AddressCity)
  - DireccionEstado (AddressState)
  - DireccionCodigoPostal (AddressPostalCode)
  - PasaporteNumero (PassportNumber)
  - PasaportePaisEmisor (PassportIssuingCountry)
  - PasaporteFechaDeExpiracion (passportExpirationDate)
  - PasaporteNombre (passportName)
9. Los Campos derivados o calculados que el desarrollador determina que no existen como campos "Auxiliares" pero en los que desea incluir la notación, deben seguir las recomendaciones para los campos "Auxiliares" como un sufijo.

Por ejemplo, el Total de la Factura se calcula, sin embargo, no es un campo que se desee "ocultar" al usuario. Sin embargo, el desarrollador tal vez desee incluir alguna notación. En tal caso, al campo se le da el nombre de TotaldeFactura\_\_1cn. Esto indica que el campo se almacena localmente, se calcula, y que el resultado es tipo cifra. Ver Asignación de nombres a utilitarios si se necesita información adicional.



### 3.3.2 Campos clave/de coincidencia

FileMaker no impone ni ordena que se utilicen muchas de las restricciones de la integridad relacional presentes en los Sistemas de Administración de Bases de Datos Relacionales basados en SQL. Por ejemplo, no ordena que se cree una clave principal para una tabla. No existe indicación alguna, visual o contextual dentro de la aplicación, de que se trata de un campo clave/de coincidencia. En la mayoría de los casos es aconsejable o necesario utilizar campos clave o de coincidencia dentro de las soluciones. Se puede utilizar alguna notación para aplicarla específicamente a la identificación de los campos clave.

Si se decide utilizar la notación para campos clave, la siguiente sintaxis es la recomendada en el criterio de cumplimiento de las normas.

Directivas para la sintaxis de campos clave/de coincidencia:

- Proporcionar un mapa universalmente disponible, comprensible y coherente para la notación de metadatos clave:
    - Identificar campos clave/de coincidencia
    - Identificar la función de los campos clave/de coincidencia
    - Identificar el almacenamiento de los campos clave/de coincidencia
    - Identificar el tipo de datos de los campos clave/de coincidencia
  - Ordenar los Campos clave/de coincidencia en la parte superior de las listas de campos mientras se mantiene un tipo de orden de campos por nombre (Tipo de orden alfabético). Esto reduce la clasificación manual por parte de los desarrolladores y presenta el orden esperado por los usuarios de la solución.
  - Utilizar caracteres de notación más conocidos (populares), como por ejemplo "p" por clave principal (primary key) y "f" por clave externa (foreign key), o bien aquellos que puedan alinearse con el entorno de desarrollo del FileMaker.
1. Sólo deben contener los siguientes caracteres:
    - Mayúsculas y minúsculas: a–z o A–Z
    - Numérico: 0,1,2,3,4,5,6,7,8,9
    - Subrayado simple y doble " \_ " " \_\_ "
  2. NO deben contener espacios.
  3. NO deben comenzar con números.
  4. Se debe ser coherente con el uso de nombres en singular o plural.
  5. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Conjuntos de privilegios) :

Minúsculas y CamelCase	nombreDeMiCampo
Mayúsculas y CamelCase	NombreDeMiCampo
Subrayado simple (minúsculas)	nombre_de_mi_campo
Subrayado simple (Tipo Título)	Nombre_de_Mi_Campo
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_CAMPO

6. Sintaxis: [k]<función>(almacenamiento)(tipo)[\_\_]Nombre\_Descriptivo  
Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.
  - "\_\_" – obligatorio  
El subrayado simple colocará claves en la parte superior de todos los cuadros de diálogo en que los campos se enumeren y del gráfico de relaciones.
  - k – minúsculas, obligatorio  
Denota el campo como campo clave/de coincidencia



- función – minúsculas; obligatorio, usa los valores provistos, si se utilizan  
Denota la categoría del campo clave y/o la función que desempeñará.
  - ▶ p – Clave principal (Primary Key)
  - ▶ f – Clave externa (Foreign Key)
  - ▶ a – Clave alternativa (Alternate Key)
  - ▶ c – Clave compuesta / concatenada / calculada (Compound / Concatenated / Calculated Key)
  - ▶ m – Clave de líneas múltiples (Multi-Line Key)
- almacenamiento – minúsculas; opcional, usa los valores provistos, si se utilizan
- Denota el almacenamiento del campo
  - ▶ l = almacenado localmente ("L" minúscula)
  - ▶ g = almacenado globalmente
- tipo – minúsculas; opcional, usa los valores provistos, si se utilizan
- Denota el tipo de campo
  - ▶ t = Texto (Text)
  - ▶ n = Número (Number)
  - ▶ d = Fecha (Date)
  - ▶ i = Hora (Time)
  - ▶ m = Fecha y hora (Time Stamp)
- \_\_ – obligatorio  
El subrayado doble debe utilizarse para separar la sintaxis de la clave del Nombre Descriptivo para el campo.
- Nombre Descriptivo – Mayúscula o minúscula elegida  
Nombre descriptivo del campo clave
- Ejemplos:

#### Formato completo utilizando todos los metadatos opcionales

```
_kplt__IdDeFactura
_kcgt__ParticipantesSelectos
_kflt__IdDeCliente
_kmgt__Vistadediaseleccionados
_kflt__Elemento_de_linea_de_factura
```

#### Formato mínimo sin utilizar los metadatos opcionales

```
_kp_IdDeFactura
_kc_ParticipantesSelectos
_kf_IdDeCliente
_km_Vistadediaseleccionados
_kf_Elemento_de_linea_de_factura
```

### 3.3.3 Campos auxiliares

La asignación de nombres a los campos utilitarios está diseñada para crear una forma coherente que se ajuste a las diversas necesidades de los desarrolladores y que enfrente los problemas más comunes de los nombres de campo para esos campos que generalmente no se desea poner a disposición del usuario final.

Directivas para campos utilitarios:

- Proporcionar un mapa universalmente disponible, comprensible y coherente para la notación de metadatos en campos auxiliares:
- Identificación del campo auxiliar
- Identificación de la función de un campo auxiliar que se puede personalizar
- Identifica el almacenamiento del campo auxiliar



- Identifica el tipo de datos del campo auxiliar
  - Identifica el campo auxiliar como un campo repetitivo
  - Ordena los campos auxiliares en la parte inferior de la lista de campos mientras mantiene un ordenamiento de los campos por nombre (Tipo de orden alfabético) Esto reduce la clasificación manual por parte de los desarrolladores y coloca estos campos de forma tal de no obstruir la acción del usuario.
1. Sólo deben contener los siguientes caracteres:
    - Mayúsculas y minúsculas: a–z / A–Z
    - Numérico: 0,1,2,3,4,5,6,7,8,9
    - Subrayado simple y doble “\_” “\_\_”
  2. NO deben contener espacios.
  3. NO deben comenzar con números.
  4. Se debe ser coherente con el uso de nombres en singular o plural.
  5. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Nombres de campo) :

Minúsculas y CamelCase	nombreDeMiCampo
Mayúsculas y CamelCase	NombreDeMiCampo
Subrayado simple (minúsculas)	nombre_de_mi_campo
Subrayado simple (Tipo Titulo)	Nombre_de_Mi_Campo
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_CAMPO

6. Sintaxis: <zz>{función}[\_\_]Nombre\_Descriptivo[\_\_(almacenamiento)(tipo)(repeticion)]  
Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.
  - “zz” o carácter Unicode que ordena y coloca en la parte inferior – obligatorio  
Indica el campo como una clave del desarrollador y lo ordena en la parte inferior de todos los cuadros de diálogo en que los nombres de los campos están presentes.
  - función – minúsculas; opcional; definida por el desarrollador  
Ofrece una categorización definida por el desarrollador que se puede ampliar en la forma necesaria. El desarrollador como tal puede definir las diferentes categorías. Debe considerarse si se usarán mayúsculas y/o minúsculas, longitud y coherencia cuando se definan. Las elecciones deben ser documentadas según las pautas de la Sección de [Adhesión](#) de este documento.
  - “\_\_” – obligatorio  
El subrayado doble denota la separación del prefijo del Nombre descriptivo.
  - Nombre Descriptivo  
El desarrollador elige el nombre deseado. Sigue las pautas generales respecto de los campos.
  - “\_\_” – obligatorio  
El doble subrayado se utilizará para indicar el final del Nombre Descriptivo y el inicio de la notación del sufijo.
  - almacenamiento – minúsculas; obligatorio, usa los valores provistos, si se utilizan  
Se utiliza para denotar el almacenamiento del campo
    - ▶ l = almacenado localmente (“l” minúscula)
    - ▶ g = almacenado globalmente
  - tipo – minúsculas; obligatorio, usa los valores provistos, si se utilizan  
Indica el tipo de datos utilizados o devueltos.



- Tipos de resultados no calculados
  - ▶ xt = Texto (Text)
  - ▶ xn = Número (Number)
  - ▶ xd = Fecha (Date)
  - ▶ xi = Hora (Time)
  - ▶ xm = Fecha y hora (Time Stamp)
  - ▶ xr = Contenedor (Container)

Tipos de resultados calculados

- ▶ ct = Texto (Text)
- ▶ cn = Número (Number)
- ▶ cd = Fecha (Date)
- ▶ ci = Hora (Time)
- ▶ cm = Fecha y hora (Time Stamp)
- ▶ cr = Contenedor (Container)

Tipo de resultado sumario

- ▶ xs = Resumen Sumario)

repetición – minúsculas; obligatorio, usa los valores provistos, si se utilizan  
Indica si los datos se almacenan con repeticiones

- ▶ p = repeticiones (repetitions)

### 3.3.4 Referencia a la notación del campo

#### Referencia a la notación del campo

	Texto	Número	Fecha	Hora	Sello fechador	Contenedor
--	-------	--------	-------	------	-------------------	------------

#### Claves (formato largo: Utilizando todos los metadatos)

Clave: principal	_kplt	_kpln	_kpld	_kpli	_kplm	N/A
Clave: externa	_kflt	_kfln	_kfld	_kfli	_kflm	N/A
Clave: alternativa	_kalt	_kaln	_kald	_kali	_kalm	N/A
Clave: Compuesta	_kclt	_kcln	_kcld	_kclic	_kclm	N/A
Clave: Multilínea	_kmlt	_kmln	_kml d	_kml i	_kml m	N/A
Repeticiones	p	p	p	p	p	N/A

#### Campos auxiliares / Campos generales y calculados (Formato completo: Utilizando todos los metadatos)

General (Almacenamiento local)	__lxt	__lxn	__lxd	__lxi	__lxm	__lxr
General (Almacenamiento global)	__gxt	__gxn	__gxd	__gxi	__gxm	__gxr
Calculada (Almacenamiento local)	__lct	__lcn	__lcd	__lci	__lcm	__lcr
Calculada (Almacenamiento global)	__gct	__gcn	__gcd	__gci	__gcm	__gcr
Repetitiva (Agregar "p" al final de la notación, en minúsculas)	p	p	p	p	p	p
Sumario (Almacenamiento local)	N/A	__lxs	N/A	N/A	N/A	N/A



### 3.4 Consideraciones complementarias:

**Separadores de campos:** Muchos desarrolladores utilizan un campo 'no funcional' para que actúe como separador en la lista de campos. Los espacios en las listas largas de elementos hacen que estos últimos sean más manejables. Permite cierta clase de división, donde el entorno de desarrollo no tenga esta posibilidad. Cuando se usa esta técnica se recomienda utilizar un tipo de campo de número y configurarlo para almacenamiento global para reducir el tamaño del almacenamiento.



## 4 ▷ Ocurrencias de tablas

A partir del FileMaker 7 y de la Ocurrencia del Gráfico de Relaciones, el producto ha ampliado sus posibilidades para admitir en mejor forma sistemas de gran envergadura y complejidad. No obstante, el Gráfico de Relaciones puede ser molesto y pesado cuando se representan aplicaciones complejas. Por ejemplo, es posible que un sistema complejo tenga la misma tabla de origen representada múltiples veces en el gráfico. Al unir esto con el requerimiento de que puede haber no más de una ruta relacional entre dos Ocurrencias de Tablas cualesquiera. El gráfico puede resultar largo y difícil de navegar. Aunque el Gráfico de Relaciones ofrece al desarrollador una Interfaz gráfica de usuario (GUI) para trabajar con las relaciones, no ofrece la posibilidad de agrupar o asignar nombres en forma lógica a los grupos de Ocurrencias de tablas. De este modo es difícil saber con qué Ocurrencia de tabla se debe trabajar al seleccionarla de una lista desplegable. Todas las Ocurrencias de Tablas se enumeran en orden alfabético en la lista y carecen de agrupamiento lógico. Existen numerosos lugares en que es necesario elegir una Ocurrencia de Tabla como por ejemplo al colocar campos en una presentación o seleccionar un campo en un cuadro de diálogo de cálculo. En estos casos, no hay posibilidad de ver la tabla de origen subyacente de la Ocurrencia de la Tabla o su contexto en el Gráfico de Relaciones. Esta limitación impulsa la búsqueda de una convención de nombres que incrusten el origen y la descripción contextual en el nombre de la Ocurrencia de la Tabla. Las convenciones de nombres propuestas tienen el objeto de indicar origen y contexto.

Es obvio que todo intento de definir una convención de nombres estará estrechamente ligado a la metodología individual del desarrollador para representar Ocurrencias de Tablas. Por ello, se ha decidido que las FDC ilustren varios métodos de construcción de Gráficos de Relaciones y las convenciones de nombres que los respaldan, como recurso para resolver consultas sobre las soluciones en lugar de proponer una única convención que pueda favorecer un método de desarrollo en particular.

Por último, independientemente del método particular utilizado, debe ser posible que otro desarrollador, con un mínimo esfuerzo, comprenda en forma clara y coherente los puntos específicos de la solución. Impulsar un criterio generalizado que se pueda afinar posteriormente es un primer paso firme. Se encuentran muchas similitudes en cada método analizado. Todos comparten un formato similar. Están agrupados mediante un prefijo. El paso siguiente es el Nombre de tabla de origen. Por último existe un Nombre Descriptivo opcional en el cual el contexto adicional es ventajoso.

### 4.1 Objetivos:

- ✓ Terminología
  - Definir la tabla
  - Definir la Ocurrencia de la tabla
  - Definir la Tabla de origen / Tabla base
  - Definir el Grupo Funcional de Ocurrencia de Tablas (Functional Table Occurrence Group FTOG)
  - Definir la Ocurrencia principal de la tabla (Primary Table Occurrence PTO)
- ✓ Conjunto de caracteres recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada
- ✓ Conflictos de longitud en Ocurrencias de tablas
- ✓ Proporcionar diferentes mecanismos para enfrentar el problema del contexto del gráfico de relaciones al elegir una Ocurrencia de tablas mientras se trabaja con cálculos, se agregan campos a una presentación, o se define el contexto de la presentación.
- ✓ Ofrecer diferentes mecanismos para enfrentar la dificultad de agrupar Ocurrencias de tablas con el fin de exponerlas al entorno de desarrollo.
- ✓ Proporcionar un método para enfrentar la dependencia de una Ocurrencia principal de la tabla (Primary Table Occurrence) de cálculos que sea autocontenida.
- ✓ Brindar diferentes métodos para superar los obstáculos organizativos que presentan las Ocurrencias de Tablas.



- ✓ Cuestiones relacionadas con tecnologías de conectividad (XML, ODBC, JDBC), incluso palabras Reservadas de SQL.

## 4.2 Descripción de problemas:

**Definiciones:** A los fines del presente documento y de la comprensión general de los términos utilizados, se incluyen las siguientes definiciones.

- **Tabla:** Recopilación de datos que pertenecen a una entidad, por ejemplo clientes o precios de acciones. El archivo de base de datos contiene una o más tablas, las cuales consisten en campos y registros. Cuando se crea una tabla nueva, una representación visual de la tabla, o una Ocurrencia de la Tabla, aparece en el Gráfico de Relaciones. Se pueden especificar múltiples Ocurrencias (con nombres únicos) de la misma tabla a fin de trabajar con relaciones complejas en el gráfico.
- **Ocurrencia de tablas** Una Ocurrencia de la tabla se refiere a una instancia de una tabla en el Gráfico de relaciones. Se debe tener en cuenta que todas las interacciones con una tabla a través del ambiente de desarrollo se realizarán con Ocurrencias de Tablas. Esta es la única forma de 'dirigirse' a una tabla y a su contenido.
- **Tabla de origen:** Una Ocurrencia de la tabla está relacionada con una Tabla. La Tabla de origen se refiere a la Tabla con la cual se relaciona la Ocurrencia de la Tabla. En el siguiente ejemplo, una Ocurrencia de Tabla denominada «InterfazDeAdmision\_Clases\_ListaDeClases» está relacionada con una Tabla de origen denominada 'Clases'. También se hará referencia a la Tabla de origen como Nombre de tabla de origen.

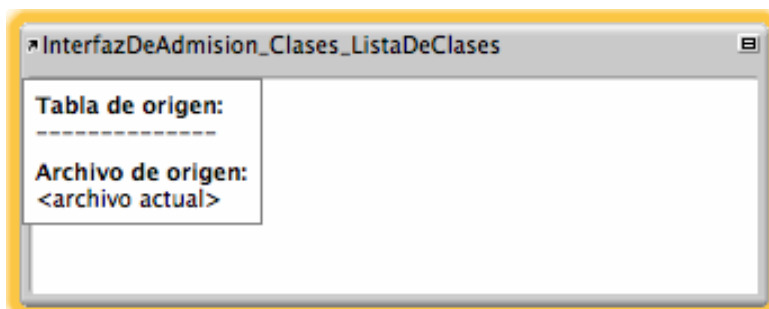


Figura 2

- **Ocurrencia principal de la tabla:** PTO es la sigla en inglés para una Ocurrencia de la tabla especial. La Ocurrencia de la tabla principal (PTO) sirve como la Ocurrencia de la tabla designada que se utilizará al crear los cálculos que constituyen 'referencia interna', esos cálculos que se derivan de datos contenidos únicamente en la misma tabla, que no utilizan datos relacionados.
- **Tabla separadora entre categorías de tablas (Spacing table):** Es el nombre de una técnica para crear tablas 'etiqueta' o 'separadora' dentro del cuadro de diálogo 'Define' (Definir). Esta técnica utiliza tablas que no contienen campos con el objeto de agrupar y categorizar tablas y Ocurrencias de tablas.

**Conjunto de caracteres recomendado:** Existen algunas restricciones básicas respecto de los caracteres que admitirá FileMaker. Por otra parte, algunos caracteres pueden provocar conflictos con la conectividad externa o con el intercambio de datos en algunos Sistemas de Administración de Bases de Datos Relacionales (RDBMS). El desarrollador debe estar al tanto de las restricciones tanto de FileMaker como de cualquier sistema externo con el cual interactuarán sus soluciones.

**Convención de mayúsculas/minúsculas recomendada:** La recomendación, en este caso, no consiste en superar un problema específico sino en aplicar un criterio coherente para adoptar la manera de dar nombres a las Ocurrencias de Tablas. El objetivo es seleccionar un método y ser coherente al respecto.

**Separación sintáctica recomendada:** La separación sintáctica para los nombres de Ocurrencias de Tablas se refiere a la forma en que se diferencia cualquier prefijo y/o sufijo (metadatos) del resto del nombre de la Ocurrencia de la Tabla. Es común observar prefijos o sufijos utilizados para indicar ciertos componentes del nombre. También se utilizan para contener características de identificación. El objetivo es aportar una forma coherente y universalmente reconocida de diferenciar estos elementos de metadatos principales/continuos.



Falta de contexto del gráfico de relaciones al elegir una Ocurrencia de tablas mientras se trabaja con cálculos, se agregan campos a una presentación, o se define el contexto de la presentación: Fuera del Gráfico de Relaciones, por ejemplo al elegir de una lista la Ocurrencia de la Tabla (Table Occurrence) (denominada "TO"), aparece una larga lista de nombres de TO que se pueden utilizar. Esta lista carece de agrupamiento u orden a menos que la convención de nombres utilizada aporte el contexto.

Dependencia de una Ocurrencia Principal de la Tabla de los cálculos que son autocontenidos: Al crear cálculos se debe elegir el contexto a partir del cual se evaluará el cálculo. En el caso en que un resultado de cálculo se derive sin hacer referencia a datos relacionados, se puede considerar que esos cálculos siempre se basarán en el mismo contexto.

Falta de posibilidades de organización de Ocurrencia de la tabla: Independientemente del método que se utilice para crear el Gráfico de relaciones, generalmente se acuerda que una cantidad de "TO" con frecuencia trabajan en conjunto para aportar alguna función, sin tener en cuenta la conectividad a otras "TO" en el gráfico. Sobre esta base, solo tiene sentido que se desee organizar este grupo de alguna forma. Esta posibilidad no existe a menos que la metodología y la convención de nombres utilizadas la ofrezcan.

### 4.3 Pautas "en cumplimiento de las normas" (Ocurrencias de Tablas):

1. Sólo deben utilizarse los siguientes caracteres:
  - Mayúsculas y minúsculas: aA o zZ
  - Números: 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble " \_ " " \_\_ "
2. NO deben contener espacios.
3. NO deben comenzar con números.
4. NO deben contener puntos.
5. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Ocurrencias de tablas) :

Minúsculas y CamelCase	nombreDeMiAparicionDeLaTabla
Mayúsculas y CamelCase	NombreDeMiAparicionDeLaTabla
Subrayado simple (minúsculas)	nombre_de_mi_aparicion_de_la_tabla
Subrayado simple (Tipo Título)	Nombre_De_Mi_Aparicion_de_la_Tabla
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_OCURENCIA_DE_LA_TABLA

#### 4.3.1. Método de Agrupamiento Funcional tipo Araña (FSG) Functional Spider Grouping

Este primer criterio se basa en la organización las Ocurrencias de tablas asignando a cada una un nombre con un prefijo que indica su función. La imagen del gráfico, una vez utilizado este método es más como una araña o tela de araña, y esta es la razón por la cual se utiliza el término araña para representar este enfoque. Obsérvese que este no es el único método para designar agrupamientos funcionales (ver el siguiente ejemplo de Grupos Funcionales de Ocurrencias de Tablas).

La principal idea respecto del agrupamiento funcional es que no se representan las Ocurrencias de tablas mediante el nombre o la relación sino mediante agrupamientos funcionales. Los tipos de funciones que pueden existir son ilimitadas y dependerán de las exigencias de la solución de que se trate y de la implementación de la misma. (guiones, listas de valores, portales, etc.).



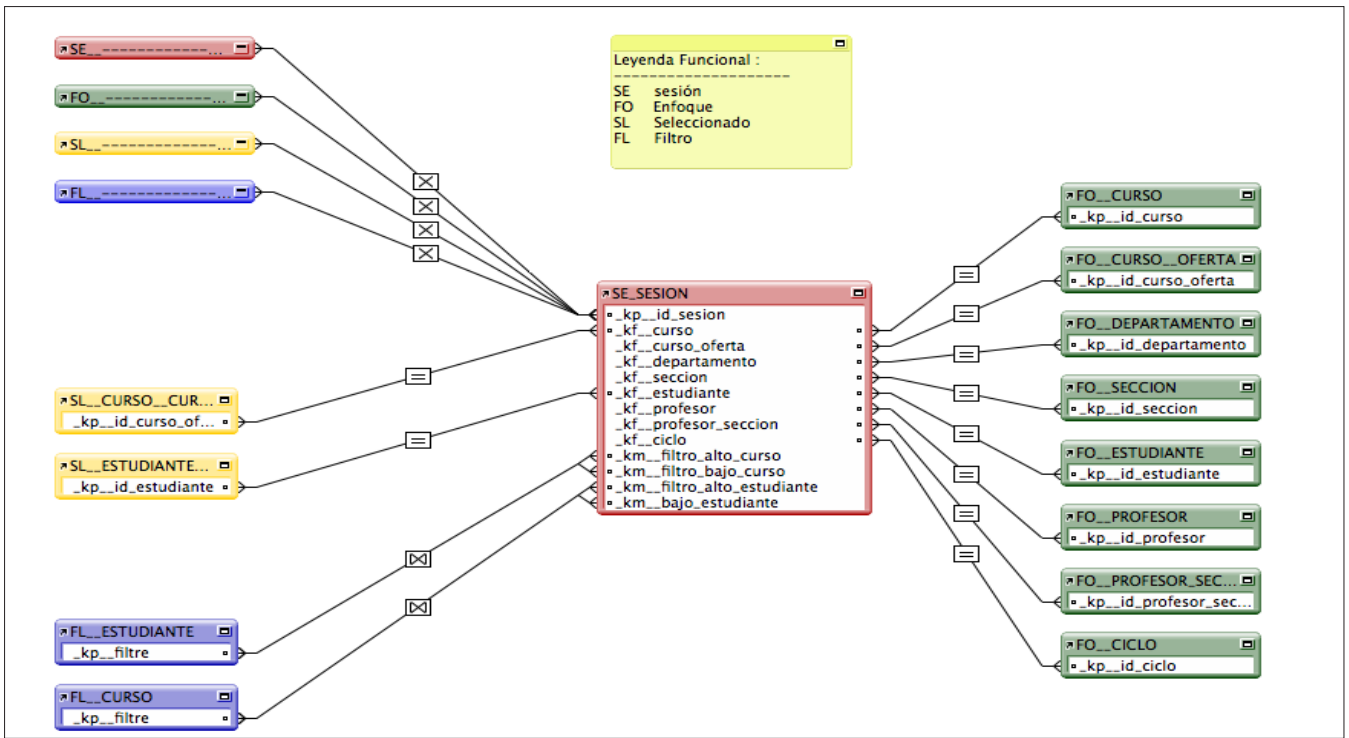


Figura 3

Con este método, los grupos funcionales se identifican mediante un “prefijo funcional”. En este caso se trata de un prefijo de dos letras. También se puede utilizar el color para ayudar a ubicar los diferentes grupos funcionales. Y la función de adjuntar notas al gráfico es nueva, del FileMaker 8. En este caso se agregó una nota para describir qué representa cada uno de los grupos funcionales de dos letras.

En este punto, el objetivo no es describir en mucho detalle cómo funciona este Grupo Funcional tipo Araña (Funcional Spider Group FSG) antes bien presentar los conceptos principales.

Uno de los grupos funcionales del gráfico está representado por el prefijo de dos letras con la etiqueta FO la cual será utilizada para representar la palabra «Enfoque». Todas las tablas del gráfico que tienen ese prefijo de dos letras serán agrupadas juntas cuando el usuario necesite ingresar información relacionada con el registro que tiene el enfoque.

Dado que habrá muchas tablas en el gráfico con este criterio, se utiliza una tabla separadora para proporcionar también una forma para que cada uno de estos agrupamientos funcionales se muestren todos juntos. Estas tablas separadoras también comparten el mismo prefijo de dos letras y cuando se conectan al FSG permiten la separación y agrupamiento para ayudar a localizar la función con la cual se desea trabajar.

Tablas relacionadas

FL_-----
FL_CURSO
FL_ESTUDIANTE
FO_-----
FO_CICLO
FO_CURSO
FO_CURSO_OFERTA
FO_DEPARTAMENTO
<b>FO_ESTUDIANTE</b>
FO_PROFESOR
FO_PROFESOR_SECCION
FO_SECCION
SE_-----
SL_-----
SL_CURSO_CURSO
SL_ESTUDIANTE_RESERVAS

Figura 4



La figura anterior es un ejemplo de selección de un campo en una presentación de sesión. Los separadores funcionales facilitan la ubicación de un grupo funcional particular en la lista desplegable de Ocurrencias de Tablas.

#### 4.3.1.1 VENTAJAS DEL AGRUPAMIENTO FUNCIONAL TIPO ARAÑA (*FUNCTIONAL SPIDER GROUPING* FSG)

- Reduce la cantidad de Ocurrencias de Tablas
- Provee organización por función
- Funciona bien en soluciones de pequeña–mediana envergadura. (Muchas soluciones se encuentran en esta categoría).
- Puede funcionar con soluciones de gran envergadura teniendo en cuenta que depende del número de necesidades funcionales. Cuantas más necesidades funcionales debe satisfacer el sistema, más atractivo se torna este método organizativo.
- Admite el modelo relacional bidireccional permitiendo que las presentaciones utilicen cualquier TOG (grupo de "TO" enlazado por relaciones) dentro del FSG. Esto reduce la cantidad de "TO" necesaria al no restringir la exposición de la presentación a sólo un "TO" en el FSG.
- Proporciona el agrupamiento funcional dentro del menú de Ocurrencia de tablas aparte del Gráfico de Relaciones con un orden agrupado en forma razonable.

#### 4.3.1.2 DESVENTAJAS DEL AGRUPAMIENTO FUNCIONAL TIPO ARAÑA (*FUNCTIONAL SPIDER GROUPING*)

- Es más atractivo para métodos "controlados por portales".. No funciona bien con soluciones más "abiertas".
- La asignación de nombres no tiene el objeto de exponer la estructura excepto en el gráfico.
- Es más difícil localizar las "TO" en el Gráfico de relaciones debido a la posibilidad de utilizar cualquier TO dentro del FSG como base de una presentación.

#### 4.3.1.3 AGRUPAMIENTO FUNCIONAL TIPO ARAÑA (*FUNCTIONAL SPIDER GROUPING*): PAUTAS "EN CUMPLIMIENTO DE LAS NORMAS"

1. Sintaxis: <<Prefijo funcional>>[\_]Nombre\_Descriptivo  
Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.
  - Prefijo funcional: obligatorio; definido por el desarrollador  
Provee el mecanismo para representar una separación lógica de Ocurrencias de tablas interconectadas. Podría tratarse de una abreviatura o nombre funcional.
  - Nombre Descriptivo:  
Provee el mecanismo para dar significado a la Ocurrencia de la Tabla.
  - " \_ " - obligatorio  
El uso de un subrayado doble " \_ " es el carácter separador recomendado.  
Esto permite el uso del subrayado dentro de los nombres de la Tabla de origen y de los Nombres lógicos de Ocurrencias de Tablas mientras incorpora posibilidades de legibilidad y análisis.

#### 4.3.2 Método de los Grupos Funcionales de Ocurrencia de Tablas (*Functional Table Occurrence Group* FTOG)

Otro criterio constituye el "Agrupamiento Funcional de Ocurrencia de Tablas" (FTOG) en el cual el Gráfico de relaciones tiene múltiples gráficos 'mini' funcionales, cada uno consiste sólo de Ocurrencias de Tablas que proporcionan la funcionalidad para dicho grupo o FTOG. Uno de los diferenciadores clave del método FTOG es que desglosa los diversos componentes o funciones de una solución en subconjuntos más pequeños. En algunos casos, esto da como resultado un gráfico más manejable. También reduce el número de Ocurrencias de tablas en las cuales efectuar búsquedas dentro del Gráfico de relaciones. Para cualquier función determinada sólo es necesario remitirse al grupo FTOG relacionado. Cada FTOG tendrá sólo aquellas Ocurrencias de Tablas correspondientes a la funcionalidad necesaria.



Con cualquier variante del método FTOG, uno de los objetivos es aportar un nombre significativo respecto del objeto de dicho grupo de Ocurrencias de tablas. El significado del agrupamiento puede variar pero lo importante es que el grupo aporta funcionalidad y que debe tener un nombre relacionado con esta funcionalidad de modo que se pueda comprender o al menos reconocer fuera del Gráfico de relaciones. La Figura 2 muestra un Gráfico de Relaciones organizado mediante el método FTOG. Cada uno de estos FTOG representa alguna funcionalidad que es independiente de los otros grupos FTOG. Mediante la herramienta de texto FileMaker 8 podemos colocar notas alrededor de cada Grupo Funcional para posibilitar una mejor comprensión del FTOG en el gráfico.

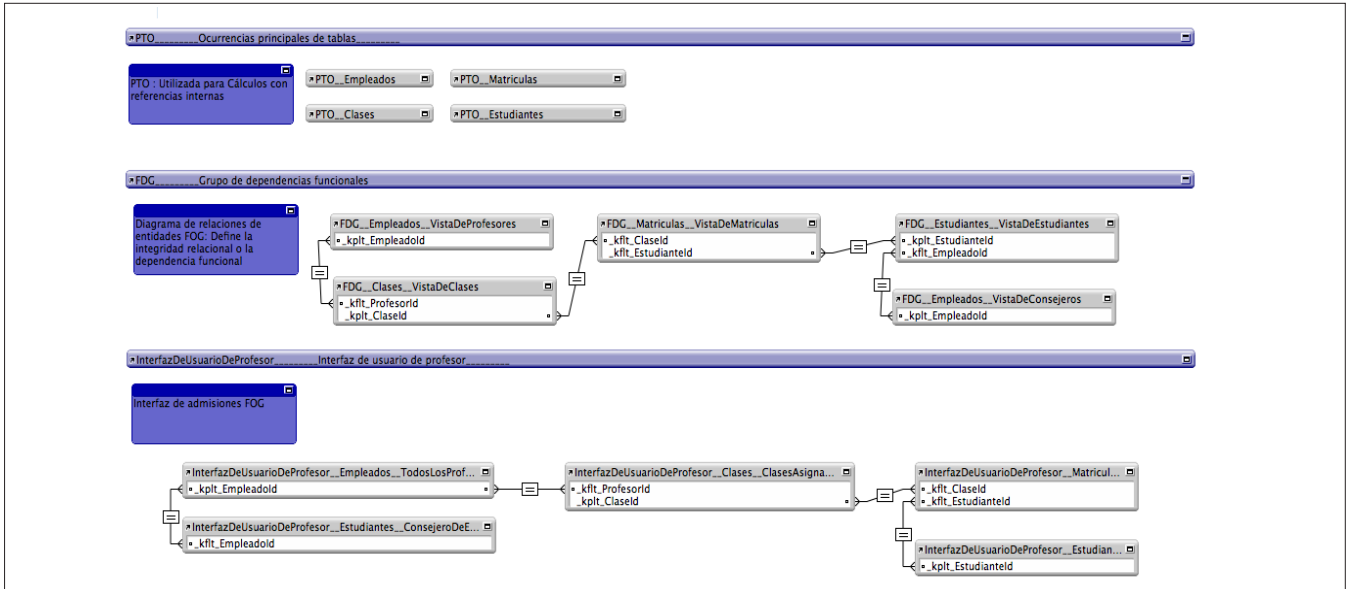


Figura 5

#### 4.3.2.1 VENTAJAS DEL AGRUPAMIENTO FUNCIONAL DE OCURRENCIA DE TABLAS (FTOG)

- Reduce la complejidad del Gráfico de relaciones
- Reduce la cantidad de Ocurrencias de Tablas
- Aporta la organización visual y de la lista por funcionalidad
- Funciona bien en soluciones de pequeña y mediana envergadura. (Muchas soluciones se encuentran en esta categoría).
- Puede funcionar con soluciones de gran envergadura teniendo en cuenta que depende del número de necesidades funcionales.
- Cuantas más necesidades funcionales, tanto menos atractiva la solución.
- Las tablas relacionadas dentro de un FTOG son bastante limitadas, y se agrupan en un tipo de orden alfabético dentro del FTOG.
- Admite el modelo relacional bidireccional permitiendo que las presentaciones utilicen cualquier TOG (grupo de Ocurrencias de Tablas (TO) enlazado por relaciones dentro del FTOG. Esto reduce la cantidad de TO necesaria al no restringir la exposición de la presentación a sólo una TO en el FTOG.
- Admite una amplia gama de metodologías de desarrollo: aquellas que se ejecutan bajo soluciones "controladas por portales" altamente encriptadas y soluciones con bases más abiertas que utilizan controles nativos FileMaker.
- Provee la agrupación funcional dentro del menú de Ocurrencia de tablas aparte del Gráfico de Relaciones con un orden agrupado en forma razonable.



#### 4.3.2.2 VENTAJAS DEL AGRUPAMIENTO FUNCIONAL DE OCURRENCIA DE TABLAS (FTOG)

- La asignación de nombres se considera más estricta.
- La asignación de nombres no tiene el objeto de exponer la estructura o el significado fuera del gráfico.
- Es más difícil localizar las "TO" específicas en el Gráfico de relaciones debido a la posibilidad de utilizar cualquier TO dentro del FTOG como base de una presentación.

#### 4.3.2.3 GRUPOS FUNCIONALES DE OCURRENCIA DE TABLAS: PAUTAS "EN CUMPLIMIENTO DE LAS NORMAS"

1. Sintaxis: <<FunctionalTableOccurrenceGroup>>[\_]<SourceTableName>[\_]DescriptiveName  
Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.

- FunctionalTableOccurrenceGroup (FTOG): obligatorio; definido por el desarrollador  
Proporciona el mecanismo para asignar el nombre a un grupo de Ocurrencias de tablas. Podría tratarse de una abreviatura o nombre funcional. Por ejemplo, se podría utilizar FD delante de un grupo de Ocurrencias de tablas que representa el grupo de Dependencia Funcional.  
'UserInterface' (Interfaz de usuario) puede ser otro ejemplo que indica el grupo de Ocurrencias de tablas que describe el grupo que provee la interfaz.
- " \_ " – obligatorio  
El uso de un subrayado doble " \_ " es el carácter recomendado. Esto permite el uso del subrayado dentro de los nombres de la Tabla de origen y de los Nombres lógicos de Ocurrencias de Tablas al tiempo que incorpora posibilidades de legibilidad y análisis.
- SourceTableName (Nombre de tabla de origen), obligatorio, utiliza el nombre de la tabla de origen.  
Incluye el Nombre de la tabla de origen en el nombre de la Ocurrencia de la tabla. Esto aporta un indicador visual y programático aparte del Gráfico de relaciones para la tabla subyacente.

Dado que la función Get (LayoutTableName) vuelve al nombre de la Ocurrencia de tablas y no al Nombre de la tabla de origen, y que no hay otra forma de obtener el nombre de la Tabla de origen. A fin de obtener programáticamente el Nombre de la tabla de origen, debe estar incluido en el Nombre de la Ocurrencia de tablas.

- " \_ " – obligatorio  
El uso de un subrayado doble " \_ " es el carácter recomendado. Esto admite el uso de subrayados dentro de los nombres de Tablas de origen y Nombres lógicos de Ocurrencias de tablas mientras aporta también posibilidades de legibilidad y análisis.
- Nombre Descriptivo  
Provee el mecanismo para dar significado a la Ocurrencia de la Tabla.  
Ejemplos:
  - ▶ FD\_Cliente\_\_TodosLosClientes
  - ▶ Interfaz\_\_Factura\_\_Facturasatrasadas
  - ▶ Interfaz\_\_Factura\_\_Facturasactuales
  - ▶ Sincronización\_\_Versión\_\_Versiones\_\_almacenadasen\_\_anfitrión

2. Separación del Grupo Funcional de Ocurrencia de Tablas: Ubicar una Ocurrencia de tablas dentro del cuadro de diálogo de cálculos (mientras se coloca un campo en una presentación o se lo asigna a una presentación) puede ser difícil. Se debe elegir de una lista desplegable que carece de categorización alguna. Por ejemplo, si se supone que una solución tiene 200 Ocurrencias de tablas, para realizar cualquiera de las tareas mencionadas anteriormente, se debe navegar a través de esta larga lista. Una convención de TOG ayudará en cierta medida al enumerar todos los TOG juntos, en orden alfabético. En el siguiente ejemplo solo hay 13 Ocurrencias de tablas, sin embargo, se debe considerar una lista mucho más larga.



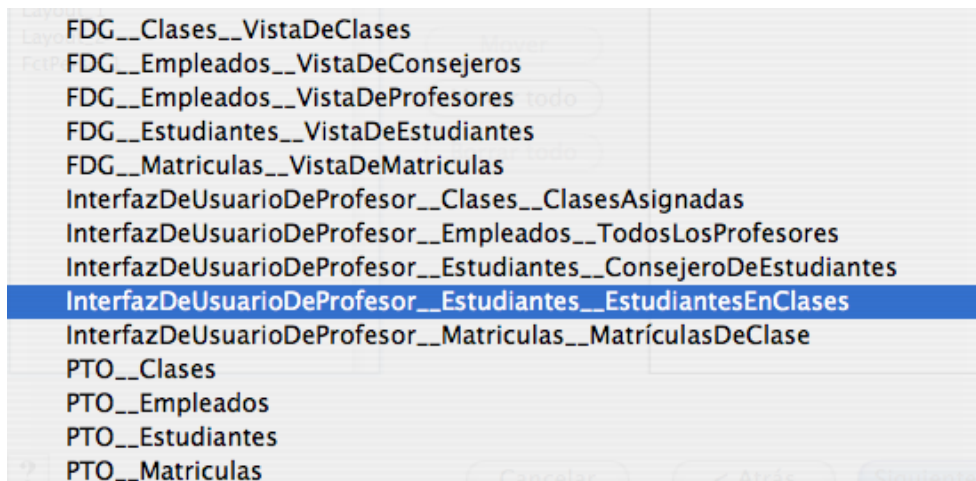


Figura 6

Lo siguiente muestra un ejemplo cuando cada FTOG tiene una cabecera que diferencia fácilmente cada TOG. En una larga lista de TOG esto puede ser muy útil.

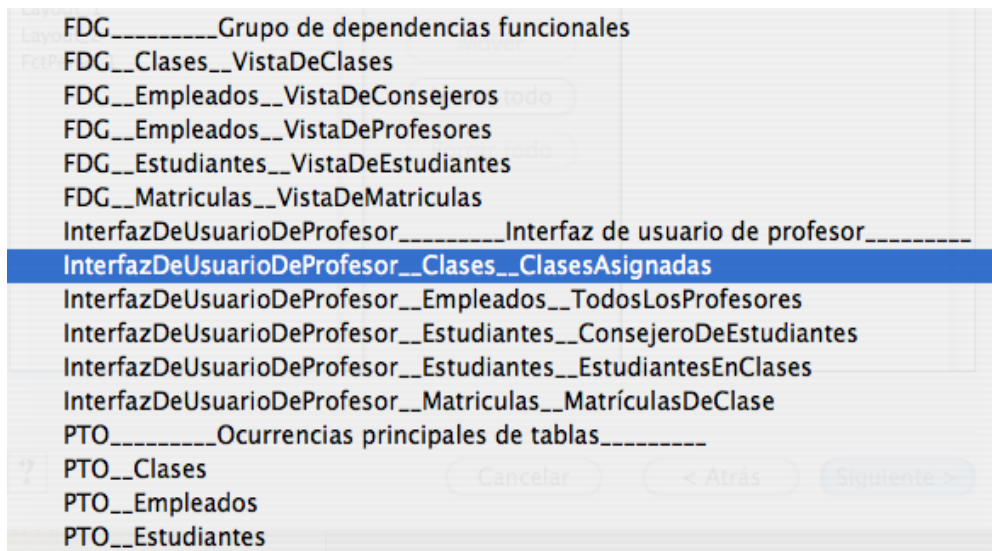


Figura 7

Este método exige que se cree una tabla separadora entre categorías de tablas o bien que se utilice una Ocurrencia de tablas ya existente como cabeceras para cada grupo funcional. En el ejemplo se ha creado una tabla denominada 10 guiones (-----). El nombre no es importante ni la creación de una tabla específica.



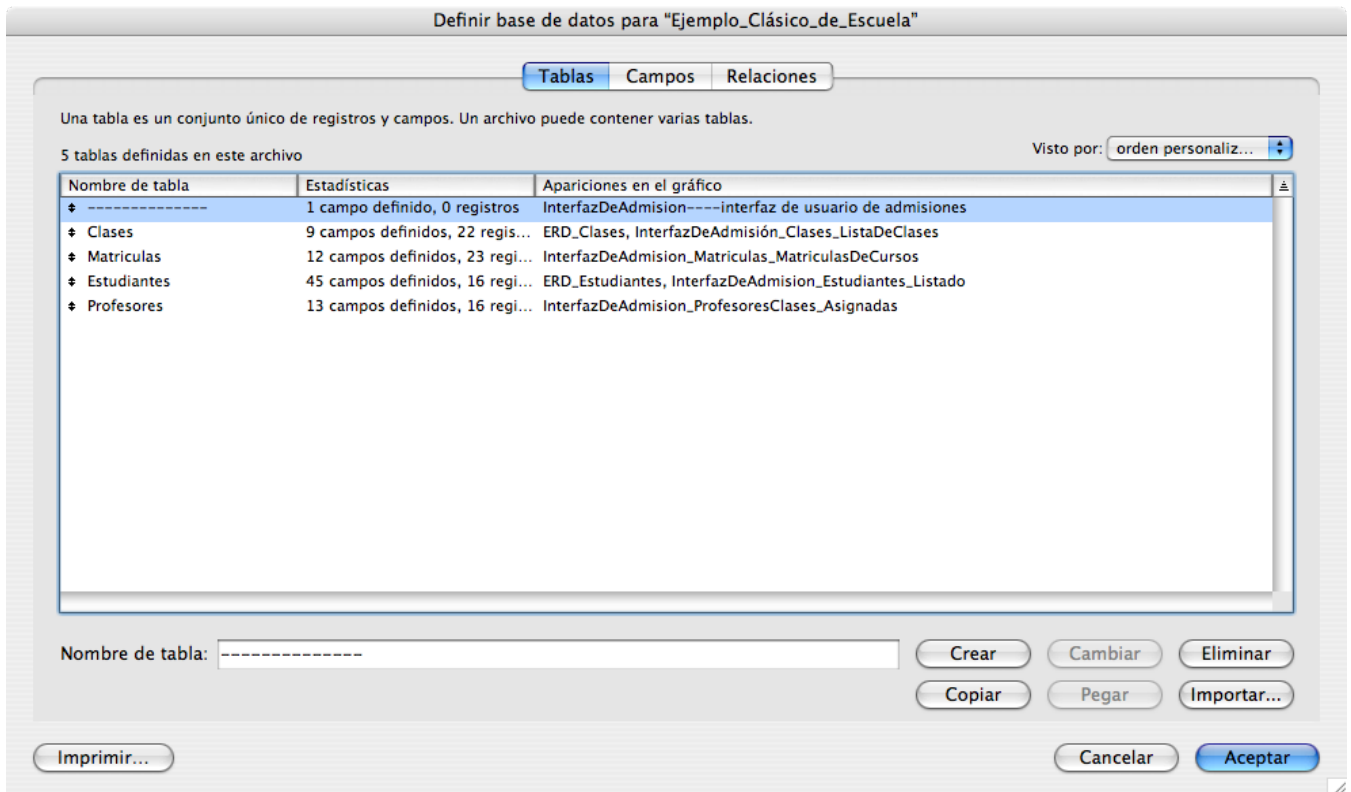


Figura 8

Cuando se crea la Tabla separadora entre categorías de tablas, se colocará en el Gráfico de relaciones su Ocurrencia de la Tabla creada en forma automática. Agregar las Ocurrencias de tablas, mediante esta tabla como Tabla de origen, y proveer un nombre que se utiliza para separar cada uno de los grupos funcionales en la lista desplegable. En el ejemplo que se observa en la figura 7 hay tres grupos funcionales.

- FDG\_\_\_\_\_Dependencia Funcional (Functional Dependency)\_\_\_\_\_
- PTO\_\_\_\_\_Ocurrencias de tablas principales (Primary Table Occurrences)\_\_\_\_\_
- InterfazDeUsuarioDeProfesor (TeacherUserInterface)\_\_\_\_\_Teacher UI\_\_\_\_\_

En una larga lista de Ocurrencias de tablas esto ayudará a encontrar la Ocurrencia de tablas que se busca al proporcionar cabeceras para cada FTOG.

### 4.3.3 Método de Agrupamiento de Ocurrencias de Tablas Base y Flotantes (Anchor Buoy) / Agrupamiento Jerárquico (Hierarchical Table Occurrence)

Este método del diseño del Gráfico de relaciones toma un grupo de Ocurrencias de Tablas (TO) que se conectan mediante líneas de relación, y agrega estructura y reglas.

En el número de septiembre de 2005 de *FileMaker Advisor*, Roger Jackes de Soliant Consulting explora este método en un artículo denominado "*Managing the FileMaker Pro 7 Relationship Graph*" (Forma de administrar el Gráfico de relaciones de FileMaker Pro7). Es imposible describir todos los detalles del método en este documento. Sin embargo, esta descripción se utilizará como base para algunos ejemplos. Roger Jackes aporta la siguiente descripción del modelo jerárquico/base y flotante.

"La naturaleza jerárquica del Método de tablas base se basa en un modelo. Cada TOG tiene una TO base y este TO se mantiene con datos relacionados a través de un número indeterminado de hilos de TO flotantes. Este modelo jerárquico se presta al requerimiento de que sólo haya una ruta entre dos TO cualesquiera. Después de iniciar un hilo, se pueden agregar todas las TO que sean necesarias para recuperar datos relacionados. La siguiente es una regla que es bueno seguir estrictamente: Si no se unen dos TOG con una línea de relación y FileMaker no permitirá que se haga una referencia circular.



“La TO base siempre es la más lejana a la izquierda y la cascada de TO flotantes la más lejana a la derecha. Se pueden agregar TO a medida que sean necesarias, ofreciendo el acceso a los datos relacionados para la TO base. En la práctica, los hilos rara vez se alargan a más de cuatro o cinco niveles hacia abajo.”

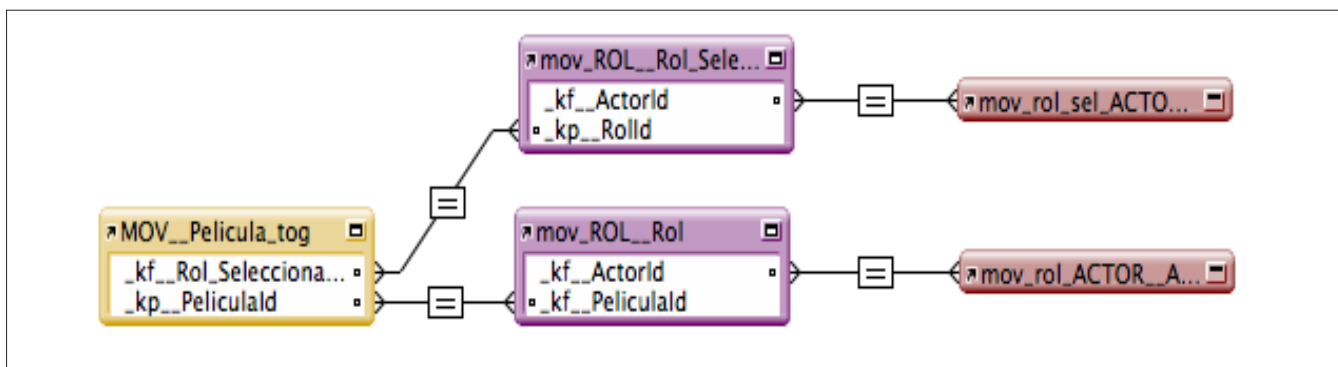


Figura 8a

“Obsérvese que ambos hilos utilizan las mismas tablas, pero con fines diferentes. En este ejemplo el hilo inferior se basa en las relaciones de clave principal y externa para las tres entidades, y enumera todos los roles y actores de la película actual. El hilo superior permite a los usuarios elegir una fila portal para los roles y ver después los detalles de dicho rol y actor en campos relaciones.»

Con este método, las presentaciones sólo pueden basarse en la TO «base» (en el extremo izquierdo). Las TO “flotantes” existen sólo para alimentar datos a cualquier presentación o lógica que se base en la tabla base”.

#### 4.3.3.1 VENTAJAS DEL MÉTODO DE AGRUPAMIENTO DE OCURRENCIAS DE TABLAS BASE Y FLOTANTES (ANCHOR BUOY) / (HTOG)

- Reduce significativamente la complejidad del gráfico
- Permite que se comprenda en cierta medida la estructura relacional del sistema, aparte del gráfico.
- La asignación de nombres es automática y no requiere demasiado pensamiento ni discreción, excepto para los metadatos de relación opcional (Nombre Descriptivo).
- En general los TOG se basan en presentaciones lo cual facilita localizar las TO en el Gráfico de relaciones.
- Funciona muy bien en soluciones de gran envergadura.
- Aporta la agrupación funcional dentro de los menús de Ocurrencia de tablas aparte del Gráfico de Relaciones en orden jerárquico.
- Admite y no resulta afectado por la norma de la «ruta única» del Gráfico de relaciones.
- Las tablas relacionadas dentro de un FTOG son bastante limitadas, y se agrupan en un tipo de orden alfabético dentro del FTOG.
- Admite una amplia gama de metodologías de desarrollo: aquellas que se ejecutan bajo soluciones “controladas por portales» altamente encriptadas y soluciones con bases más abiertas que utilizan controles nativos FileMaker.
- Es accesible para una amplia gama de desarrolladores, y lo suficientemente sólido para profesionales.

#### 4.3.3.2 DESVENTAJAS DEL MÉTODO DE AGRUPAMIENTO DE OCURRENCIAS DE TABLAS BASE Y FLOTANTES (ANCHOR BUOY) / (HTOG)

- Aumenta en forma significativa la cantidad de TO.
- No admite en forma inherente el modelo de relación bidireccional. La restricción de limitar las presentaciones sólo a la tabla base requiere que se cree otro Agrupamiento de Ocurrencias de tablas base y flotantes/ TOG para expresar los datos de derecha a izquierda en la presentación.



### 4.3.3.3 PAUTAS DE AGRUPAMIENTO DE OCURRENCIAS DE TABLAS BASE Y FLOTANTES (ANCHOR BUOY) / HTOG "EN CUMPLIMIENTO DE LAS NORMAS"

1. Las palabras individuales deben separarse en forma coherente mediante el método establecido para cada sección de la sintaxis.

#### Métodos de separación de palabras recomendados (Ocurrencias de tablas: Base y flotante)

Tablas base		
NombreDeCabeceraDeTOGBase	MAYÚSCULAS	
NombreDeTablaDeOrigen	Minúsculas y CamelCase	*Debe coincidir con las mayúsculas y/o minúsculas del nombre de la tabla
Tablas flotantes		
NombreDeCabeceraDeTOGBase	Minúsculas y CamelCase	
AbreviaturaDelNombreDeTablaDeOrigen (SourceTableNameAbbreviation)	MAYÚSCULAS	Abreviatura sugerida para minimizar la longitud
NombreDeTablaDeOrigen (SourceTableName) Minúsculas y CamelCase	Minúsculas y CamelCase	*Debe coincidir con las mayúsculas y/o minúsculas del nombre de la tabla
NombreDescriptivo	Minúsculas y CamelCase o Mayúsculas y CamelCase	

2. Sintaxis de tabla base  
`<<AnchorTOGHeaderName>>[_]<SourceTableName>`  
 Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.
  - NombreDeCabeceraDelTOGBase – obligatorio; definido por el desarrollador  
 Identifica el Grupo de Ocurrencia de Tablas (TOG) Cada Ocurrencia de tablas ‘conectada’ con esta tabla base tendrá el mismo Nombre de cabecera del TOG Base.
  - “\_” – obligatorio  
 Los subrayados dobles separan el Nombre de cabecera del TOG base del Nombre de la tabla de origen.
  - Nombre de tabla de origen:  
 Indica la Tabla relacionada con la Ocurrencia de tablas.
3. Sintaxis de tabla flotante  
`<<AnchorTOGHeaderName>>[_]<<SourceTableNameAbbreviation>>[_]<SourceTableName>[_]DescriptiveName`  
 Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.
  - Nombre de cabecera del TOG base (AnchorTOGHeaderName); obligatorio, utiliza el valor definido para la tabla base  
 Identifica cada Ocurrencia de Tablas del Grupo de Ocurrencia de tablas (TOG) ‘conectada’ con esta tabla base que tendrá el mismo Nombre de cabecera del TOG Base.
  - “\_” – obligatorio  
 Los subrayados dobles separan el Nombre de cabecera del TOG base (AnchorTOG HeaderName) del primer Nombre abreviado de la tabla de origen.
  - Abreviatura del nombre de la tabla de origen (SourceTableNameAbbreviation); obligatorio; definido por el desarrollador  
 Cada Abreviatura del nombre de la tabla de origen (SourceTableNameAbbreviation) se separa mediante un subrayado simple «\_».
  - “\_” – obligatorio  
 Los subrayados dobles separan la Abreviatura del nombre de la tabla de origen (SourceTableNameAbbreviation) del Nombre de la tabla de origen.



- SourceTableName (Nombre de la tabla de origen): obligatorio; valor previsto del Nombre de tabla de origen real.  
Indica la Tabla relacionada con la Ocurrencia de tablas.
- “\_\_” – obligatorio  
Los subrayados dobles separan el Nombre de la tabla de origen del Nombre descriptivo.
- Nombre descriptivo  
En Ocurrencias de tablas flotantes, el nombre descriptivo es opcional, pero puede aportar una descripción natural de la TO o quizás algunos metadatos que indican en qué se basa la relación. Por ejemplo, el Nombre descriptivo “CompanyID” indicaría que la relación se basa en el campo ID de la empresa.

Ejemplos:

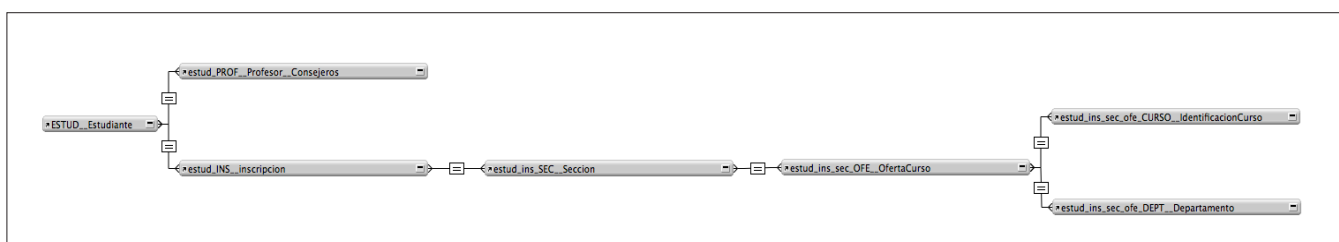


Figura 9

- Base: ESTUD\_\_Estudiante (Anchor STUD\_\_Student): Se trata de un TOG base, porque siempre es la TO que está más a la izquierda y sólo enumera la Tabla de origen. El nombre también indica que se basa en la Tabla de origen “Student” (Estudiante).
- estud\_insc\_SEC\_Seccion (stud\_reg\_SEC\_\_Section): Esta TO flotante es parte del Grupo base “stud” (“estudiante”). Es la tercera TO conectada a «ins» (inscripción) que se conecta con el «stud» base. Se basa en la Tabla de origen “Section” (Seccion). No existe Nombre Descriptivo en este caso.
- estud\_ins\_sec\_ofe\_CURSO\_\_IdentificacionCurso (stud\_reg\_sec\_\_off\_COURSE CourseIdentification): Esta TO flotante es parte del Grupo base “stud” (“estudiante”). Se ubica a cinco niveles por abajo con una Tabla de origen de “Course” (Curso). Está conectada a «ofe» (oferta de cursos), “sec” (sección), “reg” (inscripción), y por último a la base “stud” (Estudiante). También contiene el uso opcional de Nombre descriptivo con Identificación de Curso.

## 4.4 Consideraciones complementarias

Ocurrencia principal de la tabla: Sin tener en cuenta la metodología de Gráfico de relaciones utilizada, al crear cálculos se debe elegir el contexto a partir del cual se evaluará el cálculo. En el caso en que un resultado de cálculo se derive sin hacer referencia a datos relacionados, esos cálculos “internos” siempre se basarán en el mismo contexto. El concepto de una Ocurrencia de Tablas especial denominada Ocurrencia principal de tablas o PTO puede ayudar con esta cuestión. La PTO siempre funciona como la Ocurrencia de tablas designada que se utilizará al crear cálculos internos. La figura 8 demuestra el uso de la PTO para la tabla de origen Empleados.



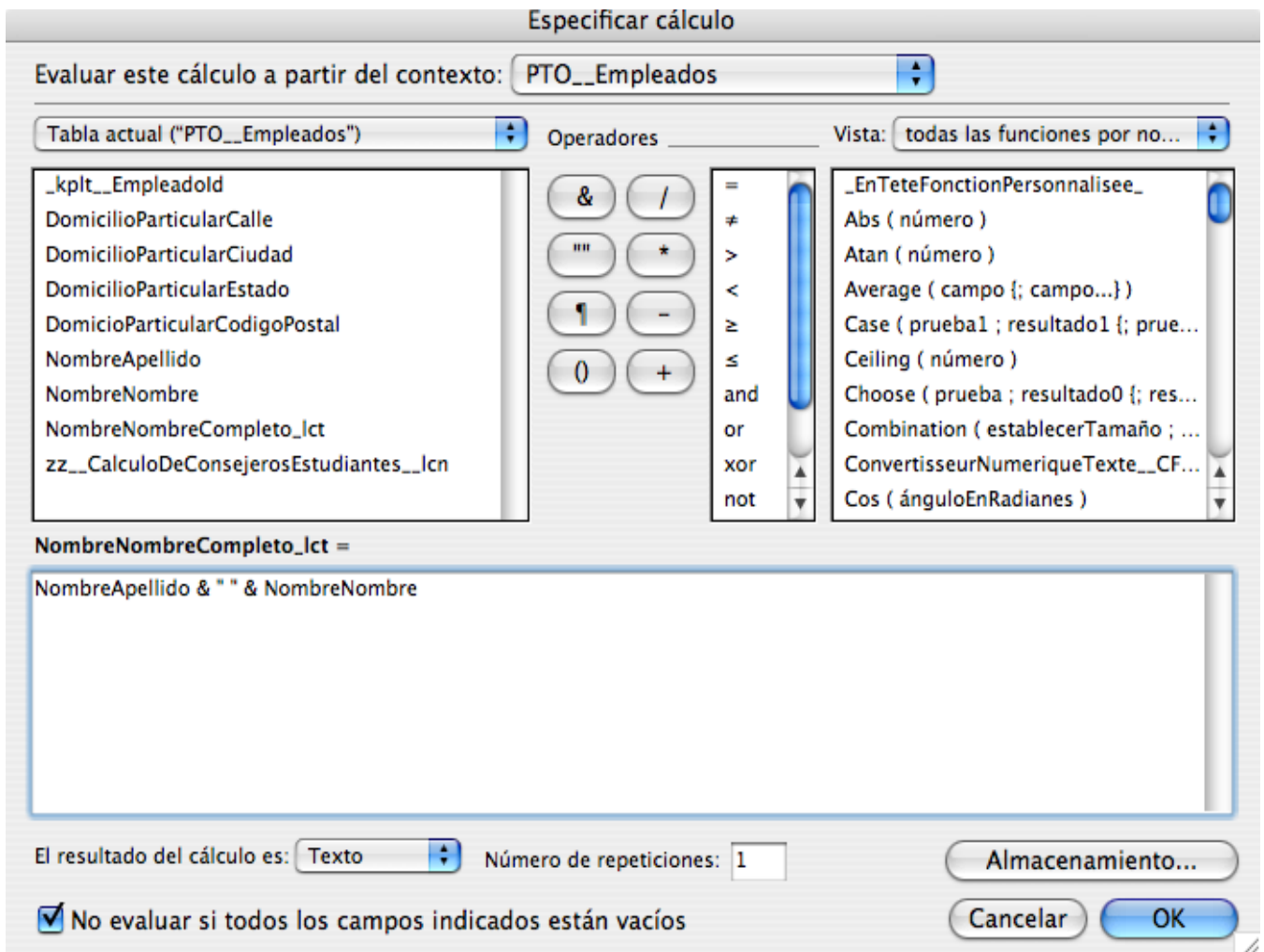


Figura 10

Al utilizar una PTO para cada cálculo interno, siempre se sabrá que la PTO suministra el contexto en el cual se evaluará el cálculo. A medida que se desarrolla una solución recordar qué serie de Ocurrencias de tablas se utilizan para evaluar estos cálculos internos no será necesario.

Eliminar otra Ocurrencia de tablas no afectará los cálculos que utilicen la PTO como contexto de evaluación.

Sintaxis:

[PTO][\_\_]<SourceTableName>

Recomendada: [PTO\_\_]<<SourceTableName>> (Nombre de tabla de origen)

Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.

- (PTO) Designador de Ocurrencia principal de tablas (PrimaryTableOccurrenceDesignator): obligatorio; definido por el desarrollador; Puede ser definido por el desarrollador. Se recomienda el uso de la "PTO" en las FDC.
- " \_\_ " – obligatorio  
Los subrayados dobles separan la PTO del Nombre de la tabla de origen.
- Nombre de tabla de origen (SourceTableName):  
El nombre de la Tabla relacionada con la Ocurrencia de tablas.



## 5 ▷ Presentaciones

Una de las primeras consideraciones es la forma en que la asignación de nombres a las presentaciones está principalmente impulsada por quienes utilizarán el nombre. En aquellas soluciones en las que el usuario final selecciona directamente la presentación del menú desplegable de presentaciones en el área de estado es importante proveer un nombre que sea autodescriptivo y en general no encriptado con metadatos adicionales. En forma inversa, en aquellas soluciones en que el usuario final está aislado o al cual se le prohíbe utilizar el menú desplegable de presentaciones, el nombre sirve al desarrollador.

Otras consideraciones incluyen utilizar un conjunto de caracteres que no afecte la conectividad externa. Por ejemplo las consultas XML utilizan el nombre de la presentación para establecer el contexto. Los nombres de las presentaciones deben ser compatibles. Una vez más, no todas las soluciones deberán ser limitadas necesariamente por esto; sin embargo, la preparación de la evolución de la solución puede reducir los esfuerzos de desarrollo en algún momento en el futuro. También se debe considerar una convención de mayúsculas y/o minúsculas por razones de coherencia y profesionalismo. La longitud de un nombre de presentación puede ser de hasta 100 caracteres, sin embargo, hay algunos cuadros de diálogo que no exhibirán todo ese contenido. El problema común es la imposibilidad de saber qué Ocurrencia de tablas (TO) está ligada a una presentación durante el desarrollo. Excepto en cuanto se refiere a dejar el contexto de las actividades de desarrollo, entrar en el modo de presentación, y en la configuración de la presentación, esta información no se encuentra disponible. Más aún, muchos diseños de soluciones buscan la posibilidad de ganar programáticamente y utilizar la asignación de la Tabla de origen a una TO para la interfaz y la lógica de programa. La imposibilidad de obtener esta información desde una perspectiva de desarrollo o de programa lleva a desear la codificación o inclusión de metadatos acerca de la Ocurrencia de la Tabla y del Nombre de la tabla de origen dentro del nombre de la presentación.

Cada vez que se utilizan notaciones se hace más difícil el nombre para el usuario final. Lamentablemente no existe forma alguna de proveer un nombre de presentación que sea fácil para el usuario y fácil para el desarrollador. Por lo tanto, según los requerimientos de la solución y el diseño será necesario considerar las compensaciones. Y por último, aparte del nombre estructurado o codificado para la presentación, no existe forma alguna de categorizar las presentaciones. Por ejemplo, las presentaciones se utilizan para interfaces, informes, trabajo 'oculto', propuestas y muchos otros usos específicos. No se pueden categorizar. Es útil para comprender cuál es el objeto de la presentación. Esta es la razón por la cual se desea encapsular alguna categorización en el nombre de la presentación.

### 5.1 Objetivos:

- ✓ Conjunto de caracteres recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada
- ✓ Cuestiones de longitud de los nombres de presentación.
- ✓ Aportar un método para encarar la imposibilidad de identificar la Ocurrencia de tablas ligada a una presentación sin ingresar la configuración de la presentación.
- ✓ Proveer un método para en afrontar entorno de desarrollo los problemas de visualización de metadatos acerca del Nombre de la tabla de origen de la presentación durante la ejecución.
- ✓ Proporcionar un método para tratar la imposibilidad de obtener en entorno de desarrollo metadatos acerca de todas las presentaciones diferentes de la que se está utilizando.
- ✓ Obsérvese la imposibilidad de ofrecer un nombre más "fácil para el usuario" a un nombre más útil para el desarrollador.
- ✓ Se debe evitar dar el mismo nombre a una TO y a una Presentación.
- ✓ Aportar un método para enfrentar la necesidad de comprender el "objeto" de la presentación.
- ✓ Proveer un método para encarar la imposibilidad de organizar presentaciones por función, categoría, etc.
- ✓ Nota solamente: Cuestiones relacionadas con tecnologías de conectividad (XML, ODBC, JDBC), incluso palabras Reservadas de SQL.



## 5.2 Definición de problemas:

**Conjunto de caracteres recomendado:** El conjunto de caracteres utilizado para las presentaciones debe incluir las limitaciones de caracteres impuestas por el mismo FileMaker así como también toda tecnología que utilice nombres de presentaciones al comunicarse con FileMaker, como por ejemplo consultas XML.

**Convención de mayúsculas/minúsculas recomendada:** La recomendación, en este caso, no consiste en superar un problema específico sino en aplicar un criterio coherente para adoptar la manera de dar nombres a las presentaciones. El objetivo es seleccionar un método y ser coherente al respecto.

**Longitud del nombre de la presentación:** Una presentación FileMaker puede tener hasta 100 caracteres. Sin embargo, en muchos casos éste no resulta un límite práctico. Con el lanzamiento de la versión 8, la mayoría de los cuadros de diálogo no presentarán dificultad alguna para exhibir el nombre completo de 100 caracteres. Sin embargo, no se puede cambiar el tamaño del cuadro de diálogo de configuración de presentaciones y esto limita los caracteres que podrán visualizarse a 42 en OS X y 48 en Windows.

**Dificultad en identificar la Ocurrencia de tablas ligada a una presentación sin entrar en la Configuración de presentaciones:** No existe forma alguna, excepto una gran memoria, de saber a qué Ocurrencia de tablas está relacionada una presentación que no sea entrar en el Modo Presentación y abrir el cuadro de diálogo de Configuración de Presentaciones. Esta falta de información mientras se efectúa un desarrollo puede ser molesta, requiere entrar y salir de las actividades de programación sólo para obtener información fundamental de la presentación.

Este hecho dio lugar al deseo de incluir notaciones de la relación de la Ocurrencia de Tablas de Presentaciones dentro del nombre de la presentación.

**Dificultad de visualizar en entorno de desarrollo metadatos acerca del nombre de la tabla de origen de la presentación durante la ejecución:** Los desarrolladores tal vez quieran obtener información en entorno de desarrollo acerca de la presentación, como por ejemplo el Nombre de la Ocurrencia de tablas o el Nombre de la tabla, a fin de realizar una acción en una forma en particular. Mientras la función Get (LayoutTableName) (Obtener (Nombre de tabla de presentación) volverá al Nombre de Ocurrencia de la tabla, no volverá al nombre de tabla real (nombre de Tabla de origen relacionado con la Ocurrencia de la tabla). Este hecho dio lugar al deseo de incluir el Nombre de la tabla de origen dentro del nombre de la presentación.

**Dificultar la visualización, en entorno de desarrollo, de metadatos acerca de todas las presentaciones diferentes de la que se está utilizando:** La necesidad de obtener en entorno de desarrollo acerca de todas las presentaciones en un archivo se une estrechamente al problema anterior. La función de diseño LayoutName(FileName) (Nombre de presentación (nombre de archivo) volverá a todas las presentaciones de un archivo. No obstante, no existe una función para obtener las Ocurrencias de tablas relacionadas con éstas que no sea utilizar la función Get (LayoutTableName) (Obtener el nombre de tabla de la presentación) para cada una individualmente, y ninguna forma de obtener el Nombre de la tabla de origen. Este hecho dio lugar al deseo de incluir el Nombre de la tabla de origen y el Nombre de la Ocurrencia de la tabla dentro del Nombre de la presentación.

**Dificultad para ofrecer un nombre más "fácil para el usuario" con el nombre útil para el desarrollador:** Hasta el FileMaker 8 inclusive, no existe la posibilidad de ofrecer un nombre que sea fácil para el usuario y útil para el desarrollador. El desarrollador debe tomar una decisión acerca del método que utilizará en su solución.

**Igual nombre de Ocurrencia de Tablas y de Presentaciones:** De forma predeterminada, FileMaker asignará a las presentaciones el mismo nombre que la Ocurrencia de tablas sobre la cual se basa la presentación. Aunque esto ahorra tiempo en muchos aspectos, puede provocar también ciertos problemas al intentar obtener información en entorno de desarrollo.

Los desarrolladores deben comprender el objeto de la presentación: Al desarrollar una solución, el desarrollador en muchos casos desea ofrecer alguna indicación del objeto de la presentación en el nombre de ésta. Por ejemplo, todas las presentaciones de informes pueden llevar el prefijo «rep» (informe). Además, las presentaciones con el prefijo "work" (trabajo) se podrían utilizar para indicar que la presentación se utiliza en entorno de desarrollo para realizar alguna función dentro de un guión. Independientemente de la categorización, se desea indicar la 'función' de la presentación, tanto desde la perspectiva de la organización como de la comprensión en entorno de desarrollo.



**Imposibilidad de organizar las presentaciones por función, categoría y otros aspectos:** Ya sea que se esté eligiendo una presentación desde una selección de la pestaña de presentaciones u otros cuadros de diálogo, puede ser molesto revisar una lista larga que sólo está categorizada en orden alfabético. A la lista de presentaciones le falta toda posibilidad de categorización. Este hecho ha dado lugar al deseo de utilizar una estructura de nombres que pueda desglosar esta lista en categorías o funciones para reducir el tiempo de localización de nombres de presentaciones.

### 5.3 Pautas “en cumplimiento de las normas” (Presentaciones):

1. Sólo deben utilizarse los siguientes caracteres:
  - Mayúsculas y minúsculas aA - zZ
  - Número: 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble “\_” “\_\_”
2. NO deben contener espacios.
3. NO debe comenzar con números.
4. Se debe ser coherente con el uso de nombres en singular o plural.
5. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Presentaciones) :

Minúsculas y CamelCase	nombreDeMiPresentacion
Mayúsculas y CamelCase	NombreDeMiPresentacion
Subrayado simple (minúsculas)	nombre_de_mi_presentacion
Subrayado simple (Tipo Título)	Nombre_de_Mi_Presentacion
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_PRESENTACION

6. Sintaxis: <<FunctionPrefix>>[\_]DescriptiveName[\_]<TableOccurrenceName>  
Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.
  - Prefijo de función - opcional; definido por el desarrollador  
Provee la sintaxis general que admite los prefijos definidos por el desarrollador, determinados por sus necesidades específicas y preferencias, mientras define una sintaxis universalmente comprendida. Si se decide no utilizar ningún Prefijo Funcional simplemente éste se omite y se comienza por el Nombre descriptivo. No es necesario incluir el primer “\_”. La recomendación general es utilizar un indicador con uno a cinco caracteres y elegir la convención de mayúsculas y/o minúsculas para la función.
  - “\_”; obligatorio  
El uso de un subrayado doble “\_\_” es el carácter. Esto permite el uso del subrayado dentro del Nombre Descriptivo y del Nombre de la Ocurrencia de Tablas mientras incorpora posibilidades de legibilidad y análisis.
  - Nombre descriptivo;  
Ofrece un método para dar un nombre a la presentación que de alguna manera sea fácil para el usuario.
  - “\_”; obligatorio  
El uso de un subrayado doble “\_\_” es el carácter. Esto permite el uso del subrayado dentro del Nombre Descriptivo y del Nombre de la Ocurrencia de Tablas al tiempo que incorpora posibilidades de legibilidad y análisis.
  - Nombre de Ocurrencia de tablas; obligatorio, utiliza Nombres de Ocurrencias de tablas definidos.  
Esto aporta un indicador/localizador visual y de programación aparte del Gráfico de relaciones para la Ocurrencia de la tabla subyacente.



## 6 ▷ Funciones personalizadas

Las funciones personalizadas ofrecen un nivel sin precedentes de extensibilidad para el entorno de desarrollo. Ofrecen a los desarrolladores la posibilidad de ir mucho más allá de las funciones predeterminadas incorporadas en el FileMaker.

FileMaker tiene una sintaxis específica y una convención de mayúsculas y/o minúsculas para todas las funciones incorporadas. Por ejemplo, la Función de Texto "MiddleWords" (Palabras intermedias) lee las `Palabras Intermedias (texto; palabra inicial; cantidad de palabras)`. La función usa Mayúsculas y CamelCase y cada parámetro usa minúsculas y CamelCase. La única excepción destacable la constituyen las Funciones 'Get' (obtener) que usan una convención de mayúsculas y/o minúsculas diferente. Esto se debe simplemente a la naturaleza de la función 'Get'. Todas las Funciones 'Get' (Obtener) usan "palabras reservadas". No existen parámetros para las Funciones 'Get' y por ello la sintaxis es diferente. La función `Get (WindowDesktopHeight)` (Obtener (Ventana Escritorio Altura) es un buen ejemplo. Obsérvese que el nombre de la función "Get" siempre utiliza la convención Mayúsculas y CamelCase y el valor que devolverá también usa Mayúsculas y CamelCase). Por razones de coherencia, tiene sentido utilizar la convención de mayúsculas y/o minúsculas y la sintaxis ya establecida dentro del producto.

Las funciones pueden ser independientes o trabajar en conjunto con otras funciones personalizadas. No existe forma alguna de categorizar o agrupar estas funciones relacionadas. Algunos desarrolladores sugieren que una Función Personalizada tiene entradas y salidas y que no necesita agrupamiento. Pero otros creen que es importante comprender que una función se apoya en otras funciones o las utiliza. En las FDC se ha optado por hacer esta distinción utilizando los términos Funciones personalizadas "Privadas" o "Públicas".

Al revisar un cálculo no siempre queda claro que éste utiliza una función personalizada. Es importante poder identificar una función Personalizada dentro de un cálculo. El motor de cálculo no diferencia entre funciones incorporadas y personalizadas. Esto dificulta localizar y resolver problemas en un cálculo y comprenderlo. Al proveer un indicador dentro del nombre de la función Personalizada éste será fácilmente reconocible dentro del cálculo.

Por último, uno de los valores clave de las Funciones Personalizadas es ofrecer funcionalidades ampliadas que sólo existen dentro de la función en sí. Cualquier persona que utilice esta función debe comprender lo que ésta hace y la forma en que lo hace. Por lo tanto, es importante documentarla ampliamente para futura referencia y para otras personas que puedan utilizarla. La recomendación incluida en las FDC es ofrecer un conjunto base de pautas sobre qué y cómo documentar una función Personalizada. Además, se debe consultar la Sección de [Cálculo](#) de este documento donde se incluyen pautas acerca del formato del cálculo de función Personalizada.

### 6.1 Objetivos:

- ✓ Conjunto de caracteres recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada
- ✓ Separación sintáctica
- ✓ Pautas de longitud
- ✓ Método para agrupar funciones relacionadas
- ✓ Método para diferenciar funciones personalizadas dentro de los cálculos
- ✓ Método para diferenciar entre funciones principales/ secundarias.
- ✓ Pautas sobre configuraciones de seguridad en funciones Personalizadas Privadas y Públicas.
- ✓ Método para documentar una función personalizada que incluya elementos mínimos y formato.



## 6.2 Definición de problemas:

**Conjunto de caracteres recomendado:** Las funciones personalizadas ofrecen ampliar las funciones predeterminadas en el FileMaker Pro/Advanced. Por razones de coherencia, se recomienda continuar con el conjunto de caracteres utilizado dentro de la lista de funciones del FileMaker Pro y del FileMaker Pro 8 Advanced.

**Convención de mayúsculas/minúsculas recomendada:** La decisión respecto de la forma de separar las palabras dentro de los nombres de las funciones Personalizadas deberá ser coherente con las funciones incorporadas. Todas las funciones incorporadas aplican la misma convención de mayúsculas y/o minúsculas. Se utilizan mayúsculas y CamelCase para el nombre y minúsculas y CamelCase para todos los parámetros. Además, se necesita la sintaxis de la función Personalizada para separar todo prefijo necesario con fines organizativos.

**Sintaxis:** La sintaxis de la función personalizada debe incluir coherencia con las funciones incorporadas, diferenciar los prefijos del nombre, y brindar soporte al agrupamiento de funciones relacionadas por nombre.

**Recomendaciones respecto de la longitud:** Una función Personalizada puede tener hasta 100 caracteres. Sin embargo, se debe tener en cuenta que los nombres largos pueden ser incómodos para trabajar en el cuadro de diálogo de cálculos. El cuadro de diálogo de cálculos exhibe 26 caracteres en OS X y 30 caracteres en Windows tanto en FileMaker Pro como en FileMaker Pro 8 Advanced.

**Funciones relacionadas con agrupamientos:** Es habitual ver un grupo de dos o más funciones que trabajan en conjunto para ofrecer una determinada funcionalidad. La convención debe definir un método que permita el reconocimiento visual de las funciones relacionadas y que los desarrolladores comprendan las dependencias entre los elementos del grupo.

**Diferenciación de las Funciones personalizadas dentro de los Cálculos:** Al interrogar un cálculo no queda claro si un elemento es una función incorporada o personalizada. La convención debe definir un método que distinga claramente entre ambos.

**Seguridad de las Funciones personalizadas:** La convención debe ofrecer cierta comprensión respecto de configurar la disponibilidad de las funciones Personalizadas Públicas y Privadas.

**Documentación de las Funciones personalizadas:** La convención ofrecerá un modelo para la documentación de los diversos elementos de una función Personalizada. Éstos deben incluir los elementos, el formato y la ubicación.

## 6.3 Pautas “en cumplimiento de las normas” (Funciones personalizadas):

### 6.3.1 Funciones personalizadas públicas

Una función Personalizada Pública es una función personalizada que el diseñador creó para ser llamada en forma directa. Estas funciones podrían ser las funciones “principales” dentro de un grupo de funciones Personalizadas (que a su vez convocan las funciones Personalizadas privadas dentro del grupo), o podrían ser funciones personalizadas individuales e independientes, sin dependencias. En todos los casos el término se refiere a funciones cuyo objeto es ser llamadas en forma directa. Por ejemplo, si se tienen tres funciones Personalizadas;

- `UrlDestacar (UrlHighlight)`
- `UrlUltCar (UrlLastChar)`
- `UrlDevolverTodos (UrlReturnAll)`
- `UrlDevolverUno (UrlReturnOne)`

Al utilizar el agrupamiento de nombres es fácil reconocer que este conjunto de funciones personalizadas están relacionadas. Todas aportan determinada funcionalidad de trabajo a algo relacionado con los URL (Localizador uniforme de recursos). No queda clara la estructura de la función. Sin conocimiento interno del conjunto de funciones, es imposible saber si uno cualquiera o más de éstas dependen una de otra. Clasificando cada función como Pública o Privada se puede saber qué función debe llamarse primero.



- `UrlDestacar_CFpub` (`UrlHighlight_CFpub`)
- `UrlUltCar_CFpvt` (`UrlLastChar_CFpvt`)
- `UrlDevolverTodos_CFpvt` (`UrlReturnAll_CFpvt`)
- `UrlDevolverUno_CFpvt` (`UrlReturnOne_CFpvt`)

Esta convención indica que `UrlHighlight_CFpub`, dentro del grupo URL, está diseñada para ser llamada primero y que cada una de las otras funciones son subordinadas de la primera.

Sintaxis:

`CustomFunctionName[___][CFpub]`

Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.

- Con el sufijo “\_CFpub” (CF en mayúsculas, pub en minúsculas): Identifica, dentro de un cálculo, que la función es una función personalizada de tipo público.
- Sólo deben utilizarse los siguientes caracteres:
  - aA - zZ mayúsculas y minúsculas
  - Número: 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble “\_” “\_\_”
- NO deben contener espacios.
- NO deben comenzar con números.
- NO deben contener puntos.
- Se deben utilizar mayúsculas y CamelCase para el nombre de la función
- Ejemplo: `ArrayBuilder_CFpub`

### 6.3.2 Funciones personalizadas privadas

El objeto del término “privado” no es denotar acceso limitado, antes bien comunicar la intención del diseñador de la función: Una función personalizada privada tiene el objeto de ser convocada sólo por otra función personalizada, ya sea una función personalizada pública u otra función personalizada privada. Al clasificar una función personalizada como pública se indica que su objeto es que sea llamada en forma directa, y que puede tener dependencias con otras funciones personalizadas.

Por el contrario, una función Personalizada privada tiene siempre el objeto de ser convocada por otra función personalizada y nunca en forma directa.

¿Qué sucede si se llama directamente a una función “privada”?

¡Nada! El objetivo de los términos “pública” y “privada” es simplemente ilustrar una relación de dependencia y no restringir lo que se puede hacer con una función. En la práctica, si se trata de llamar a una función Privada directamente, es posible que se desee renombrarla como función Pública en lugar de privada.

Sintaxis:

`CustomFunctionName[___][CFpvt]`

Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.

- Con el sufijo “\_CFpvtf” (CF en mayúsculas, pvt en minúsculas): Identifica, dentro de un cálculo, que la función es una función personalizada de tipo privado.
- Sólo deben utilizarse los siguientes caracteres:
  - aA - zZ mayúsculas y minúsculas
  - Número: 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble “\_” “\_\_”
- NO deben contener espacios.
- NO deben comenzar con números.
- NO deben contener puntos.



- Se deben utilizar mayúsculas y CamelCase para el nombre de la función
- Ejemplo: `ArrayBuilderHelper_CFpvt`

### 6.3.3 Parámetros de la función personalizada

Ya sea en Funciones Privadas o Públicas, los parámetros deben seguir el formato estándar para todas las funciones del FileMaker.

Sintaxis:

`CustomFunctionName[___] [CFpub] or [CFpvt] ( parameterOne, parameterTwo) (Nombre de la función personalizada[___] [CFpub] or [CFpvt] (parámetro uno, parámetro dos)`

Ver en el punto [Leyenda de sintaxis](#) la descripción de ésta.

- Sólo deben utilizarse los siguientes caracteres:
  - aA - zZ mayúsculas y minúsculas
  - Número: 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble " \_ " \_ "
- NO deben contener espacios.
- NO deben comenzar con números.
- NO deben contener puntos.
- Se deben utilizar minúsculas y CamelCase.
- Ejemplo: `MyCustomFunction_CFpub(parameterOne, parameterTwo ) (MiFunciónPersonalizada_CFpub (parámetro uno, parámetro dos)`

### 6.3.4 Ejemplos de asignación de nombres a funciones personalizadas

La figura 11 ilustra un conjunto de funciones Personalizadas que utilizan la convención de nombres en cumplimiento con las normas.

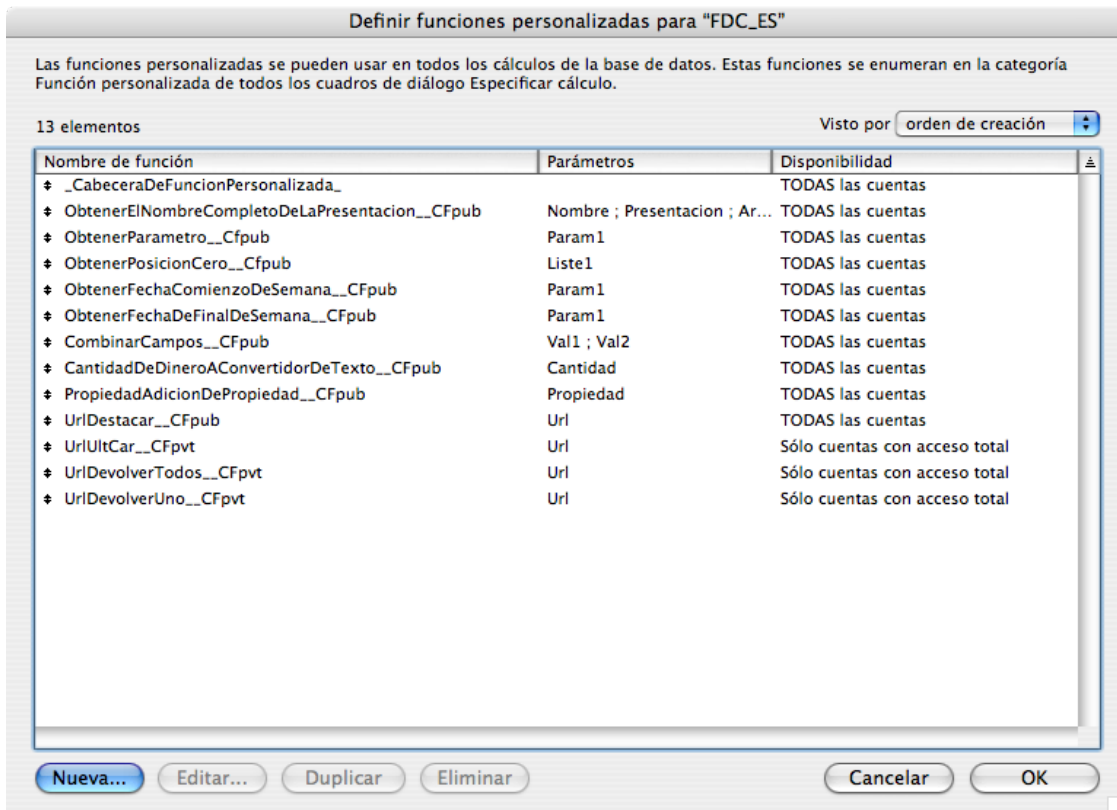


Figura 11



### 6.3.5 Documentación de las funciones personalizadas

Cada función personalizada debe contener un juego mínimo de documentación. Deben estar representadas línea por línea y ser presentadas en el siguiente orden:

1. Nombre: Nombre de la función
2. Historial: (podría incluir cualquiera de los siguientes elementos u otros que se consideren necesarios)
  - Nombre del desarrollador: Nombre del desarrollador de la función
  - E-mail del desarrollador: Correo electrónico del desarrollador de la función
  - Nombre del modificador: Nombre del desarrollador de la función que la modificó por última vez.
  - E-mail del modificador: Correo electrónico del desarrollador que modificó la función por última vez.
  - Fecha de la creación: Fecha en formato DD–MMM–YYYY en que se creó la función.
  - Fecha de la última modificación: Fecha en formato DD–MMM–YYYY en que se modificó la función por última vez.
3. Objeto: Descripción de aquello para lo cual se ha diseñado la función.
4. Parámetros: Enumera todos los parámetros utilizados por la función.
5. Notas importantes. Información adicional.
6. Ejemplo:

/\*

Nombre:

*MiFuncionPersonalizada\_\_CFpub*

Historial:

*Creado por N.N.*

*Fecha de creación: 01–ENE– 2005.*

*Fecha de modificación: 01–ENE–2005*

Objeto:

*Esta función es un ejemplo que muestra el formato de la documentación.*

Parámetros:

*parámetro uno: Ejemplo de explicación de parámetro. Esto debe incluir todo detalle pertinente.*

*parámetro dos: Ejemplo de explicación de parámetro. Esto debe incluir todo detalle pertinente.*

Notas importantes:

*Provee toda la información adicional que constituirá conocimiento útil para los desarrolladores del futuro y como referencia.*

\*/

### 6.3.6 Formato de la función personalizada

Ver la Sección de [cálculo](#) donde se describe el Formato de los Cálculos de funciones personalizadas.

## 6.4 Consideraciones complementarias

Las funciones Personalizadas se pueden configurar para que estén disponibles en el motor de cálculo para "Todas las cuentas" o "Sólo las cuentas a las que se han asignado privilegios de Acceso Completo o Total". Como una forma de aplicar privacidad se podría considerar configurar las funciones Personalizadas públicas para "todas las cuentas" y las funciones Personalizadas privadas para "Sólo las cuentas a las que se han asignado privilegios de Acceso Completo o Total". Esto colocará las funciones Personalizadas públicas a disposición del motor de cálculo, mientras se ocultan las funciones privadas.



## 7 ▷ Guiones

La administración de guiones se desglosa en dos categorías de organización y documentación. Primero, no se debe tener en cuenta, realmente, la organización de guiones cuando se trata de asignar nombres. El Script-Maker no incluye herramientas organizativas y muchos desarrolladores han recurrido a utilizar sus asignaciones de nombres a guiones para ayudarse en cierta forma. Aunque sí se pueden mover manualmente los guiones hacia arriba y hacia abajo en una lista larga no hay una forma tangible de categorizar un conjunto de guiones. Por lo tanto, las únicas opciones que quedan son la posición y el nombre. Esta versión de las FDC no tiene como objetivo enfocarse en una nomenclatura que trate con la organización. No obstante, hay algunas pocas áreas que se aplican a la documentación algunas normas acerca de la asignación de nombres que pueden ser útiles al enfrentar algunos de los obstáculos que dificultan la tarea de todos los desarrolladores.

### 7.1 Objetivos:

- ✓ Conjunto de caracteres recomendado
- ✓ Conjunto de caracteres variables recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada (Variables)
- ✓ Cuestiones de longitud de los nombres de guiones.
- ✓ Encara la imposibilidad de reconocer un guión que requiere parámetros al visualizar el nombre del guión.
- ✓ Imposibilidad en ofrecer nombres más "fáciles para el usuario" o más útiles para el desarrollador para aquellos guiones que son accesibles para el usuario final
- ✓ Imposibilidad de detener el guión en el último punto mientras se utiliza el depurador, requiere un paso extra en el guión.
- ✓ Modelo coherente para documentar un guión.
- ✓ Problemas con las tecnologías de conectividad (guiones de llamada XML)

### 7.2 Definición de problemas:

**Conjunto de caracteres recomendado:** Existen algunas restricciones básicas respecto de los caracteres que admitirá FileMaker en los nombres de guiones. Además, cuando se llama a los guiones desde una llamada externa, como por ejemplo XML, los caracteres utilizados pueden presentar problemas. El desarrollador debe estar al tanto de las restricciones tanto de FileMaker como de cualquier sistema externo con el cual interactuarán sus soluciones.

**Convención de mayúsculas/minúsculas recomendada:** La recomendación, en este caso, no consiste en superar un problema específico sino en aplicar un criterio coherente para adoptar la manera de dar nombres a los archivos. El objetivo es seleccionar un método y ser coherente al respecto.

**Modelo coherente para documentar un guión:** La convención debe ofrecer un modelo para la documentación de los diversos elementos de un guión. Éstos deben incluir los elementos, el formato y la ubicación.

### 7.3 Pautas "en cumplimiento de las normas" (Guiones):

#### 7.3.1 Nombres de guiones

1. Los Nombres de guiones sólo deben utilizar los siguientes caracteres:
  - Mayúsculas y minúsculas aA - zZ
  - Número 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble " \_ " " \_\_ "
2. No pueden exceder 100 caracteres de longitud.



### 7.3.2 Variables

1. Deben comenzar con \$ para local, \$\$ para global.
2. No deben contener espacios.
3. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Variables) \$, \$\$

Minúsculas y CamelCase	nombreDeMiVariable
Mayúsculas y CamelCase	NombreDeMiVariable
Subrayado simple (minúsculas)	nombre_de_mi_variable
Subrayado simple (Tipo Título)	Nombre_de_Mi_Variable
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_VARIABLE

### 7.3.3 Documentación de guiones

El paso del Comentario en un guión se debe utilizar para almacenar la documentación. El comentario debe ser el primer paso en el guión. Los comentarios deben incluir cada uno de los componentes que se enumeran a continuación y se les dará el formato que muestra el ejemplo. Si cualquier comentario excede el límite de 30K de caracteres, se debe agregar un paso adicional de comentarios en el guión.

1. Nombre: Nombre del guión
2. Historial: (podría incluir cualquiera de los siguientes elementos u otros que se consideren necesarios)
  - Nombre del desarrollador: Nombre del desarrollador del guión
  - E-mail del desarrollador: Correo electrónico del desarrollador del guión
  - Nombre del modificador: Nombre del desarrollador del guión que lo modificó por última vez.
  - E-mail del modificador: Correo electrónico del desarrollador que modificó por última vez el guión.
  - Fecha de la creación: Fecha en formato DD–MMM–YYYY en que se creó el guión.
  - Fecha de la última modificación: Fecha en formato DD–MMM–YYYY en que se modificó el guión por última vez.
3. Objeto: Descripción de aquello para lo cual se ha diseñado el guión.
4. Variables declaradas: Enumera todas las variables que se declaran en el guión.
5. Variables de referencia: Enumera todas las variables a las cuales el guión hará referencia, en la misma forma que las variables globales previamente declaradas.
6. Parámetros: Enumera todos los parámetros requeridos por el guión.
7. Notas importantes. Información adicional.

Se incluye un ejemplo en la figura 12.



## Especificar

**Nombre:**

Mi Nombre de Guión

**Historial:**

Creado por Thomas K

Fecha de creación: 01-ENE-2005

Fecha de modificación: 01-ENE-2007

**Objeto:**

Es un ejemplo de función que muestra el formato de la documentación Si el renglón se ajusta, debe permitirse

**Variables declaradas:**

`$variable uno`: enumera el objeto de esta variable. Si el renglón se ajusta, debe permitirse hasta el renglón siguiente sin sangría.

`$variable dos`: enumera el objeto de esta variable. Si el renglón se ajusta, debe permitirse hasta el renglón siguiente sin sangría.

**Variables de referencia:**

`$$variable tres`: enumera el objeto de esta variable. Si el renglón se ajusta, debe permitirse hasta el renglón siguiente sin sangría.

`$$variable cuatro`: enumera el objeto de esta variable. Si el renglón se ajusta, debe permitirse hasta el renglón siguiente sin sangría.

**Parámetros:**

parámetro uno: Ejemplo de explicación de parámetro. Esto debe incluir todo detalle pertinente.

parámetro dos: Ejemplo de explicación de parámetro. Esto debe incluir todo detalle pertinente.

**Resultados:**

Provee información correspondiente a cualquier salida o Resultado producido por el guión.

**Notas Importantes:**

Provee toda la información adicional que constituirá conocimiento útil para los creadores de soluciones del futuro y como referencia

Cancelar

Figura 12



## 8 ▷ Cálculos

La claridad de una fórmula de cálculo aumenta a través del formato y de los comentarios. Es responsabilidad del desarrollador ofrecer un formato significativo. Los buenos comentarios deben explicar el "porqué" más que el "cómo". Además, los buenos comentarios deben ofrecer la comprensión de todo código que no sea autodescriptivo. En algunos casos, las mejores técnicas de codificación pueden proveer una comprensión más clara. Por ejemplo, utilizar la función Let() (Permitir) en todos los cálculos, aún en los más simples, puede ayudar drásticamente a mejorar la comprensión y legibilidad del código.

La convención recomendada está diseñada para ofrecer un enfoque coherente del formato y de los comentarios de los cálculos. Los beneficios del formato y de la documentación no son obvios en cálculos simples, no obstante, a medida que los cálculos se tornan más complejos, se transforman en una herramienta útil para desglosarlos y documentarlos para futura referencia, para el desarrollador de la solución o para el sucesor.

### 8.1 Objetivos:

- ✓ Modelo coherente para variables utilizadas dentro de los cálculos.
- ✓ Modelo coherente para documentar cálculos.
- ✓ Categorías de comentarios recomendadas: Provee ejemplos de espacios y formatos para resolver el problema de codificación por color para varios componentes, como por ejemplo: funciones, cadenas, campos, Ocurrencias de tablas, operadores y funciones personalizadas.

### 8.2 Definición de problemas

**Documentación de los cálculos:** Muchos desarrolladores hacen diferentes comentarios en diferente medida. Algunos se apoyan en un criterio minimalista mientras que otros hacen más comentarios. En general, todos concuerdan en que cierta documentación es buena, especialmente con cálculos complejos. Las recomendaciones en cumplimiento de las normas describen una cantidad mínima de información así como también una ubicación y formato. Esto se puede ampliar pero debe contener los componentes mínimos.

**Formato de los cálculos:** El formato apropiado de los cálculos provee mejor legibilidad y ayuda a comprender los componentes individuales. Además, facilita al desarrollador y a sus colegas comprender su sentido. Dado que el motor de cálculo no da formato automáticamente a los cálculos, proveer dicho formato es responsabilidad del desarrollador. Se decidió que las FDC ofrezcan orientación general pero no que recomienden específicamente formato alguno.

### 8.3 Pautas "en cumplimiento de las normas" (Cálculos):

#### 8.3.1 Variables

1. Deben comenzar con \$ para local, \$\$ para global.
2. No deben contener espacios.
3. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Variables) \$,\$\$ :

Minúsculas y CamelCase	\$nombreDeMiVariable
Mayúsculas y CamelCase	\$NombreDeMiVariable
Subrayado simple (minúsculas)	\$nombre_de_mi_variable
Subrayado simple (Tipo Título)	\$Nombre_de_Mi_Variable
Subrayado simple (MAYÚSCULAS)	\$NOMBRE_DE_MI_VARIABLE



### 8.3.2 Comentarios de la cabecera del bloque de cálculo

Cada cálculo comentado debe contener una cabecera de bloque con las siguientes secciones:

- Objeto: provee una descripción general de aquello para lo cual el cálculo está diseñado.  
Dependencias: provee información sobre los elementos de los cuales el cálculo puede depender.
- Los ejemplos incluyen campos, funciones personalizadas y variables globales.
- Historial: aporta un marcador para cualquier historia correspondiente al cálculo. Esto puede incluir el desarrollador, la fecha de creación e información sobre versiones.

Ejemplo:

```
/*
```

*Objeto:*

*Este cálculo se utiliza como ejemplo del formato de un campo de cálculo y del comentario apropiado para éste. El comentario del objeto aparecerá en la lista de campos.*

*Dependencias:*

*Campo de Texto: Campo de texto utilizado por este cálculo*

*Campo numérico: Campo numérico utilizado como contador*

*Mi función personalizada\_\_CFpub Función personalizada utilizada en este cálculo*

*Historial:*

*Creado por Thomas K.*

*Fecha 1/1/2007*

*Modificado: 1/2/2007 para la versión x de la solución*

```
*/
```

### 8.3.3 Comentarios de cálculos en el renglón

Los cálculos se comentarán en la sección comentarios, no a través del cálculo. Existe una excepción que es crear variables con la función 'Let' (Permitir). A menos que el nombre de la variable evidencie su objeto, el desarrollador debe hacer comentarios acerca de la variable. Si no hubiera espacio suficiente para comentar la variable con posterioridad a su creación, se dejará una sangría en el renglón siguiente y se consignará el comentario después de ésta.

Se deben utilizar las sangrías libremente para mejorar la legibilidad. Muchas funciones se prestan para las sangrías tal como se demuestra con las funciones Let (Permitir) y case (mayúsculas y/o minúsculas) en el siguiente ejemplo: Esto permite que las pares se separen rápidamente, por ejemplo el cuerpo del cálculo (la instrucción de mayúsculas y/o minúsculas) de la creación de la variable. (Ver el punto [Formato del cálculo](#) donde se analiza este tema).

```
Let(  
  [  
    $Texto = Campo de texto1;  
    // Poner el valor de un campo en una variable de modo que sea llamada una vez  
    $Variable1 = expresión1;  
    // El valor de la primera expresión  
    $Variable2 = Funcion_personalizada_1_CFpub ($Texto)  
    // Renglon de comentario opcional utilizado si el cálculo es demasiado largo para poner  
    // el comentario después de éste..  
  ];  
  Case(  
    prueba1; resultado1;  
    prueba2; resultado2;  
    ResultadoPredeterminado )  
)
```



Las FDC reconocen que algunos campos de cálculo son extremadamente simples y que sus objetivos son evidentes de por sí. En dichos cálculos, el comentario objetivo es valioso para documentar la base de datos aunque no son necesarios posteriores comentarios.

### 8.3.4 Formato de cálculos

El formato apropiado de los cálculos provee mejor legibilidad y ayuda a comprender los componentes individuales. Además, facilita al desarrollador y a sus colegas comprender su sentido. Dado que el motor de cálculo no da formato automáticamente a los cálculos, proveer dicho formato es responsabilidad del desarrollador. Se decidió que las FDC ofrezcan orientación general pero no que recomienden específicamente formato alguno.

#### 8.3.4.1 EJEMPLO DE FORMATO #1

El siguiente cálculo efectúa cierto tipo de sustitución. Sin embargo, leer una cadena larga de texto dificulta saber si la función de `Substituir` es la función que inicia y termina este cálculo. Con el formato, se puede decir rápidamente que la función `Substituir` inicia y termina este cálculo, con un carácter de retorno de carro añadido al final.

```
Substitute(DATOS::zz_renglon_en_uso; Left(DATOS::zz_renglon_en_uso; Position(DATOS::
zz_renglon_en_uso; "/" ; 1 ; 1) - 1); Middle(DATOS::zz_renglon_en_uso; Position(DATOS::
zz_renglon_en_uso; "/" ; 1;
1) + 1; Length(DATOS::zz_renglon_en_uso) - 1))
& "¶"
```

Al ver este cálculo con formato, se recuerda que la función `Substituir` requiere tres parámetros:

```
Substitute ( texto ; textoBuscado ; textoDeSustitución )
```

```
Substitute (
    DATOS::zz_renglon_en_uso ;
    Left (
        DATOS::zz_renglon_en_uso;
        Position ( DATOS::zz_renglon_en_uso ; "/" ; 1 ; 1 ) - 1
    ) ;
    Middle (
        DATOS::zz_renglon_en_uso ;
        Position ( DATOS::zz_renglon_en_uso ; "/" ; 1 ; 1 ) + 1 ;
        Length ( DATOS::zz_renglon_en_uso ) - 1
    )
) & "¶"
```

Pero cuando las otras funciones se dividen también, esto facilita comprender donde están esos parámetros. Ver la Figura 13 a continuación.



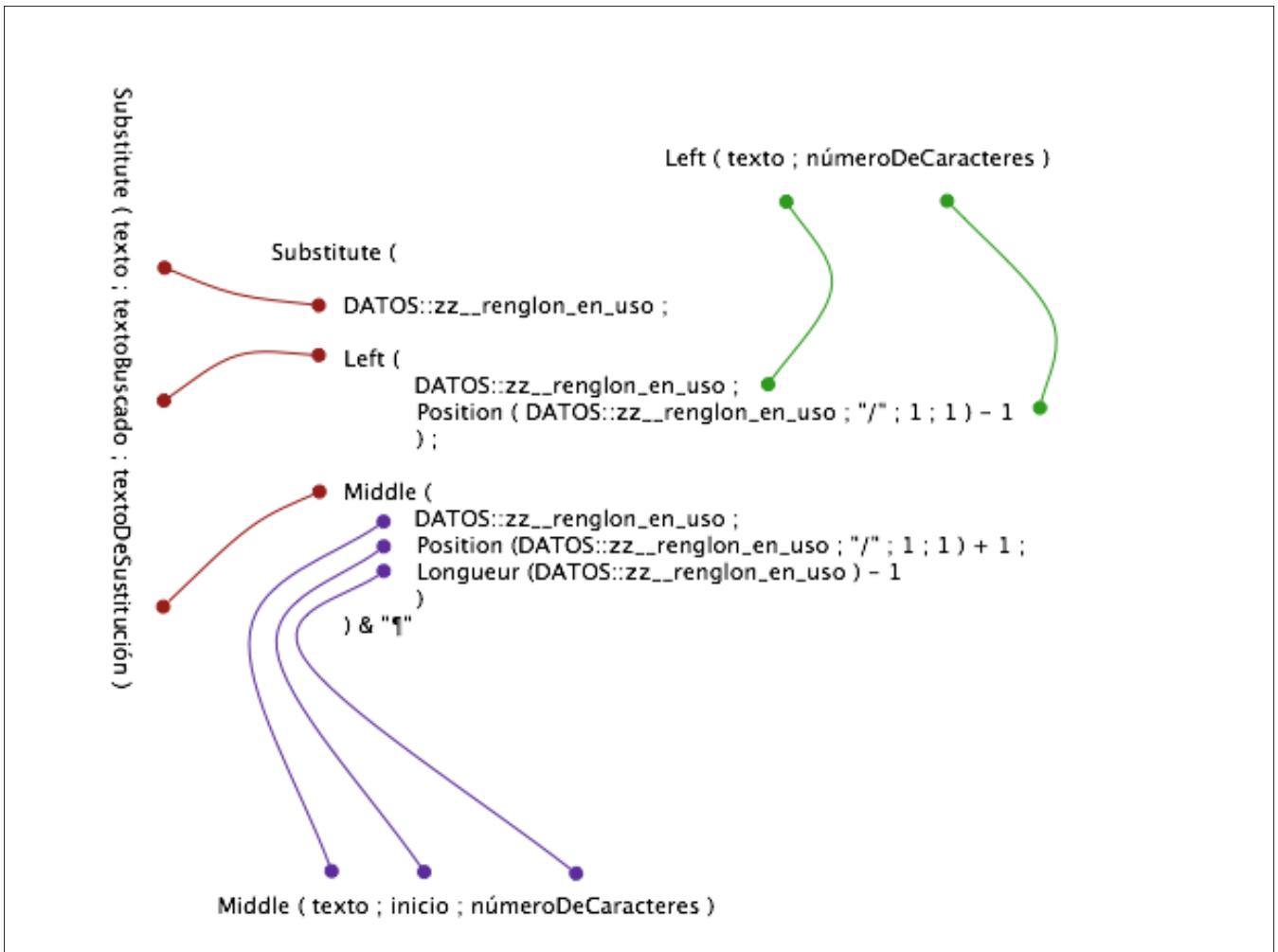


Figura 13

Al separar cada parámetro se puede ver de inmediato cuánto más fácil es distinguir las funciones.

### 8.3.4.2 EJEMPLO DE FORMATO #2

El principio general es agregar espacio en blanco y saltos de renglón a los cálculos en forma tal que sea claro donde comienza y termina cada función y cada argumento de función. Si cualquier llamada de función es demasiado larga para ajustarse a un renglón, se desglosa de modo tal que:

- El nombre de la función y el paréntesis inicial son seguidos por un nuevo renglón.
- Cada argumento de la función se inicia en un renglón nuevo y se le da más sangría que al nombre de la función misma.
- El paréntesis final inicia un renglón nuevo, con una sangría al mismo nivel que el nombre de la función.



```

If(
  Position( menu_seleccion; "Fruta alternativa"; 0; 1 );
  Case(
    DayName( Get( CurrentDate ) ) = "Martes";
    "Naranjas";
    DayName( Get( CurrentDate ) ) = "Jueves";
    "Manzanas";
    /* de otra manera */
    "Bananas"
  );
  "Peras"
) & " and " &
If(
  Position( menu_seleccion; "Vegetal alternativo"; 0; 1 );
  Case(
    DayName( Get( CurrentDate ) ) = "Jueves";
    "Guisantes";
    /* de otra manera */
    "Zanahorias"
  );
  "Maíz"
)

```

Se debe tener en cuenta que el cuadro de diálogo de los cálculos del FileMaker Pro no utiliza una fuente de anchura fija. Por ello, es necesario comenzar el argumento de la primera función en un renglón nuevo, antes que poner tanto la llamada de la función y su primer argumento en el mismo renglón. Por ejemplo, el siguiente código aunque es más compacto que el anterior, es menos efectivo cuando se exhibe con una fuente de anchura variable.

```

If( Position( menu_seleccion; "Fruta alternativa"; 0; 1 );
  Case( DayName( Get( CurrentDate ) ) = "Martes";
    "Naranjas";
    DayName( Get( CurrentDate ) ) = "Jueves";
    "Manzanas";
    /* de otra manera */
    "Bananas"
  );
  "Peras"
) & " and " &
If( Position( menu_seleccion; "Vegetal alternativo"; 0; 1 );
  Case( DayName( Get( CurrentDate ) ) = "Jueves";
    "Guisantes";
    /* de otra manera */
    "Zanahorias"
  );
  "Maíz"
)

```

Se debe utilizar el formato en que la coincidencia de paréntesis sea más obvia. De este forma, la persona que da el formato al cálculo siempre comienza cerrando los paréntesis en un renglón nuevo cuando una función se divide en renglones múltiples, antes que devolver algo como lo siguiente.



```

If(
  Position( menu_seleccion; "Fruta alternativa"; 0; 1 );
  Case(
    DayName( Get( CurrentDate ) ) = "Martes";
    "Naranjas";
    DayName( Get( CurrentDate ) ) = "Jueves";
    "Manzanas";
    /* de otra manera */
    "Bananas" );
  "Peras") & " and " &
If(
  Position( menu_seleccion; "Vegetal alternativo"; 0; 1 );
  Case(
    DayName( Get( CurrentDate ) ) = "Jueves";
    "Guisantes";
    /* de otra manera */
    "Zanahorias" );
  "Maíz")

```

Se pueden aplicar sangrías adicionales en los renglones que sean resultado del "ajuste de renglón".

```

Right(
  Middle( 10 ^ 11 + Round( Abs( Cant ); Precision ); 1; 3 ) & «,» &
  Middle( 10 ^ 11 + Round( Abs( Cant ); Precision ); 4; 3 );
  Length( Int( Round( Abs( Cant ); Precision ) ) ) +
  If( Precision > 0; 1 + Precision; 0 ) +
  Int(
    ( Length( Int( Round( Abs( Cant ); Precision ) ) ) - 1 ) / 3
  )
)

```

Sin sangría por el ajuste de renglón el resultado sería:

```

Right(
  Middle( 10 ^ 11 + Round( Abs( Cant ); Precision ); 1; 3 ) & «,» &
  Middle( 10 ^ 11 + Round( Abs( Cant ); Precision ); 4; 3 );
  Length( Int( Round( Abs( Cant ); Precision ) ) ) +
  If( Precision > 0; 1 + Precision; 0 ) +
  Int(
    ( Length( Int( Round( Abs( Cant ); Precision ) ) ) - 1 ) / 3
  )
)

```

## 8.4 Consideraciones complementarias

Herramienta de formato de cálculo automática de FileMaker: Debi Fuchs de Aptworks Consulting creó una "Herramienta de formato de cálculo" que hace la mayor parte de este trabajo. Se puede probar en <http://www.aptworks.com/tools>. La herramienta de formato no ofrece sangrías adicionales en los renglones que sean resultado del "ajuste de renglón". El desarrollador puede aumentar el formato de sus cálculos agregando sus propias sangrías de ajuste si lo desea.



## 9 ▷ Listas de valores

El efecto de las convenciones en las listas de valores es algo limitado. Sin embargo, algunas pautas generales pueden ayudar a eliminar problemas comunes, ofrecer extensibilidad y organizar. Se debe considerar sólo el uso de aquellos caracteres que no acarrearán problemas con tecnologías externas, como por ejemplo XML que pueden remitir a nombres de listas de valores. Se debe elegir específicamente mayúsculas y/o minúsculas y utilizarlo en forma coherente.

Algunos desarrolladores incluyen metadatos para incluir comentarios o identificar un tipo de lista. Si la práctica de desarrollo apunta en esta dirección se debe tener en cuenta donde colocar estos metadatos y cuál es la forma más conveniente para éstos. Colocar los metadatos al final del nombre presenta ciertas ventajas. Esta colocación permite que se utilice la clasificación alfabética para los nombres de la lista, permite que los nombres de lista se basen en el tipo y no en los metadatos, y presenta la lista en un formato esperado. La otra consideración es la forma que estos metadatos deben tener. Se decidió que las FDC ofrezcan un marcador que defina cierta notación básica, pero que también permita al desarrollador extensibilidad dejando que este componente de la sintaxis sea definido por él según sus necesidades específicas. Toda extensión será documentada en la sección de adhesión de la solución (Ver la sección de [Adhesión](#) de este documento donde se incluye más información).

### 9.1 Objetivos:

- ✓ Conjunto de caracteres recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada
- ✓ Separación sintáctica recomendada
- ✓ Pautas de longitud
- ✓ Pautas para singular vs. plural
- ✓ Criterio para aplicar los Comentarios de la lista de valores
- ✓ Enfoque para manejar los Metadatos (Fuente/Tipo) fuera de "Definir la lista de valores"
- ✓ Criterio para encarar la imposibilidad de organizar listas de valores que no sea manualmente en orden secuencial.
- ✓ Nota: Cuestiones relacionadas con actualizar listas de valores personalizadas independientemente de los datos o esquema del archivo.
- ✓ Los cálculos se enlazan al nombre antes que al ID interno de la lista de valores.

### 9.2 Definición de problemas:

**Conjunto de caracteres recomendado:** El efecto de caracteres impropios en listas de valores es mínimo. Sin embargo, se debe tener en cuenta lo siguiente. Los desarrolladores deben tener cuidado y utilizar caracteres que no entren en conflicto con los cálculos del FileMaker que pueden usarse para interactuar con los nombres de las listas de valores. Además, los desarrolladores tener en cuenta todos los caracteres que pueden presentar problemas con las consultas XML. Restringir el conjunto de caracteres a las recomendaciones enumeradas aquí puede ayudar a limitar la exposición a este tipo de conflictos.

**Convención de mayúsculas/minúsculas recomendada:** La decisión de cómo separar las palabras dentro de las listas de valores incidirá en la coherencia de conjunto en todas las convenciones. La convención recomendada sugiere que se utilice sólo uno de los siguientes estilos: minúsculas y Camel case, mayúscula y Camel case, Subrayados, y la convención de todas letras mayúsculas para los nombres de las listas de valores.

**Separación sintáctica recomendada:** En el supuesto de que el desarrollador desee incluir algunos metadatos acerca de una lista de valores, aunque sea con un alcance limitado, es necesario que los desarrolladores actuales y futuros así como también el usuario final comprendan claramente qué separa el nombre de la lista de valores de los metadatos o de la notación. La selección de caracteres para separar los metadatos y el nombre de la lista de valores debe ser diferente de la que separa las palabras individuales dentro del nombre de la lista de valores.



**Longitud del nombre de la lista de valores:** Una lista de valores FileMaker puede tener hasta 100 caracteres. Sin embargo, en muchos casos éste no resulta un límite práctico. Con el lanzamiento del FileMaker 8, todos los cuadros de diálogo de Macintosh no presentarán dificultad alguna para exhibir el nombre completo. En la plataforma Windows, existen unas pocas áreas donde un nombre de lista de valores con más de 30 caracteres de largo no será exhibido. No obstante, los nombres completos son accesibles navegando en el cuadro de diálogo

“Definir ... lista de valores”. Esta decisión respecto de la longitud realmente es una cuestión de elección antes que una restricción específica.

**Pautas para singular vs. Plural:** La decisión de utilizar nombres en singular o colectivos en lugar de la alternativa en plural se incluye sólo con el fin de sugerir al desarrollador que adopte una decisión respecto de cuál utilizará y que sea coherente con dicha decisión.

### 9.3 Pautas “en cumplimiento de las normas” (Listas de valores):

1. Sólo deben utilizarse los siguientes caracteres:
  - Mayúsculas y minúsculas aA - zZ
  - Número 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble “\_” “\_\_”
2. NO deben contener espacios.
3. Se debe ser coherente con el uso de nombres en singular o plural.
4. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Listas de valores) :

Minúsculas y CamelCase	nombreDeMiListaDeValores
Mayúsculas y CamelCase	NombreDeMiListaDeValores
Subrayado simple (minúsculas)	nombre_de_mi_lista_de_valores
Subrayado simple (Tipo Título)	Nombre_de_Mi_Lista_De_Valores
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_LISTA_DE_VALORES

5. Sintaxis:

ValueListName[\_\_\_]{suffix}

(Ver el punto [Leyenda de sintaxis](#) al finalizar este documento donde se describe la sintaxis).

Explicación de la Sintaxis: La sintaxis de las listas de valores está diseñada para crear un enfoque coherente de las listas de valores que enfrente la mayoría de los problemas identificados. El uso de un sufijo en lugar de un prefijo permite mayor flexibilidad en los nombres de las listas de valores que se ajustan al estilo de los desarrolladores mientras que a su vez mantiene una cierta coherencia básica. Provee los metadatos básicos. Permite la agrupación de listas de valores por función.

- Nombre de la lista de valores:  
Se respetarán las recomendaciones descriptas precedentemente seguidas por el subrayado doble “\_\_”.
- “\_\_” – obligatorio  
El subrayado doble actúa como separador entre el nombre de la lista de valores y el sufijo.
- Sufijo – opcional definido por el desarrollador (se proveen recomendaciones de referencia)  
Podría ser uno de los elementos enumerados en la siguiente tabla. Esto se puede ampliar para cumplir con cualquier necesidad específica. Cualquier extensión debe tener la forma de un sufijo para que se pueda continuar ordenando los elementos de la lista de valores de acuerdo con el nombre y para que se permita que los nombres de la lista se basen en el tipo de funcionalidad antes que en los prefijos de metadatos. Cualquier extensión debe ser documentada en la sección de adhesión de la solución. (Ver la sección de [Adhesión](#) de este documento dónde se incluye más información). En las FDC se recomienda lo siguiente como requerimiento básico.



### Sufijos de tipo recomendados como referencia de listas de valores

- c** Valor personalizado
- x** Lista de valores de otro archivo
- d** Lista de valores de campo, incluyendo todos los valores (para Dinámicos)
- r** Lista de valores de campo, incluyendo los valores relacionados (para Relacionados)



## 10 ▷ Cuentas y seguridad

Las cuentas del FileMaker se pueden configurar para autenticarlas en forma interna o externa. Las cuentas internas identifican únicamente a un usuario específico y se gestionan totalmente dentro del FileMaker. Las cuentas autenticadas externamente identifican únicamente un grupo externo que puede estar integrado por uno o más usuarios. La pertenencia a Cuentas Externas se gestiona en forma externa al FileMaker. FileMaker utiliza "conjuntos de privilegios" para determinar las posibilidades o derechos que una cuenta autenticada tiene respecto de la base de datos. La Figura 14 ilustra la relación entre estos componentes.

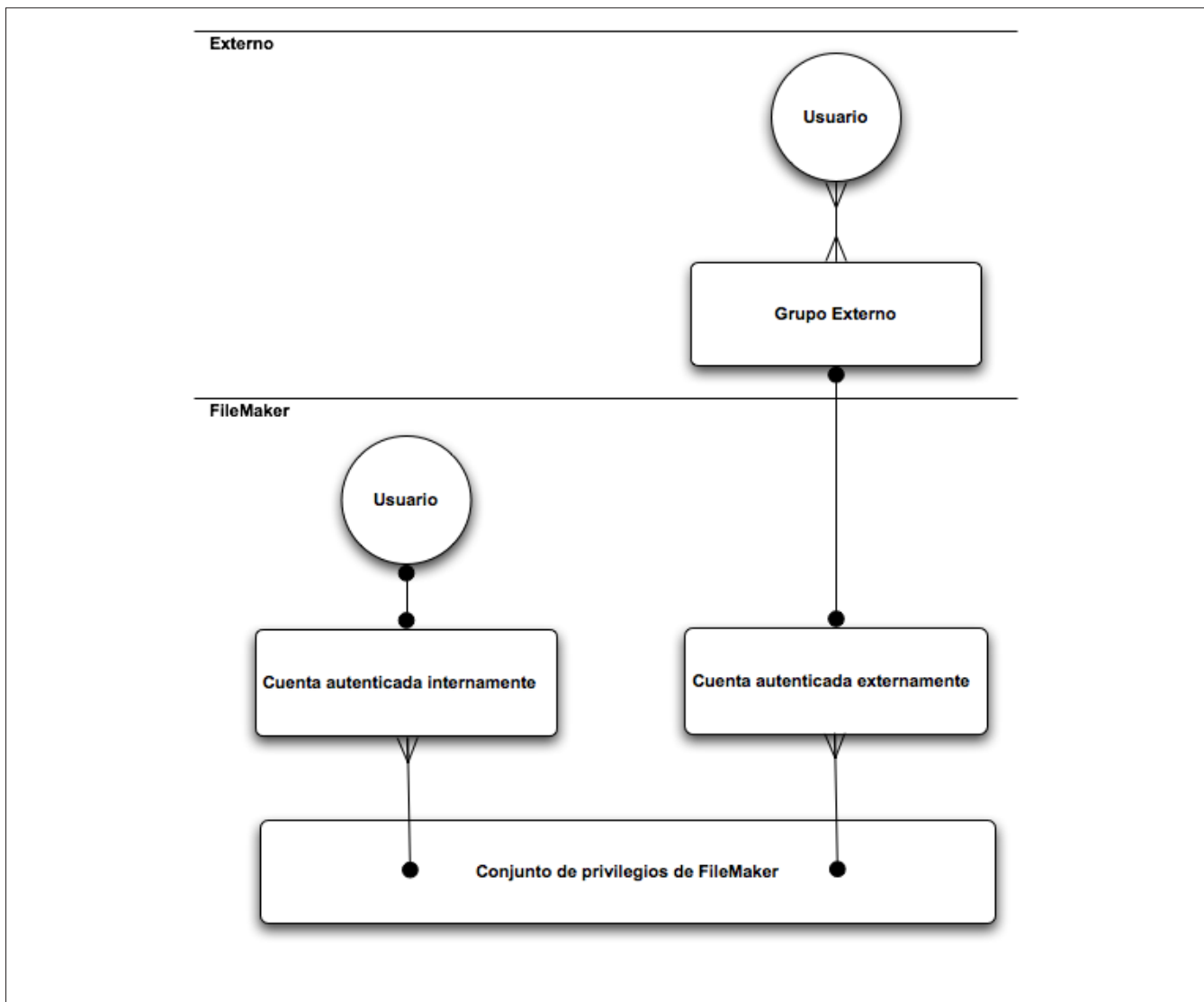


Figura 14

El conjunto de privilegios define las posibilidades que un usuario o grupo de usuarios tiene dentro del archivo. Esta es su función. Por ejemplo, si existen un desarrollador, administrador, sólo lector, persona que ingresa los datos y varias otras cuentas. El nombre debe representar apropiadamente la función. Este hecho es especialmente importante cuando la administración está 'afuera' de FileMaker o es externa a este producto.

Una de las ventajas clave de autenticación externa es la posibilidad de aprovechar la infraestructura de seguridad de la organización. Esto ofrece un lugar único para gestionar el acceso a todos los recursos disponibles para cualquier usuario dentro del alcance de la organización. También significa que los administradores de TI, que no están familiarizados con los puntos específicos de la solución o de todas las soluciones dentro del entorno, deberán asignar membrecías por nombre. Será necesario articular claramente el archivo o archivos / la solución y la función de asignar pertenencias teniendo esto en consideración; debe existir cierta coherencia para los conjuntos de privilegios y las cuentas autenticadas externamente.



Es necesario incluir una combinación de componentes en el nombre de la cuenta/ grupo autenticada externamente para identificarlo. Como mínimo debe contener el nombre de la solución, para identificar el grupo como asociado con una solución particular. También debe contener la función para esa solución. Por ejemplo, en el supuesto de existir una solución denominada "Rastreador de Ventas". Contiene tres conjuntos de privilegios: Operadores, Ventas Norteamericanas, y Ventas Globales. En el supuesto de que exista otra solución denominada "Rastreador de embarques" que contenga los mismos conjuntos de privilegios. Dejando de lado la posibilidad de que cada función planifica los mismos usuarios para cada solución, lo cual no será el caso en muchas situaciones. Será necesario especificar la solución y la función que se asignará al usuario. Esto exige una convención de nombres que use tanto el conjunto de privilegios como el nombre de la solución para lograr el reconocimiento externo.

## 10.1 Objetivos:

- ✓ Conjunto de caracteres recomendado
- ✓ Convención de mayúsculas/minúsculas recomendada
- ✓ Separación sintáctica recomendada
- ✓ Pautas de longitud
- ✓ Permitir el fácil reconocimiento del nombre de asociación del grupo externo a la solución específica
- ✓ Permitir el fácil reconocimiento de los privilegios por nombre del grupo externo respecto de la solución específica
- ✓ Aplicar soluciones duplicadas en servidores separados cuando se utilice autenticación externa.

## 10.2 Definición de problemas

**Conjunto de caracteres recomendado:** Existen algunas restricciones básicas respecto de los caracteres que admitirán FileMaker y el sistema operativo que autentica. Para evitar las restricciones de caracteres, la recomendación general es utilizar mayúsculas y minúsculas, caracteres ACSII en minúsculas y números.

**Convención de mayúsculas/minúsculas recomendada:** La elección efectuada afecta otras áreas donde se aplica una convención. La decisión, en este caso, no consiste en superar un problema específico sino en aplicar un criterio coherente en la forma de utilizar mayúsculas y/o minúsculas a través de todas las demás convenciones. El objetivo es seleccionar un método y ser coherente al respecto.

**Separación sintáctica recomendada:** La separación sintáctica para asignar nombres a las cuentas se refiere a la forma de diferenciar la sintaxis de los nombres. Es habitual ver prefijos o sufijos utilizados para indicar que un archivo es parte de un grupo de archivos. También se utilizan para contener características de identificación. Se busca una forma coherente de diferenciar estos metadatos principales/continuos del nombre de la cuenta.

**Pautas de longitud:** Los conjuntos de privilegios y los nombres de cuentas se limitan a 100 caracteres. El Windows Server 2003 se limita a nombres de grupo de 64 caracteres. El Macintosh OS X Server admitirá nombres de grupo de 100 caracteres.

**Reconocimiento de la solución:** Específico de la autenticación externa: Los grupos creados aparte del entorno de desarrollo deben coincidir exactamente con los nombres creados dentro del entorno de desarrollo de los archivos FileMaker Pro. Teniendo en cuenta que muchas organizaciones tendrán una cantidad indefinida de soluciones, puede ser molesto encontrar los grupos pertinentes a una solución específica. Incluir el nombre de la solución dentro del nombre del grupo mitiga este problema.

**Reconocimiento de Privilegios (Función):** Específico de la autenticación externa: Los grupos creados fuera del entorno de desarrollo deben coincidir exactamente con los nombres creados dentro del entorno de desarrollo. Teniendo en cuenta que muchas organizaciones tendrán una cantidad indefinida de soluciones, puede ser molesto navegar a través de varios nombres de grupos para encontrar aquellos pertinentes a una solución específica. Además, también es necesario ubicar la función para esa solución específica. Incluir el nombre de la función mitiga este problema.



**Soluciones duplicadas en servidores separados:** Existen situaciones en que múltiples servidores alojan la "misma" solución. Son esencialmente copias exactas pero sirven a diferentes conjuntos de usuarios. Se deben ampliar las recomendaciones generales para manejar esta situación especial y en cierto modo extraña.

## 10.3 Pautas "en cumplimiento de las normas" (Seguridad):

### 10.3.1 Conjuntos de privilegios

- Sólo deben utilizarse los siguientes caracteres:
  - Mayúsculas y minúsculas aA - zZ
  - Número 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble " \_ " \_ "
- No deben contener espacios.
- Debe utilizar en forma coherente nombres en singular o en plural.
- No pueden exceder 100 caracteres de longitud. (Al utilizar las pautas recomendadas para cuentas autenticadas externamente, la longitud debe permitir la sintaxis completa que no puede exceder 64 caracteres).
- Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Conjuntos de privilegios) :

Minúsculas y CamelCase	nombreDeMiConjuntoDePrivilegios
Mayúsculas y CamelCase	NombreDeMiConjuntoDePrivilegios
Subrayado simple (minúsculas)	nombre_de_mi_conjunto_de_privilegios
Subrayado simple (Tipo Titulo)	Nombre_De_Mi_Conjunto_De_Privilegios
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_MI_CONJUNTO_DE_PRIVILEGIOS

### 10.3.2 Nombres de cuentas autenticadas internamente

- Sólo deben utilizarse los siguientes caracteres:
  - Mayúsculas y minúsculas aA - zZ
  - Número 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble " \_ " \_ "
- No deben contener espacios.
- Debe utilizar en forma coherente nombres en singular o en plural.
- No pueden exceder 100 caracteres de longitud.
- Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Cuentas autenticadas internamente) :

Minúsculas y CamelCase	nombreDeCuenta
Mayúsculas y CamelCase	NombreDeCuenta
Subrayado simple (minúsculas)	nombre_de_cuenta
Subrayado simple (Tipo Titulo)	Nombre_De_Cuenta
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_CUENTA



### 10.3.3 Nombres de cuentas autenticadas externamente (Nombres de grupo)

1. Sólo deben utilizarse los siguientes caracteres:
  - Mayúsculas y minúsculas aA - zZ
  - Número 0,1,2,3,4,5,6,7,8,9
  - Subrayado simple y doble " \_ " \_ "
2. No deben contener espacios.
3. Debe utilizar en forma coherente nombres en singular o en plural.
4. No deben tener más de 64 caracteres de longitud para poder trabajar en las plataformas de Windows y OS X.
5. Las palabras individuales deben separarse en forma coherente, utilizando uno de los siguientes métodos:

#### Métodos de separación de palabras recomendados (Cuentas autenticadas internamente) :

Minúsculas y CamelCase	nombreDeCuenta
Mayúsculas y CamelCase	NombreDeCuenta
Subrayado simple (minúsculas)	nombre_de_cuenta
Subrayado simple (Tipo Título)	Nombre_De_Cuenta
Subrayado simple (MAYÚSCULAS)	NOMBRE_DE_CUENTA

#### 6. Sintaxis:

<<FileMakerGroupDesignation>>[\_]<SolutionName>[\_]<PrivilegeSetName>

La sintaxis está diseñada para ofrecer una forma coherente de asignar nombres a grupos autenticados externamente. La convención ayuda a gestionar grupos externos al proveer una convención que identifica el grupo como un «Grupo de seguridad de FileMaker», una solución / un archivo específico, y la función o conjunto de privilegios. Con estos 3 elementos, la gestión de grupos se simplifica y se organiza bien. Se utiliza " \_ " (subrayado doble) para separar cada una de las secciones. Estos caracteres son compatibles con el AD (Directorio Activo) anterior y posterior a Windows 2000. También se admiten caracteres de OD (Directorio Abierto). Con anterioridad al producto Windows 2000, los nombres de los grupos no pueden contener los siguientes signos: ; | = + , \* ? < > " / \ [ ] .

NOTA: Habrá instancias en que las normas organizativas puedan dictar los nombres de grupo. En tales casos, se utilizarán los conocimientos del presente documento para adaptar la convención en la forma necesaria.

- Designación del grupo FileMaker: obligatorio; definida por el desarrollador  
Identifica el grupo como un grupo de seguridad FileMaker. Esto agrupa a los grupos de seguridad FileMaker juntos dentro de la herramienta de gestión de grupos de trabajo. La recomendación es utilizar "fm" en minúsculas. La selección específica es arbitraria pero se recomienda por razones de coherencia.
- Nombre de la solución:  
Identifica la solución / archivo específico con el cual el grupo está asociado. Esto continuará admitiendo la clasificación correcta al ordenar adicionalmente todos los grupos de una solución en particular juntos. El nombre debe coincidir con la selección efectuada para el archivo principal o archivos de punto de entrada sin la extensión .fpX.
- Nombre del conjunto de privilegios: obligatorio, utiliza el nombre del conjunto de privilegios:  
Identifica la función o posibilidades que los miembros del grupo tienen dentro de una solución. Incluir este elemento en el nombre de la cuenta / del grupo provee a la gestión externa la fácil identificación del grupo que debe ser modificado.



- Conjuntos de privilegios especiales: El producto FileMaker tiene 3 conjuntos de privilegios predefinidos:
  - ▶ [Full Access] (Acceso total),
  - ▶ [Data Entry Only] (Sólo entrada de datos),
  - ▶ [Read-Only Access] (Acceso de sólo lectura).

Estos conjuntos de privilegios no se pueden modificar, renombrar ni eliminar. Al utilizar cualquiera de estos conjuntos de privilegios, se deben abandonar los corchetes [ ] del nombre del conjunto de privilegios cuando se configura la cuenta del grupo autenticado externamente. Windows prohíbe estos caracteres en los nombres de grupo previos al Windows 2000.

[Sólo lectura] = Sólo\_Lectura, SóloLectura, sóloLectura, SOLO\_LECTURA

Ejemplo: fm\_Mi Solución\_\_Sólo\_Lectura.

7. Acceso completo con cuentas autenticadas externamente: *En Seguridad de FileMaker: El Libro* de Steven H. Blackwell, él comenta el uso de los conjuntos de privilegios de Acceso Completo con cuentas autenticadas en forma externa. Escribió: "Recomiendo enfáticamente que el desarrollador nunca configure una Cuenta con el Conjunto de privilegios de Acceso Completo predeterminado que esté sujeta a la Autenticación externa. Se debe autenticar dicha Cuenta mediante el método interno FileMaker Pro solamente. Primero, todos los archivos tienen al menos una Cuenta de Acceso Completo autenticada internamente. Segundo, la versión inicial de FileMaker Server 7 y FileMaker Server 7 Advanced no admiten el sistema de ID Global Universal (GUID). Si un pirata informático o espía industrial obtiene una copia física de un archivo FileMaker Pro y puede recrear o "engañar" a la estructura del dominio adivinando los nombres del Grupo, dicho pirata podría lograr acceso completo e irrestricto al archivo. Este es otro argumento para utilizar nombres de grupo distintivos." Las FDC apoyan esta posición.
8. Existe una nota especial adicional para agrupar nombres en la forma identificada en el documento de información técnica de FileMaker Inc. sobre autenticación externa de servidores, disponible en [http://www.filemaker.es/downloads/pdf/techbrief\\_fm8\\_server\\_auth.pdf](http://www.filemaker.es/downloads/pdf/techbrief_fm8_server_auth.pdf)

"El FileMaker Server 7 en Mac OS X Server busca el nombre corto del grupo devuelto desde los Servicios de Directorio. Ese es el nombre oficial que identifica el grupo ante el sistema, no el nombre largo (fácil para el usuario). De este modo, en la definición de cuentas de FileMaker Pro 7 para la autenticación del Servidor Externo, el nombre de grupo definido debe coincidir con el nombre corto del Grupo del Servicio de Directorio. En muchos casos los nombres corto y largo serán idénticos, no obstante, en algunos casos no será así. Los espacios y los caracteres ASCII altos podrían ser eliminados, por ejemplo. Por esa razón, se recomienda evitar a ambos. Los desarrolladores y los administradores deben verificar el nombre corto."

## 10.4 Consideraciones complementarias:

**Designación del servidor anfitrión:** Existen situaciones en que múltiples servidores alojan la "misma" solución. Son esencialmente copias exactas pero sirven a diferentes conjuntos de usuarios. En estos casos es posible que se considere una adición a la sintaxis del nombre del grupo externo. La inclusión de la designación del servidor anfitrión ayudará a asignar pertenencias de grupo para la solución en el servidor apropiado.

Sintaxis:

<<FileMakerGroupDesignation>>[\_]<HostServer>[\_]<SolutionName>[\_]<PrivilegeSetName>

**Servidor anfitrión:** Debe utilizar el DNS, Nombre de sistema o Nombre personalizado del servidor anfitrión.

Generalmente estos nombres serán iguales, sin embargo, es posible que no sea así en todas las implementaciones. Se debe elegir el nombre de Servidor anfitrión que identifica el servidor específico que está reconocido por la organización.

Ejemplos:

- Sintaxis estándar:
  - fm\_MiSolución\_\_Sólolectura
  - fm\_MiSolución\_\_Usuarios\_\_Generales
  - fm\_MiSolución\_\_asociadosdeVentas



- Uso de Consideraciones auxiliares para el servidor anfitrión.
  - fm\_servidorfm1\_MiSolución\_Usuarios Generales
  - fm\_servidorfm2\_MiSolución\_Usuarios Generales



## 11 ▷ Adhesión

Esta sección se basa en el supuesto de que las FDC se consideren valiosas y que los desarrolladores implementen pautas de cumplimiento de las normas en las soluciones que se apoyen en las recomendaciones incluidas en ellas. Teniendo en cuenta que mucha de la información contenida en este documento está sujeta a ampliación y cambios por diversas razones, es importante proveer una forma de documentar estas diferencias en forma coherente. La adhesión no se basa tanto en seguir estrictamente las pautas sino en documentar toda ampliación o diferencia que se produzca. El objeto es permitir que las FDC sirvan como referencia y reducir los esfuerzos necesarios para documentar el cómo y el porqué de una convención utilizada. Al permitir que las FDC se constituyan en referencia, se puede eliminar una parte significativa del esfuerzo. Lo que se necesita es indicar donde se produjo el cambio y la ampliación de las FDC. Esto permite que los futuros desarrolladores puedan referirse tanto a la razón fundamental de la asignación de nombres como a todas las consideraciones adicionales que se haya decidido implementar.

### 11.1 Objetivos de la convención

- ✓ Referencia a la versión de las FDC que se está utilizando
- ✓ Referencia a los desvíos de la versión de las FDC
- ✓ Referencia a cualquier otra información de importancia relacionada con la solución
- ✓ Definir como se debe almacenar / tener acceso a la información
- ✓ Definir el conjunto de información mínimo

### 11.2 Definición de problemas

**Referencia a la versión de las FDC que se está utilizando:** A medida que la línea de productos FileMaker evoluciona, se modificarán las FDC para mantenerlas actualizadas. Se deduce que una convención de soluciones se basará en una versión en particular de las FDC y en las ampliaciones específicas de la solución en cuestión. Por ello, es importante indicar la versión de las FDC.

**Referencia a los desvíos de la versión de las FDC:** Como con las FDC en sí, es muy posible que las ampliaciones y desvíos cambien y evolucionen. Por esta razón, es importante indicar los desvíos con alguna constancia de versión también.

**Referencia a cualquier otra información de importancia relacionada con la solución:** Algunos proyectos / soluciones sirven a los fines de versión, desarrollador, y otros elementos de metadatos. No obstante, el concepto de lanzamiento/versión se aplica sólo dentro del ciclo de vida del proyecto de banda estrecha, que transcurre durante la etapa de desarrollo inicial de un proyecto que se inicia de la nada. Estos conceptos no son demasiado valiosos en la fase de "larga cola" o "evolutiva" de un proyecto. Aún en el caso de que un único desarrollador haga la mayor parte de la construcción inicial, muchos desarrolladores pueden trabajar en este proyecto a lo largo de su vida útil. No obstante, por un determinado período o solución particular tiene sentido tener estos metadatos disponibles.

**Cómo almacenar y mantener accesible la información de adhesión:** Por razones de coherencia, esta información debe estar disponible en una ubicación comprendida universalmente.

**Conjunto de información mínimo:** Generalmente se acepta que toda solución debe estar acompañada por documentación. Algunos sistemas contienen también sus propios sistemas de ayuda que contiene tanto información para el usuario como para el desarrollador. Antes que imponer un método en particular de construcción de sistemas de ayuda/documentación, la recomendación debe establecer simplemente las pautas de un conjunto de información mínimo que debe estar disponible y la forma de acceder a éste.

### 11.3 Pautas de convenciones en cumplimiento de las normas

1. Toda solución debe tener un "Acerca de..." Un usuario o desarrollador con privilegios para abrir el archivo debe poder obtener los siguientes elementos:



- Versión de las FDC utilizada
  - Enlace URL para la ubicación de las FDC en el sitio web de FileMaker
  - Las FDC almacenadas como archivo PDF dentro de un campo contenedor que se pueda exportar.
- Desvíos de la versión de las FDC.
  - Enlace URL para la ubicación de desvíos de la versión en el sitio web del desarrollador.
  - Documento almacenado como PDF dentro de un campo contenedor que pueda exportarse y que contenga todos los desvíos y ampliaciones aplicables a la solución.
- Metadatos de documentación: toda otra información, a discreción del desarrollador de la solución, que se considere necesaria. Esto se puede almacenar en un único campo o en la cantidad necesaria de campos. Debe estar disponible en la interfaz "Acerca de..." que se presenta. Algunos ejemplos pueden incluir lo siguiente, sin carácter limitativo:
  - Empresa
  - Versión de la solución
  - Agradecimientos
  - Fecha de lanzamiento
  - Nombre o nombres de los desarrolladores
  - Información de contacto

Nota:

Los desarrolladores de Runtime tienen obligaciones legales respecto de "Acerca de Presentaciones". Ver la Guía de Desarrollo FileMaker 8, página 82.



## Apéndice A ▷ Palabras reservadas de SQL

La siguiente tabla enumera todas las palabras reservadas de la norma SQL.

Fuente: <http://developer.mimer.se/validator/sql-reserved-words.tml>

SQL-92	SQL-99	SQL-2003
ABSOLUTE	ABSOLUTE	
ACTION	ACTION	
ADD	ADD	ADD
	AFTER	ALL
ALL	ALL	
ALLOCATE	ALLOCATE	ALLOCATE
ALTER	ALTER	ALTER
AND	AND	AND
ANY	ANY	ANY
ARE	ARE	ARE
	ARRAY	ARRAY
AS	AS	AS
ASC	ASC	
	ASENSITIVE	ASENSITIVE
ASSERTION	ASSERTION	
	ASYMMETRIC	ASYMMETRIC
AT	AT	AT
	ATOMIC	ATOMIC
AUTHORIZATION	AUTHORIZATION	AUTHORIZATION
AVG		
	BEFORE	
BEGIN	BEGIN	BEGIN
BETWEEN	BETWEEN	BETWEEN
		BIGINT
	BINARY	BINARY
BIT	BIT	
BIT_LENGTH		
	BLOB	BLOB
	BOOLEAN	BOOLEAN
BOTH	BOTH	BOTH
	BREADTH	
BY	BY	BY
CALL	CALL	CALL
	CALLED	CALLED
CASCADE	CASCADE	
CASCADED	CASCADED	CASCADED
CASE	CASE	CASE



SQL-92	SQL-99	SQL-2003
CAST	CAST	CAST
CATALOG	CATALOG	
CHAR	CHAR	CHAR
CHAR_LENGTH		
CHARACTER	CHARACTER	CHARACTER
CHARACTER_LENGTH		
CHECK	CHECK	CHECK
	CLOB	CLOB
CLOSE	CLOSE	CLOSE
COALESCE		
COLLATE	COLLATE	COLLATE
COLLATION	COLLATION	
COLUMN	COLUMN	COLUMN
COMMIT	COMMIT	COMMIT
CONDITION	CONDITION	CONDITION
CONNECT	CONNECT	CONNECT
CONNECTION	CONNECTION	
CONSTRAINT	CONSTRAINT	CONSTRAINT
CONSTRAINTS	CONSTRAINTS	
	CONSTRUCTOR	
CONTAINS		
CONTINUE	CONTINUE	CONTINUE
CONVERT		
CORRESPONDING	CORRESPONDING	CORRESPONDING
COUNT		
CREATE	CREATE	CREATE
CROSS	CROSS	CROSS
	CUBE	CUBE
CURRENT	CURRENT	CURRENT
CURRENT_DATE	CURRENT_DATE	CURRENT_DATE
	CURRENT_DEFAULT_TRANSFORM_	
GROUP	CURRENT_DEFAULT_TRANSFORM_	
GROUP		
CURRENT_PATH	CURRENT_PATH	CURRENT_PATH
	CURRENT_ROLE	CURRENT_ROLE
CURRENT_TIME	CURRENT_TIME	CURRENT_TIME
CURRENT_TIMESTAMP	CURRENT_TIMESTAMP	CURRENT_TIMESTAMP
	CURRENT_TRANSFORM_GROUP_FO	



SQL-92	SQL-99	SQL-2003
R_TYPE	CURRENT_TRANSFORM_GROUP_ FO	
R_TYPE		
CURRENT_USER	CURRENT_USER	CURRENT_USER
CURSOR	CURSOR	CURSOR
	CYCLE	CYCLE
	DATA	
DATE	DATE	DATE
DAY	DAY	DAY
DEALLOCATE	DEALLOCATE	DEALLOCATE
DEC	DEC	DEC
DECIMAL	DECIMAL	DECIMAL
DECLARE	DECLARE	DECLARE
DEFAULT	DEFAULT	DEFAULT
DEFERRABLE	DEFERRABLE	
DEFERRED	DEFERRED	
DELETE	DELETE	DELETE
DEPTH		
	DEREF	DEREF
DESC	DESC	
DESCRIBE	DESCRIBE	DESCRIBE
DESCRIPTOR	DESCRIPTOR	
DETERMINISTIC	DETERMINISTIC	DETERMINISTIC
DIAGNOSTICS	DIAGNOSTICS	
DISCONNECT	DISCONNECT	DISCONNECT
DISTINCT	DISTINCT	DISTINCT
DO	DO	DO
DOMAIN	DOMAIN	
DOUBLE	DOUBLE	DOUBLE
DROP	DROP	DROP
	DYNAMIC	DYNAMIC
	EACH	EACH
		ELEMENT
ELSE	ELSE	ELSE
ELSEIF	ELSEIF	ELSEIF
END	END	END
	EQUALS	
ESCAPE	ESCAPE	ESCAPE
EXCEPT	EXCEPT	EXCEPT
EXCEPTION	EXCEPTION	
EXEC	EXEC	EXEC
EXECUTE	EXECUTE	EXECUTE



SQL-92	SQL-99	SQL-2003
EXISTS	EXISTS	EXISTS
EXIT	EXIT	EXIT
EXTERNAL	EXTERNAL	EXTERNAL
EXTRACT		
FALSE	FALSE	FALSE
FETCH	FETCH	FETCH
	FILTER	FILTER
FIRST	FIRST	
FLOAT	FLOAT	FLOAT
FOR	FOR	FOR
FOREIGN	FOREIGN	FOREIGN
FOUND	FOUND	
	FREE	FREE
FROM	FROM	FROM
FULL	FULL	FULL
FUNCTION	FUNCTION	FUNCTION
	GENERAL	
GET	GET	GET
GLOBAL	GLOBAL	GLOBAL
GO	GO	
GOTO	GOTO	
GRANT	GRANT	GRANT
GROUP	GROUP	GROUP
	GROUPING	GROUPGROUPING
HANDLER	HANDLER	HANDLER
HAVING	HAVING	HAVING
	HOLD	HOLD
HOUR	HOUR	HOUR
IDENTITY	IDENTITY	IDENTITY
IF	IF	IF
IMMEDIATE	IMMEDIATE	IMMEDIATE
IN	IN	IN
INDICATOR	INDICATOR	INDICATOR
INITIALLY	INITIALLY	
INNER	INNER	INNER
INOUT	INOUT	INOUT
INPUT	INPUT	INPUT
INSENSITIVE	INSENSITIVE	INSENSITIVE
INSERT	INSERT	INSERT
INT	INT	INT
INTEGER	INTEGER	INTEGER
INTERSECT	INTERSECT	INTERSECT



SQL-92	SQL-99	SQL-2003
INTERVAL	INTERVAL	INTERVAL
INTO	INTO	INTO
IS	IS	IS
ISOLATION	ISOLATION	
	ITERATE	ITERATE
JOIN	JOIN	JOIN
KEY	KEY	
LANGUAGE	LANGUAGE	LANGUAGE
	LARGE	LARGE
LAST	LAST	
	LATERAL	LATERAL
LEADING	LEADING	LEADING
LEAVE	LEAVE	LEAVE
LEFT	LEFT	LEFT
LEVEL	LEVEL	
LIKE	LIKE	LIKE
LOCAL	LOCAL	LOCAL
	LOCALTIME	LOCALTIME
	LOCALTIMESTAMP	LOCALTIMESTAMP
	LOCATOR	
LOOP	LOOP	LOOP
LOWER		
	MAP	
MATCH	MATCH	MATCH
MAX		
		MEMBER
		MERGE
	METHOD	METHOD
MIN		
MINUTE	MINUTE	MINUTE
	MODIFIES	MODIFIES
MODULE	MODULE	MODULE
MONTH	MONTH	MONTH
		MULTISET
NAMES	NAMES	
NATIONAL	NATIONAL	NATIONAL
NATURAL	NATURAL	NATURAL
NCHAR	NCHAR	NCHAR
	NCLOB	NCLOB
NEXT	NEXT	
NO	NO	NO
	NONE	NONE



SQL-92	SQL-99	SQL-2003
NOT	NOT	NOT
NULL	NULL	NULL
NULLIF		
NUMERIC	NUMERIC	NUMERIC
	OBJECT	
OCTET_LENGTH		
OF	OF	OF
	OLD	OLD
ON	ON	ON
ONLY	ONLY	ONLY
OPEN	OPEN	OPEN
OPTION	OPTION	
OR	OR	OR
ORDER	ORDER	ORDER
	ORDINALITY	
OUT	OUT	OUT
OUTER	OUTER	OUTER
OUTPUT	OUTPUT	OUTPUT
	OVER	OVER
OVERLAPS	OVERLAPS	OVERLAPS
PAD	PAD	
PARAMETER	PARAMETER	PARAMETER
PARTIAL	PARTIAL	
	PARTITION	PARTITION
PATH	PATH	
POSITION		
PRECISION	PRECISION	PRECISION
PREPARE	PREPARE	PREPARE
PRESERVE	PRESERVE	
PRIMARY	PRIMARY	PRIMARY
PRIOR	PRIOR	
PRIVILEGES	PRIVILEGES	
PROCEDURE	PROCEDURE	PROCEDURE
PUBLIC	PUBLIC	
	RANGE	RANGE
READ	READ	
	READS	READS
REAL	REAL	REAL
	RECURSIVE	RECURSIVE
	REF	REF
REFERENCES	REFERENCES	REFERENCES
	REFERENCING	REFERENCING



SQL-92	SQL-99	SQL-2003
RELATIVE	RELATIVE	
	RELEASE	RELEASE
REPEAT	REPEAT	REPEAT
RESIGNAL	RESIGNAL	RESIGNAL
RESTRICT	RESTRICT	
	RESULT	RESULT
RETURN	RETURN	RETURN
RETURNS	RETURNS	RETURNS
REVOKE	REVOKE	REVOKE
RIGHT	RIGHT	RIGHT
ROLLBACK	ROLLBACK	ROLLBACK
	ROLLUP	ROLLUP
ROUTINE	ROUTINE	
	ROW	ROW
ROWS	ROWS	ROWS
	SAVEPOINT	SAVEPOINT
SCHEMA	SCHEMA	
	SCOPE	SCOPE
SCROLL	SCROLL	SCROLL
	SEARCH	SEARCH
SECOND	SECOND	SECOND
SECTION	SECTION	
SELECT	SELECT	SELECT
	SENSITIVE	SENSITIVE
SESSION	SESSION	
SESSION_USER	SESSION_USER	SESSION_USER
SET	SET	SET
	SETS	
SIGNAL	SIGNAL	SIGNAL
	SIMILAR	SIMILAR
SIZE	SIZE	
SMALLINT	SMALLINT	SMALLINT
SOME	SOME	SOME
SPACE	SPACE	
SPECIFIC	SPECIFIC	SPECIFIC
	SPECIFICTYPE	SPECIFICTYPE
SQL	SQL	SQL
SQLCODE		
SQLERROR		
SQLEXCEPTION	SQLEXCEPTION	SQLEXCEPTION
SQLSTATE	SQLSTATE	SQLSTATE
SQLWARNING	SQLWARNING	SQLWARNING



SQL-92	SQL-99	SQL-2003
	START	START
	STATE	
	STATIC	STATIC
		SUBMULTISET
SUBSTRING		
SUM		
	SYMMETRIC	SYMMETRIC
	SYSTEM	SYSTEM
SYSTEM_USER	SYSTEM_USER	SYSTEM_USER
TABLE	TABLE	TABLE
		TABLESAMPLE
TEMPORARY	TEMPORARY	
THEN	THEN	THEN
TIME	TIME	TIME
TIMESTAMP	TIMESTAMP	TIMESTAMP
TIMEZONE_HOUR	TIMEZONE_HOUR	TIMEZONE_HOUR
TIMEZONE_MINUTE	TIMEZONE_MINUTE	TIMEZONE_MINUTE
TO	TO	TO
TRAILING	TRAILING	TRAILING
TRANSACTION	TRANSACTION	
TRANSLATE		
TRANSLATION	TRANSLATION	TRANSLATION
	TREAT	TREAT
	TRIGGER	TRIGGER
TRIM		
TRUE	TRUE	TRUE
	UNDER	
UNDO	UNDO	UNDO
UNION	UNION	UNION
UNIQUE	UNIQUE	UNIQUE
UNKNOWN	UNKNOWN	UNKNOWN
	UNNEST	UNNEST
UNTIL	UNTIL	UNTIL
UPDATE	UPDATE	UPDATE
UPPER		
USAGE	USAGE	
USER	USER	USER
USING	USING	USING
VALUE	VALUE	VALUE
VALUES	VALUES	VALUES
VARCHAR	VARCHAR	VARCHAR
VARYING	VARYING	VARYING
VIEW	VIEW	



SQL-92	SQL-99	SQL-2003
WHEN	WHEN	WHEN
WHENEVER	WHENEVER	WHENEVER
WHERE	WHERE	WHERE
WHILE	WHILE	WHILE
	WINDOW	WINDOW
WITH	WITH	WITH
	WITHIN	WITHIN
	WITHOUT	WITHOUT
WORK	WORK	
WRITE	WRITE	
YEAR	YEAR	YEAR
ZONE	ZONE	



## Apéndice B ▷ Tabla de uso de caracteres

- [X] No permitido o con Advertencia al usarlo  
 [NR] No recomendado  
 [C] Usos de la convención  
 [R] Carácter reservado

	Carácter	Nombres de archivo	Nombres de tablas	Nombres de TO	Nombres de campo	Nombres de Presentaciones	Nombres de funciones personalizadas	Nombres de listas de valores	Nombres de guiones	Nombres de cuenta	Variables
.	Punto	X	X	X	X	NR	NR	NR	NR	NR	NR
+	Más	NR	X	X	X	NR	X	NR	NR	NR	NR
*	Asterisco	NR	X	X	X	NR	X	NR	NR	NR	NR
^	Oclusión	NR	X	X	X	NR	X	NR	NR	NR	NR
=	Igual	NR	X	X	X	NR	X	NR	NR	NR	NR
>	Mayor que	NR	X	X	X	NR	X	NR	NR	NR	NR
(	Paréntesis izquierdo	NR	X	X	X	NR	X	NR	NR	NR	NR
«	Comillas	NR	X	X	X	NR	X	NR	NR	NR	NR
:	Dos puntos	NR	NR	NR	X	NR	NR	NR	NR	NR	NR
,	Coma	NR	X	X	X	NR	X	NR	NR	NR	NR
-	Menos	NR	X	X	X	NR	X	NR	NR	NR	NR
/	Barra	NR	X	X	X	NR	X	NR	NR	NR	NR
&	Ampersand	NR	X	X	X	NR	X	NR	NR	NR	NR
≠	No igual	NR	X	X	X	NR	X	NR	NR	NR	NR
<	Menor que	NR	X	X	X	NR	X	NR	NR	NR	NR
)	Paréntesis derecho	NR	X	X	X	NR	X	NR	NR	NR	NR
;	Punto y coma	NR	X	X	X	NR	X	NR	NR	NR	NR
::	Indicador relacional	NR	NR	X	X	NR	X	NR	NR	NR	NR
{	Llave izquierda	NR	NR	X	NR	NR	NR	NR	NR	NR	NR
}	Llave derecha	NR	X	X	NR	NR	NR	NR	NR	NR	NR
?	Pregunta	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
~	Tilde	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
`	Apóstrofe	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
!	Exclamación	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
%	Porcentaje	NR	NR	X	NR	NR	NR	NR	NR	NR	NR
	Barra vertical	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
\$	Dólar	NR	X	X	X	NR	NR	NR	NR	NR	R
\$\$	Doble Dólar	NR	X	X	X	NR	NR	NR	NR	NR	R



	Carácter	Nombres de archivo	Nombres de tablas	Nombres de TO	Nombres de campo	Nombres de Presentaciones	Nombres de funciones personalizadas	Nombres de listas de valores	Nombres de guiones	Nombres de cuenta	Variables
[	Corchete izquierdo	NR	X	X	X	NR	X	NR	NR	NR	NR
]	Corchete derecho	NR	X	X	X	NR	X	NR	NR	NR	NR
—	Doble subrayado	C	C	C	C	C	C	C	C	C	NR



## Apéndice C ▷ Terminología y definiciones

**CamelCase:** Práctica de escribir palabras compuestas o frases donde las palabras están unidas entre sí sin espacios, y cada palabra está en mayúsculas dentro de la composición. El nombre proviene de las mayúsculas que “sobresalen” en el medio de la palabra o frase compuesta como si fueran las jorobas de un camello.

`CamelCaseLooksLikeThis` (CajatipocamelloseVedeEstaForma)

`lowerCamelCaseLooksTheSame` (minúsculasyCajatipocamelloseVedelMismoModo)

`UpperCamelCaseLooksLikeThis` (MayúsculasyCajatipocamelloseVedeEstaForma)

Existen dos variedades comunes de CamelCase, que se distingue por el manejo de la letra inicial. La variedad en la cual la primera letra está en mayúsculas se denomina habitualmente UpperCamelCase (Mayúscula y CamelCase), tipo Pascal o con dos mayúsculas. La variedad en la cual la primera letra se deja en minúsculas se denomina habitualmente lowerCamelCase (minúscula y CamelCase) o a veces simplemente CamelCase. Por razones de claridad, en este documento se utilizan los términos mayúscula y CamelCase y minúsculas y CamelCase respectivamente.

Fuente: <http://en.wikipedia.org/wiki/CamelCase>

**Función Personalizada pública:** Una función personalizada Pública es la primera función convocada, o principal, dentro de un grupo de funciones personalizadas. En muchos casos se utiliza un ‘conjunto’ o ‘grupo’ de funciones personalizadas para ofrecer un juego completo de funcionalidades. En otros casos, una función personalizada puede existir por sí misma. En ambos casos, una función personalizada que se convoca directamente se considera una función personalizada Pública.

**Función Personalizada privada:** Una función personalizada privada nunca se llama en forma directa en un cálculo. Sólo se puede convocar desde otra función personalizada Pública. Las funciones Privadas son subordinadas de las funciones Públicas. Es muy posible tener una función personalizada que sea siempre privada. Por ejemplo, se puede crear una función personalizada que esté diseñada como función “ayudante” de otras múltiples funciones. En otras palabras, sólo se utilizará, siempre, en conjunto con otra función personalizada.

**Ocurrencia principal de la tabla:** La PTO es una Ocurrencia de tablas especial. La PTO siempre funciona como la Ocurrencia de tablas designada que se utilizará al crear cálculos con referencias internas. Esos cálculos derivan de los datos contenidos dentro del contexto de la tabla en sí y nunca de datos de otra tabla.

**Tabla separadora entre categorías de tablas (Spacing table):** Es el nombre de una técnica para crear tablas ‘etiqueta’ o ‘separadora’ dentro del cuadro de diálogo ‘Define’ (Definir). Esta técnica utiliza tablas que no contienen campos con el objeto de agrupar y categorizar tablas y Ocurrencias de tablas.

**Tabla:** Recopilación de datos que pertenecen a un sujeto, por ejemplo clientes o precios de acciones. El archivo de base de datos contiene una o más tablas, las cuales consisten en campos y registros. Cuando se crea una tabla nueva, una representación visual de la tabla, o una Ocurrencia de la Tabla, aparece en el Gráfico de Relaciones.

Se pueden especificar múltiples Ocurrencias (con nombres únicos) de la misma tabla a fin de trabajar con relaciones complejas en el gráfico.

**Ocurrencia de tablas:** Una Ocurrencia de la tabla se refiere a una instancia de una tabla en el Gráfico de relaciones. Se debe tener en cuenta que todas las interacciones con una tabla a través del ambiente de desarrollo se realizarán con Ocurrencias de Tablas. Esta es la única forma de ‘dirigirse’ a una tabla y a su contenido.

**Tabla de origen:** Una Ocurrencia de la tabla está relacionada con una tabla. La tabla de origen se refiere a la tabla con la cual se relaciona la Ocurrencia de la tabla. Por ejemplo, una Ocurrencia de la tabla denominada “AdmissionInterface|Classes|ClassList” (Interfaz de admisión|Clases|Lista de clases) tiene una tabla de origen denominada “classes” (clases). También se hará referencia a la Tabla de origen como Nombre de tabla de origen.

**Identificador lógico de soluciones:** El LSI es un prefijo utilizado para cada archivo secundario y un sufijo opcional para cada archivo primario de una solución.

**Designador de grupo FileMaker:** Es un prefijo utilizado para identificar grupos autenticados en forma externa.



## Apéndice D ▷ Leyenda de sintaxis

El documento de las FDC utiliza un estilo específico para indicar los diversos componentes de toda sintaxis utilizada dentro de las diferentes secciones. La siguiente tabla explica las diversas notaciones.

### Leyenda de sintaxis de las FileMaker Convenciones de Desarrollos

	No incluir caracteres entre signos indica asignación de nombres opcionales y de forma natural/libre
[ ]	El objeto de los componentes incluidos entre estos signos son ser utilizados en la forma en que se exhiben
{ }	Los componentes incluidos entre estos signos son opcionales y pueden ser definidos por el desarrollador
( )	Los componentes incluidos entre estos signos son opcionales pero si son utilizados deben tener un valor definido o inferido
< >	El objeto de los componentes incluidos entre estos signos es que se aporten pero deben tener un valor definido o inferido
<< >>	El objeto de los componentes incluidos entre estos signos es que el desarrollador los aporte y defina

