



**UNIVERSIDAD LATINA
DE COSTA RICA**
LAUREATE INTERNATIONAL UNIVERSITIES®

Facultad de Tecnologías de la Información y Comunicación.

Escuela de Ingeniería de Sistemas Informáticos.

Lista de Propiedades de MigLayout.

Programación III (BSI-090).

Profesor: Lic. Jorge Isaac Vásquez Valenciano.

Estudiante: Anthony Quirós Alfaro.

(20160111305).

Universidad Latina San Pedro.

11 junio de 2018.

Índice

INTRODUCCIÓN:	1
LISTA DE TIPOS DE UNIDADES MIGLAYOUT:	2
LISTA DE TIPOS DE PROPIEDADES DE MIGLAYOUT	3
CONCLUSIÓN:	17
BIBLIOGRAFÍA:	19

Introducción:

En el presente trabajo vamos a investigar las propiedades del MigLayout, para así entender su significado y su respectivo uso a la hora de hacer trabajos de programación, esto nos facilitara la comprensión de las a la hora de crear código.

Lista de tipos de unidades MigLayout:

1. **""** – No se especifica ninguna unidad.
2. **px** – Píxeles. Píxeles normales asignados directamente a la pantalla.
3. **%** - Un porcentaje del tamaño del contenedor.
4. **lp** – Píxeles lógicos.
5. **pt** – Puntos.
6. **mm** – Milímetros.
7. **cm** – Centímetros.
8. **in** – Pulgadas.
9. **sp** – Porcentaje de pantalla.
10. **al** – Aliniacion de limites visuales.
11. **n/ null** – Valor nulo
12. **r/ rel / related** - indica que dos componentes o columnas/ filas se consideran relacionados.
13. **u / unrel / unrelatedated** - indica que dos componentes o columnas / filas se consideran no relacionados.
14. **p / para / paragraph** - Se utiliza un espaciado que se considera apropiado para un párrafo.
15. **i / ind / indent** - Un espaciado que se considera apropiado para sangría.
16. **min/mínimum** - una referencia al tamaño mínimo más grande de la columna / fila.
17. **p / pref / preferred** - Una referencia al tamaño preferido más grande de la columna / fila.

18. **max / maximum** -una referencia al tamaño máximo más pequeño de la columna / fila.

19. **Button**- Una referencia al tamaño mínimo de plataforma para un botón.

Lista de tipos de propiedades de MigLayout

wrap	Establece el modo de envoltura automática para el diseño. Esto significa que la cuadrícula se ajustará a una nueva columna / fila después de un cierto número de columnas (para flujo horizontal) o filas (para flujo vertical). El número se especifica como un entero después de la palabra clave o, si no, se usará el número de restricciones de columna / fila especificadas. Un diseño de envoltura significa que después de que countse haya agregado el componente: th, el diseño se ajustará y continuará en la siguiente fila / columna. Si envoltura está apagado (por defecto) el componente de restricción "wrap"y "newline" se puede utilizar para controlar el ajuste.	"wrap" "wrap 4"
<u>gap</u> <u>gapx</u> <u>gapy</u>	Especifica el espacio predeterminado entre las celdas de la grilla y, por lo tanto, anula el valor predeterminado de la plataforma. Los espacios se especifican como BoundSize .	"gap 5px 10px" "gap unrel rel" "gapx 10::50" "gapy 0: rel:null" "gap 10! 10!"
debug		
novisualpadding	Desactiva el relleno de los límites visuales (por ejemplo, compensación de sombras)	"novisualpadding "
nogrid	Pone el diseño en modo de solo flujo. Todos los componentes en la dirección de flujo se colocarán en la misma celda y, por lo tanto, no se alinearán con el componente en otras filas / columnas. Para el flujo horizontal normal, esto es lo mismo que decir que todos los	"nogrid"

	componentes se colocarán en la primera y única columna.	
fill fillx filly	Reclama todo el espacio disponible en el contenedor para las columnas y / o filas. Al menos un componente debe tener una "grow" constancia para que llene el contenedor. El espacio se dividirá en partes iguales, aunque honrando "growpriority". Si no se han establecido columnas / filas "grow", el peso de crecimiento de los componentes en las filas / columnas migrará a esa fila / columna.	"fill" "fillx" "filly"
ins/insets ["dialog"] ["panel"] [top/all [left] [bottom] [right]]	Especificó las inserciones para el contenedor presentado. Las brechas antes / después de la primera / última columna / fila anulan estas inserciones de diseño. Esto es lo mismo que establecer un EmptyBorder en el contenedor, pero sin quitar ningún borde que ya esté allí. El valor predeterminado es "panel"(o cero si hay componentes de acoplamiento). El tamaño "dialog" y las "panel" inserciones son devueltos por la corriente PlatformConverter. Los valores del recuadro alrededor también se pueden establecer explícitamente para uno o más lados. Las inserciones en los lados que se establecen "null" o "n" obtendrán los valores predeterminados proporcionados por PlatformConverter. Si se especifican menos de cuatro lados, el último valor se usará para el lado restante. Los espacios se especifican como un valor de unidad. Véase más arriba. Tenga en cuenta que las inserciones predeterminadas son "panel"	"insets dialog" "ins 0" "insets 10px n n n" "insets 10 20 30 40"
flowy	one el diseño en modo de flujo vertical. Esto significa que la siguiente celda normalmente está debajo y el siguiente componente se colocará allí en lugar de a la derecha. El valor predeterminado es flujo horizontal.	"flowy"
al/align alignx [aligny] aligny/ay align aligny/ax align	Especifica la alineación para los componentes presentados como un grupo. Si los límites totales de todos los componentes presentados no cubren todo el contenedor, el valor de alineación se usa para colocar los componentes dentro del contenedor sin cambiar sus posiciones relativas. La alineación se puede especificar como UnitValue o AlignKeyword . Véase más arriba. Si se utiliza AlignKeyword, la "align" palabra clave puede omitirse.	"align 50% 50%" "aligny top" "alignx leading" "align 100px" "top, left"

	Tenga en cuenta que la alineación de línea base no funciona, ya que no es para componentes individuales.	
ltr/lefttoright rtl/righttoleft	Anula la ComponentOrientation propiedad del contenedor para este diseño. Normalmente, este valor depende de la Localización de la aplicación. Esta restricción anula ese valor.	"ltr" "lefttoright" "rtl"
ttb/toptobottom btt/bottomtotop	Especifica si los componentes se deben agregar en la cuadrícula de abajo hacia arriba o de arriba hacia abajo. Este valor no se recoge del contenedor y es de arriba a abajo de forma predeterminada.	"ttb" "toptobottom" "btt"
hidemode	Establece el modo oculto predeterminado para el diseño. Este modo oculto puede ser anulado por la restricción del componente. El modo ocultar especificó cómo el administrador de diseño debe manejar un componente que no está visible. Los modos son: 0- Predeterminado. Significa que los componentes invisibles se manejarán exactamente como si fueran visibles. 1- El tamaño de un componente invisible se establecerá en 0, 0. 2- El tamaño de un componente invisible se establecerá en 0, 0y los espacios también se establecerán a su 0alrededor. 3 - Los componentes invisibles no participarán del diseño en absoluto y, por ejemplo, no ocuparán una celda de la grilla.	"hidemode 1"
nocache	Indica al motor de diseño que no use cachés. Normalmente, esto solo debería ser necesario si "%"se usa la unidad ya que es una función del tamaño principal. Si está experimentando problemas de revalidación, puede intentar establecer esta restricción.	"nocache"
sizegroup [name] sg [name]	Le da a la fila un nombre de grupo de tamaño. Todas las filas que comparten un nombre de grupo de tamaño obtendrán el mismo BoundSize que la fila con el tamaño mínimo / preferido más grande. Esto es más útil cuando el tamaño de la fila no está explícitamente establecido y, por lo tanto, está determinado por el componente más grande que es la (s) fila (s). Se "" puede usar un nombre vacío a menos que haya más de un grupo.	"sg" "sg group1" "sizegroup props"
fill	Establecer el valor predeterminado para los componentes de "grow" la dimensión de la fila. Por lo tanto, para las columnas, los	"fill"

	componentes en esa columna tendrán por defecto una "growx" restricción (que puede ser anulada por las restricciones individuales del componente). Tenga en cuenta que esta propiedad no afecta el tamaño de la fila, sino el tamaño de los componentes en la fila.	
nogrid	Pone la fila en modo solo flujo. Todos los componentes en la dirección de flujo se colocarán en la misma celda y, por lo tanto, no se alinearán con el componente en otras filas / columnas. Esta propiedad solo se cumplirá si la fila está en la dirección del flujo. Por lo tanto, para el flujo horizontal normal ("flowx") solo se usa para filas y "flowy" solo se usa para columnas.	"nogrid"
grow [weight]	Establece qué tan buena debería ser la fila para crecer en relación con otras filas. El peso (por defecto es 100 si no se especifica) es puramente un valor relativo al peso de otras filas. El doble de peso duplicará el espacio extra. Si no se establece esta restricción, el peso de crecimiento se establece en cero y la columna no crecerá (a menos que "fill" se establezca en las Restricciones de diseño y ninguna otra fila tenga un peso mayor que cero). El peso de crecimiento solo se comparará con los pesos de las filas con la misma prioridad de crecimiento. Vea abajo.	"grow 50" "grow"
growprio prio	Establece la prioridad de crecimiento para la fila (no para los componentes en la fila). Al crecer, todas las filas con mayores prioridades crecerán hasta su tamaño máximo antes de considerar cualquier fila con menor prioridad. La prioridad predeterminada de crecimiento es 100. Esto se puede usar para hacer que ciertas filas crezcan al máximo antes de que otras filas comiencen a crecer.	"growprio 50"
shrinkweight	Establece qué tan entusiasta / renuente debería ser la fila de encogerse en relación con otras filas. El peso es puramente un valor relativo para los pesos de otras filas. El doble de peso se reducirá dos veces cuando el espacio sea escaso. Si no se establece esta restricción, el peso de contracción se establece de manera predeterminada en 100, lo que significa que todas las filas de forma predeterminada pueden reducirse a su tamaño mínimo, pero no menos. El peso de contracción solo se comparará con	"shrink 50" "shrinkweight 0"

	los pesos en el mismo grupo de prioridad de contracción (otras filas con la misma prioridad de contracción). Vea abajo.	
shrinkprio prio shp prio	Establece la prioridad de reducción para la fila (no para los componentes en la fila). Cuando el espacio es escaso y las filas deben reducirse, todas las filas con prioridades más altas se reducirán a su tamaño mínimo antes de considerar cualquier fila con prioridad más baja. La prioridad de contracción predeterminada es 100. Esto se puede usar para hacer que ciertas filas se reduzcan a min antes de que otras filas comiencen a contraerse.	"shrinkprio 50" "shp 110"
align align al align	Especifica la alineación predeterminada para los componentes en la fila. Esta alineación predeterminada puede anularse configurando la alineación para el componente en la Restricción de componentes. La alineación de filas predeterminada es "left" para columnas y "center" para filas. La alineación se puede especificar como UnitValue o AlignKeyword. Véase más arriba. Si se usa AlignKeyword, la "align" parte se puede omitir. Tenga en cuenta que la alineación de línea de base no funciona si el componente no puede obtener su tamaño preferido en la dimensión vertical.	"align 50%" "align top" "al leading" "align 100px" "top, left" "align baseline"
gap gapbefore [gap] gapbefore gap gapafter gap	Especifica el espacio antes y / o después de la fila. El espacio se especifica entre las restricciones de fila (entre "[") ("]"). "gapleft", "gapright", "gaptop", "gapbottom" también puede ser usado.	"gap 10 20" "gap 10:20:30 10px:20%:30in" "gapbefore 10px, gapafter 20px"
wrap [gapsize]	Se ajusta a una nueva columna / fila después de que el componente se haya colocado en la siguiente celda disponible. Esto significa que el siguiente componente se colocará en la nueva fila / columna. ¡Propina! Lea el wrap como "wrap after". Si se especifica "gapsize" se anulará el tamaño del espacio entre la fila actual y la siguiente (o la columna si "flowy"). Tenga en cuenta que el tamaño de los huecos es posterior a la fila en la que terminará este componente.	"wrap" "wrap 15px" "wrap push" "wrap 15:push"
newline [gapsize]	Se ajusta a una nueva columna / fila antes de colocar el componente en la siguiente celda disponible. Esto significa que el presente componente será puesto en una nueva fila / columna. ¡Propina! Leer el ajuste como "en	"newline" "newline 15px" "newline push" "newline 15:push"

	una nueva línea". Si se especifica "gapsize" se anulará el tamaño del espacio entre la fila actual y la siguiente (o la columna si "flowy"). Tenga en cuenta que el tamaño de las lagunas está antes de la fila en la que terminará este componente.	
push [weightx][weighty] pushx [weightx] pushy [weighty]	Hace crecer la fila y / o columna en la que reside el componente "weight". Esto se puede usar en lugar de tener una palabra clave "crecer" en las restricciones columna / fila.	"push" "pushx 200" "pushy"
skip [count]	Salta una cantidad de celdas en el flujo. Esto se usa para pasar por varias celdas antes de buscar la siguiente celda libre. La omisión se realiza antes de que este componente se coloque en una celda y, por lo tanto, esta celda se ve afectada por ella. "count" se predetermina a 1 si no se especifica.	"skip" "skip 3"
span [countx] [county] spany/sy [count] spanx/sx [count]	Atraviesa la celda actual (se funde) en varias celdas. En la práctica esto significa que esta célula y el <i>recuento de</i> número de células serán tratados como una célula y el componente pueden utilizar el espacio que todas estas células tienen. <i>contar por defecto</i> tiene un valor realmente alto que prácticamente significa <i>abarcar hasta el final o la fila / columna</i> . Tenga en cuenta que una celda puede dividirse y dividirse al mismo tiempo, por lo que, por ejemplo, puede abarcar 2 celdas y dividir ese espacio para tres componentes. "span" para la primera celda en una fila es lo mismo que establecer "nogrid" en la restricción de fila.	"span" "span 4" "span 2 2" "spanx 10" "spanx 2, spany 2"
split [count]	Divide la celda en varias subcélulas. Básicamente, esto significa que el siguiente countnúmero de componentes se colocará en la misma celda, uno al lado del otro con espacios vacíos. Solo el primer componente en una celda puede establecer la división, cualquier palabra "split" clave subsiguiente en la celda será ignorada. countel valor predeterminado es <i>infinito</i> si no se especifica, lo que significa que "split" solo colocará todos los componentes posteriores en la misma celda. "skip", "wrap"y "newline" saldrá de la	"split" "split 4"

	celda dividida. Los dos últimos se moverán a una nueva fila / columna como de costumbre. "skip" saltará si la división y continuará en la siguiente celda.	
<u>cell</u> <i>col row</i> [span x [span y]]	Establece la celda de la grilla en la que debe colocarse el componente. Si ya hay componentes en la celda, compartirán la celda. Si hay dos enteros especificados, se interpretarán como coordenadas absolutas para la columna y la fila. El flujo continuará después de esta celda. Cuantas celdas se extenderán es opcional pero puede especificarse. Es lo mismo que usar las palabras clave <code>spanxy spany</code> .	"cell 2 2" "cell 1 1 2 2"
<u>flowx</u> <u>flowy</u>	Establece la dirección del flujo en la celda. Por defecto, la dirección del flujo en la celda es la misma que la dirección del flujo para el diseño. Entonces, si los componentes fluyen de izquierda a derecha, también lo harán para el flujo dentro de la celda. El primer componente agregado a una celda puede cambiar el flujo de la celda. Si se cambia la dirección del flujo a <code>flowy</code> los componentes en la celda se colocarán encima / debajo de la otra.	"flowy" "flowx"
<u>w/width</u> <i>size</i> <u>h/height</u> <i>size</i>	Reemplaza el tamaño predeterminado del componente establecido por el delegado de UI o por el desarrollador explícitamente en el componente. El tamaño se especifica como BoundSize . Consulte la sección de <i>Tipos de argumentos comunes</i> más arriba para obtener una explicación. Tenga en cuenta que las expresiones son compatibles y, por ejemplo, puede establecer el tamaño de un componente "width pref+10px" para que sea 10 píxeles más grande que lo normal o "width max(100, 10%)" para que sea el 10% del ancho del contenedor, pero un máximo de 100 píxeles.	"width 10!" "width 10" "h 10:20" "height pref!" "w min:100:pref" "w100!,h100!" "width visual.x2-pref"
<u>wmin/wmax</u> <i>x-size</i> <u>hmin/hmax</u> <i>y-size</i>	Sobrescribe el tamaño predeterminado del componente para el tamaño mínimo o máximo establecido por el delegado de UI o por el desarrollador explícitamente en el componente. El tamaño se especifica como BoundSize . Consulte la sección de <i>Tipos de argumentos comunes</i> más arriba	"wmin 10" "hmax pref+100"

	<p>para obtener una explicación. Tenga en cuenta que las expresiones son compatibles y, por ejemplo, puede establecer el tamaño de un componente "wmin pref-10px" para que sea no menos de 10 píxeles más pequeño que lo normal. Estas palabras clave son cortos sintácticos para "width size:pref" o "width min:pref:size" con exactamente lo mismo para el mínimo y el máximo, respectivamente.</p>	
<p>grow [weightx] [weighty] growx [weightx] growy [weighty]</p>	<p>Establece qué tan interesante debe ser el componente para crecer en relación con otro componente en la misma celda. El peso (por defecto es 100 si no se especifica) es puramente un valor relativo para el peso de otros componentes. El doble de peso duplicará el espacio extra. Si no se establece esta restricción, el peso de crecimiento se establece en 0 y el componente no crecerá (a menos que fillse establezca en la fila / columna, en cuyo caso "grow 0" se puede usar para hacer que no crezca explícitamente). El peso de crecimiento solo se comparará con los pesos en el mismo grupo de prioridad de crecimiento y para la misma celda. Vea abajo.</p>	<p>"grow 50 20" "growx 50" "grow" "growx" "growy 0"</p>
<p>growprio/gp prio growprio/gpx prio growprio/gpy prio</p>	<p>Establece la prioridad de crecimiento para el componente. Al crecer, todos los componentes con mayores prioridades crecerán hasta su tamaño máximo antes de considerar cualquier componente con menor prioridad. La prioridad predeterminada de crecimiento es 100. Esta restricción se puede usar para hacer que ciertos componentes crezcan al máximo antes de que otros componentes comiencen a crecer.</p>	<p>"growprio 50 50" "gp 110 90" "gpx 200" "growprio 200"</p>
<p>shrink weightx [weighty]</p>	<p>Establece qué tan entusiasta / renuente debería ser el componente a encogerse en relación con otros componentes. El peso es puramente un valor relativo al peso de otros componentes. El doble de peso se reducirá dos veces cuando el espacio sea escaso. Si no se establece esta restricción, el peso de contracción predeterminado es 100, lo que significa que todos los componentes de</p>	<p>"shrink 50" "shrink 50 50 "</p>

	forma predeterminada pueden reducirse a su tamaño mínimo, pero no menos. El peso de contracción solo se comparará con los pesos en el mismo grupo de prioridad de contracción (otros componentes con la misma prioridad de contracción). Vea abajo.	
<u>shrinkprio/shp</u> <i>priox [prio]</i> <u>shrinkprio/shpxprio</u> <i>x</i> <u>shrinkprio/shpyprio</u> <i>y</i>	Establece la prioridad de contracción para el componente. Cuando el espacio es escaso y los componentes deben ser reducidos, todos los componentes con mayor prioridad se reducirán a su tamaño mínimo antes de considerar cualquier componente con menor prioridad. La prioridad de contracción predeterminada es 100. Esto se puede usar para hacer que ciertos componentes se reduzcan a min antes de que otros comiencen a contraerse.	"shrinkprio 50" "shp 200 200" "shpx 110"
<u>sizegroup/sg</u> [name] <u>sizegroupx/sgx</u> [name]] <u>sizegroupy/sgy</u> [name]]	Le da al componente un nombre de grupo de tamaño. Todos los componentes que comparten un nombre de grupo de tamaño obtendrán el mismo BoundSize (min / preferred / max). Se utiliza para garantizar que todos los componentes en el mismo grupo de tamaños obtengan el mismo tamaño min / preferred / max, que es el del componente más grande del grupo. Se "" puede usar un nombre vacío .	"sg" "sg group1" "sizegroup props" "sgx" "sizegroupy grp1"
<u>endgroup/eg</u> [name] <u>endgroupx/egx</u> [name]] <u>endgroupy/egy</u> [name]]	Le da al componente un nombre y asociación de grupo final. Todos los componentes que comparten un nombre de grupo final alinearán el lado derecho / inferior del componente. El lado derecho / inferior será el del componente más grande del grupo. Si "eg" o "endgroup" se utiliza y por lo tanto no se especifica la dimensión será utilizada la dimensión flujo de corriente (ver "flowx"). Entonces "eg" será lo mismo que "egx" en el caso normal. Se "" puede usar un nombre vacío .	"eg" "eg group1" "endgroup props" "egx" "endgroupy grp1"
"eg" "eg group1" "endgroup props" "egx" "endgroupy grp1"	Especifica la brecha entre los componentes en la celda o en el borde de la celda, según lo que esté alrededor de este componente. Si falta un tamaño de hueco, se interpreta como 0px. Los espacios se especifican como BoundSize . Véase más arriba.	"gap 5px 10px" "gap unrel rel" "gapx 10:20:50" "gapx 0:rel:null" "gap 10! 10!"

apx <i>left [right]</i> gapy <i>top [bottom]</i>	Especifica el espacio horizontal o vertical entre los componentes en la celda o en el borde de la celda, según lo que se encuentre alrededor de este componente. Si falta un tamaño de hueco, se interpreta como 0px. Los espacios se especifican como BoundSize . Véase más arriba.	"gapx 5px 10px" "gapy unrel rel"
id [<i>groupid.</i>] id	Establece el id (o nombre) para el componente. Si id no se especifica ComponentWrapper.getLinkId() , se usa el valor. Este valor le dará al componente una manera de ser referenciado desde otros componentes. Dos o más componentes pueden compartir el group id pero id deben ser únicos dentro de un diseño. El valor se convertirá en minúsculas y, por lo tanto, no es sensible a mayúsculas y minúsculas. No debe haber un punto primero ni último en la cadena de valores.	"id button1" "id grp1.b1"
pos <i>x y [x2] [y2]</i>	<p>osiciona el componente con coordenadas absolutas relativas al contenedor. Si se utiliza esta palabra clave, el componente no se colocará en una celda de la grilla y, por lo tanto, no afectará el flujo en la grilla. Uno de cualquiera x/ x2y uno de y/ y2 no debe ser null. La coordenada que se establece en null se colocará de manera que el componente obtenga su tamaño preferido en esa dimensión. Los valores no especificados se establecerán en null, por ejemplo, "abs 50% 50%" es lo mismo que "abs 50% 50% null null". Si la posición y el tamaño se pueden determinar sin referencias al tamaño de los contenedores principales, afectará el tamaño preferido del contenedor.</p> <p>Ejemplo: "pos 50% 50% n n" or "pos 0.5al 0.5al" or "pos 100px 200px" or "position n n 200 200".</p> <p>Las posiciones absolutas también pueden enlazar a los límites de otros componentes usando su ids o groupIds. Incluso puede usar expresiones alrededor de estos enlaces. Por ejemplo, "pos (butt.x+indent) butt1.y2" colocará el componente</p>	<p>"pos (b1.x+b1.w/2) (b1.y2+rel)"</p> <p>"pos (visual.x2-pref) 200"</p> <p>"pos n b1.y b1.x-rel b1.y2"</p> <p>"pos 100 100 200 200"</p>

	<p>directamente debajo del componente con id "butt1", con sangría ligeramente hacia la derecha. Hay dos límites especiales que siempre se establecen. "container" se establecen en los límites si el contenedor y "visual" se establecen en los límites del contenedor menos las inserciones especificadas. Las coordenadas que se pueden utilizar para estos enlaces son:</p> <ul style="list-style-type: none"> • .x o .y : la coordenada superior izquierda de los límites del componente al que se hace referencia • .x2 o .y2 - La coordenada inferior derecha de los límites del componente referenciado • .w or .h : el ancho y la altura actuales del componente al que se hace referencia. • .xpos o .ypos : la coordenada superior izquierda del componente al que se hace referencia en las coordenadas de la pantalla . 	
tag [name]	<p>Marca el componente con el nombre de metadato que puede usar el motor de diseño. La etiqueta se puede usar para explicar al administrador de diseño qué muestran los componentes, como un botón Aceptaro Cancelar . Las etiquetas desconocidas serán descartadas sin error ni indicación alguna.</p> <p>Actualmente, las etiquetas reconocidas se utilizan para red denominar botones por plataforma. Consulte JavaDoc for PlatformConverter.getButtonBarOrder(int type) para obtener una explicación más larga.</p> <p>Las etiquetas compatibles son:</p> <ul style="list-style-type: none"> • ok - Un botón OK. • cancel - Un botón Cancelar. • help - Botón de ayuda que normalmente está a la derecha. 	<p>"tag ok"</p> <p>"tag help2"</p>

	<ul style="list-style-type: none"> • help2 - Botón de ayuda que en algunas plataformas se coloca a la izquierda. • yes - Un botón Sí. • no - Un botón No. • apply - Un botón Aplicar. • next - Un botón Siguiente o Adelante. • back - Un botón Anterior o Atrás. • finish - Un botón finalizado. • left - Un botón que normalmente siempre debe colocarse en el extremo izquierdo. • right : un botón que normalmente siempre debe colocarse en el extremo derecho. 	
<u>x</u> x <u>x2</u> x2 <u>y</u> y <u>y2</u> y2	<p>Se usa para posicionar el inicio (x o y), el final (x2 o y2) o ambos bordes de un componente en coordenadas absolutas. Esto se utiliza para cuando un componente se encuentra en una grilla o un dock y, por ejemplo, necesita ajustarse para alinearse con otra cosa o de alguna otra manera estar completamente posicionado. La celda en la que se encuentra el componente no cambiará de tamaño, tampoco lo hará la cuadrícula. Las palabras clave x, y, x2 y y2 se aplican en la última etapa y, por lo tanto, no afectarán a otros componentes en la grilla o el dock, a menos que estén explícitamente vinculados a los límites del componente. Si la posición y el tamaño se pueden determinar sin referencias al tamaño de los contenedores principales, afectará el tamaño preferido del contenedor.</p>	"x button1.x" "x2 (visual.x2-50)1" "x 100, y 300"
<u>dock</u> ("north" "west" "south" "east") o <u>north</u> / <u>west</u> / <u>south</u> / <u>east</u>	<p>Se utiliza para acoplar el componente en un borde, o el centro, del contenedor. Funciona de forma muy parecida, BorderLayout excepto que puede haber una cantidad arbitraria de componentes de acoplamiento. Consiguen el espacio atracado en el orden en que se agregan al contenedor y "corta esa pieza". La "dock" palabra clave se puede</p>	"dock north" "north" "west, gap 5"

	omitir para todos pero solo está disponible para mayor claridad. El componente se pondrá en células circundantes especiales que se extiende por el resto de las filas que significa que la restricción de acoplamiento se puede combinar con muchas otras limitaciones tales como , , y ."center"paddingwidthheightgap	
pad <i>top [left]</i> <i>[bottom] [right]</i>	Establece el relleno para el componente en píxeles absolutos. Este es un ajuste absoluto de los límites si el componente se hace en la última etapa del proceso de disposición. Esto significa que no afectará las lagunas o el tamaño de la celda ni moverá otros componentes. Se puede usar para compensar algo que por alguna razón es difícil de hacer con las otras restricciones. Por ejemplo "ins -5 -5 5 5", agrandará el componente cinco píxeles en todas las direcciones, lo que lo hará 10 píxeles más alto y más ancho. Si se omiten los valores, se establecerán en 0. ¡Nota! Relleno de componentes de varias líneas derivados de JTextComponent (como JTextArea) sin establecer un tamaño mínimo explícito puede dar lugar a una escalada de tamaño continuo (¡animado!). Esto no es un error en el administrador de diseño, sino una "característica" derivada de cómo estos componentes calculan su tamaño mínimo. Si el tamaño se rellena de modo que aumente en un píxel, el componente de texto emitirá automáticamente una revalidación y el ciclo de diseño se reiniciará, ahora con un nuevo tamaño aumentado como el nuevo tamaño mínimo . Esto continuará hasta que se alcance el tamaño máximo. Esto solo ocurre para los componentes que tienen configurado el "ajuste de línea" true.	"padding 10 10" "pad 5 5 -5 -5" "pad 0 0 1 1"
al/align <i>alignx[aligny]</i> alignx/ax <i>alignx</i> aligny/ay <i>aligny</i>	Especifica la alineación para el componente si la celda es más grande que el componente más sus espacios. La alineación se puede especificar como UnitValue o AlignKeyword . Véase más arriba. Si se	"align 50% 50%" "aligny top" "alignx leading" "align 100px" "top, left"

	<p>usa AlignKeyword, la "align" palabra clave puede ser omitida. En una celda donde hay más de un componente, el primer componente puede establecer la alineación para todos los componentes. No es posible, por ejemplo, configurar el primer componente para alinear a la izquierda y el segundo para alinear a la derecha y así obtener un espacio entre ellos. Ese efecto se puede lograr mejor al establecer un espacio entre los componentes que tienen un tamaño mínimo y un tamaño preferido grande. Tenga en cuenta que la alineación de línea de base no funciona si el componente no puede obtener su tamaño preferido en la dimensión vertical.</p>	"aligny baseline"
<u>external</u>	<p>Inhibe MigLayout para cambiar los límites para el componente. Los límites deben manejarse / configurarse desde un código fuera de este administrador de diseño llamando al setBounds(..)(o equivalente dependiendo del kit de herramientas GUI utilizado) directamente en el componente. Los límites de este componente aún pueden vincularse con otros componentes si tiene una "id"etiqueta, o si el enlace proporciona una identificación de enlace ComponentWrapper. Esta es una forma muy simple y poderosa de ampliar los usos de MigLayout y reducir el número de veces que se debe escribir un administrador de diseño personalizado. El código de aplicación normal se puede usar para establecer los límites, algo que no se puede hacer con ningún otro administrador de diseño.</p>	"external" "external,id butt"
<u>hidemode</u>	<p>Establece el modo oculto para el componente. Si se ha especificado el modo oculto en el Este modo oculto puede ser anulado por la restricción del componente. El modo ocultar especificó cómo el administrador de diseño debe manejar un componente que no está visible. Los modos son: 0 - Predeterminado. Significa que los componentes invisibles se manejarán</p>	"hidemode 1"

	<p>exactamente como si fueran visibles.</p> <p>1- El tamaño del componente (si es invisible) se establecerá en 0, 0.</p> <p>2 - El tamaño del componente (si es invisible) se establecerá en 0, 0 y los espacios también se establecerán a su alrededor.</p> <p>3- Los componentes invisibles no participarán del diseño en absoluto y, por ejemplo, no ocuparán una celda de la grilla.</p>	
--	--	--

Conclusión:

Luego de la realizar la investigación he aprendido cada significado y su uso, también he introducción ejemplos de cómo podemos utilizarlos, para así cuando lo necesite tendré una noción clara el funcionamiento de cada una de las propiedades.

Bibliografía:

MigLayout - Java Layout Manager for Swing, SWT and JavaFX. Recuperado de:
<http://www.migcalendar.com/miglayout/>

Grev M. Creating Simple to Advanced Swing and SWT Layouts Easily with MiG Layout. Recuperado de: <http://www.oracle.com/technetwork/systems/ts-4928-159120.pdf>

Documentation MigLayout Recuperado de:
<https://www.formdev.com/jformdesigner/doc/layouts/miglayout/>

MigLayout. Recuperado de: <http://migcalendar.com/miglayout/whitepaper.html>