

Sumario <https://github.com/bpstahlman/txtfmt>

1. Txtfmt (El resaltador Vim).....	2
1.1. ¡Resaltado de "texto enriquecido" en Vim! (color del texto, resaltado, negrita, subrayado, cursiva, etc.).....	2
2. Introducción.....	2
3. Mapas automáticos.....	3
3.1. Mapas automáticos visuales.....	3
3.2. Mapas automáticos del Usuario.....	3
4. Especificaciones básicas de color.....	3
4.1. Especificaciones de color.....	4
4.2. Formato de especificaciones de atributos.....	4
4.3. Indicadores de atributo de formato de resaltado.....	4
4.4. Ejemplos de especificaciones de atributos de resaltado.....	5
4.5. Resaltado selectivo (basado en patrones).....	5
4.6. Ejemplo de patrones de selector.....	6
4.7. Anulación de operador de eliminación "inteligente".....	7
5. Sangría inicial (resaltado vs relleno de página/párrafo).....	7
5.1. Anulaciones de sangría / mayúsculas "inteligentes".....	8
5.2. "Smart": Volver a tabular comando.....	8
5.3. Mapas Manuales.....	8
5.4. Posicionamiento del cursor con punto (.).....	9
5.5. Ir a mapas de fichas.....	10
5.6. Inspección de tokens.....	10
6. Configuración.....	11
6.1. Opciones.....	11
6.2. Configuración de color.....	11
7. Rango de token.....	12
8. Visualización del rango de token.....	13
9. Instalación.....	13
10. Carga del complemento.....	13
10.1. Tipo de archivo.vim personalizado.....	14
10.2. Archivo específico del complemento en ftdetect.....	14
10.3. Modeline.....	14
10.4. Combinando Txtfmt con otros tipos de archivos.....	14

1. Txtfmt (El resaltador Vim)

1.1. ¡Resaltado de "texto enriquecido" en Vim! (color del texto, resaltado, negrita, subrayado, cursiva, etc.)

Nota: Si desea probar los ejemplos siguientes, primero deberá instalar el complemento. Si no está familiarizado con la instalación de complementos, lea *Instalación*. La sección sobre *Carga del complemento* puede ser útil si desea asegurarse de que el complemento `Txtfmt` se cargue automáticamente cada vez que abra ciertos archivos o tipos de archivos. Si tiene prisa por probar los ejemplos, puede ejecutar lo siguiente en la línea de comando de `Vim` para habilitar el resaltado en el búfer actual:

```
set ft=txtfmt
```

Alternativamente, puede hacer que `Txtfmt` cree una "página de prueba" para usted:

```
MakeTestPage
```

La página de prueba ofrece una forma rápida de verificar que el complemento esté instalado correctamente, así como un mecanismo conveniente para visualizar los efectos de los cambios de configuración. Cuando se ejecuta sin argumentos, `:MakeTestPage` usa la configuración predeterminada, teniendo en cuenta cualquier variable de opción de `Txtfmt` que haya establecido en su `vimrc`. Pero si proporciona argumentos, `:MakeTestPage` los procesará como si fueran parte de una "`Txtfmt modeline`" dentro de la página de prueba.

```
:help txtfmt-:MakeTestPage
```

2. Introducción

`Txtfmt` trae el resaltado de texto enriquecido a archivos de texto sin formato. Los comandos utilizados para lograr el resaltado se dividen en 2 categorías básicas:

- **Mapas automáticos:** cambia el resaltado de un rango de texto. Actualmente existen 2 tipos de mapas automáticos:
 - Visual: opera sobre el texto seleccionado visualmente
 - Operador: opera sobre el texto movido (o incluido en un "objeto de texto")
- **Mapas manuales:** inserta uno o más tokens de resaltado, que afectan a todo el texto posterior hasta el siguiente token.

Nota: El resaltado de `Txtfmt` se basa en caracteres invisibles ("tokens") en el texto, pero no necesita saberlo ni preocuparse por esto si está usando mapas automáticos, que automatizan completamente la inserción/eliminación de tokens. Los mapas manuales (el único tipo disponible hasta `Txtfmt 3.0`) no protegen al usuario tan bien de este detalle de implementación y, por lo tanto, se proporcionan principalmente para compatibilidad con versiones anteriores.

Nota: Todos los ejemplos en esta página asumen que su `<LocalLeader>` está en el valor predeterminado (es decir, barra invertida). Si ha configurado `<LocalLeader>` en un valor diferente al predeterminado, reemplace la barra invertida en los ejemplos con la secuencia de teclas adecuada.

```
:help <LocalLeader>
```

3. Mapas automáticos

Antes de profundizar en los detalles de los mapas automáticos, presentaré algunos ejemplos que muestran lo fáciles que son de usar.

3.1. Mapas automáticos visuales

Objetivo: convertir el texto seleccionado en azul, negrita y cursiva.

1. Seleccione el texto que se resaltará usando el mouse o el modo visual (p. Ej., Modo normal `v` o `V` comando).
2. Escriba `\h`.
3. Cuando se le solicite, escriba `f=bi,cb`.
4. Presiona Enter para aplicar el resaltado.

Nota: También podría haber escrito `fbi,cb` (sin el `=`) o incluso `f+bi,cb`, cualquiera de los cuales habría agregado negrita-cursiva encima de cualquier atributo de formato existente.

3.2. Mapas automáticos del Usuario

Objetivo (parte 1): Ponga la palabra debajo del cursor en rojo negrita.

- 1 En el modo normal, coloque el cursor en la palabra a resaltar.
- 2 Escriba `\h` (ingresa al modo `operator-pending` "pendiente de operador").
- 3 Escriba `iw` (especifica el objeto de texto `inner word` "palabra interna").
- 4 En el símbolo del sistema, escriba `cr,f=b`.
- 5 Presiona Enter para aplicar el resaltado.

Objetivo (parte 2): Poner en cursiva y subrayar las 2 líneas actual y las siguientes, resaltando el fondo en verde (sin perder el resaltado aplicado a la palabra en la parte 1).

- 1 En el modo Normal, con el cursor todavía en la línea del ejemplo anterior ...
 - 1.1 Escriba `\h` (ingresa al modo `operator-pending` "pendiente de operador").
- 2 Hit `2j` (incluye la línea actual y las 2 siguientes en el rango).
- 3 Cuando se le solicite, escriba `fui,kg`.

Presiona Enter para aplicar el resaltado.

Nota: `fui` es la abreviatura de `f+ui`, que agrega los atributos subrayado y cursiva al resaltado existente. Por lo tanto, la palabra que resaltó en el ejemplo anterior ha conservado el atributo negrita y ahora está subrayado en negrita y cursiva. Si hubiera usado `f=` en lugar de `f+`, el atributo en negrita se habría perdido, ya que `f=` reemplaza o *sobrescribe* el resaltado existente.

4. Especificaciones básicas de color

Las cadenas como `f=bi,cb` y `fui` se conocen como especificaciones de resaltado (`highlighting specs`). Cada especificación puede ser una lista separada por comas o espacios de componentes de formato/color (`Format/color`), por lo que es posible alterar el color de primer plano, el color de fondo y los atributos de formato con un solo comando. La primera letra de cada componente determina el tipo de resaltado afectado:

Letra	Tipo
f	atributos de formato (<code>format attriutes</code>)
c	color de primer plano (texto) (<code>foreground (text) color</code>)
k	color de fondo (resaltado) (<code>background (highlighting) color</code>)

Nota: Debido a que los 3 tipos de resaltado son completamente ortogonales, no es necesario especificar componentes cuyo resaltado no desea cambiar: por ejemplo, `fb` agrega un atributo en negrita sin afectar el color de primer plano o de fondo de ninguna manera.

4.1. Especificaciones de color

Para los colores de *primer-plano/fondo*, el nombre de un color o un guión (' - ') debe seguir a la `c` o `k`. Los nombres de los colores son configurables, y los valores predeterminados permiten que los colores comunes se abrevian a una sola letra: p. Ej., `R=red` (rojo), `b=blue` (azul), `g=green` (verde), etc. A ' - ' especifica "sin color": es decir, quita el color de la región. Tanto los valores de color RGB como los nombres de los colores son completamente configurables. Consulte Configuración de color para obtener más detalles ...

4.2. Formato de especificaciones de atributos

La `f` de una especificación de formato va seguida de una cadena de operadores y banderas de atributos de formato de un solo carácter:

Operador	Descripción
<code>+</code> (<i>default</i>)	Agregar a los atributos de formato existentes
<code>-</code>	Eliminar de los atributos de formato existentes
<code>=</code>	Reemplazar/sobrescribir atributos de formato existentes

Cada operador se aplica a todos los indicadores de atributo de formato subsiguientes hasta el siguiente operador.

4.3. Indicadores de atributo de formato de resaltado

Operador	Descripción
<code>u</code>	<code>underline</code> (subrayado)
<code>b</code>	<code>bold</code> (negrita)
<code>i</code>	<code>italic</code> (cursiva)
<code>s</code>	<code>destacar</code> (destacar) Solamente: <code>long tokrange</code>
<code>r</code>	<code>reverse</code> (invertir)

	Solamente: <code>long tokrange</code>
<code>c</code>	<code>undercurl</code> (rizado bajo la línea) Solamente: <code>long tokrange</code>

Nota: La configuración predeterminada solo proporciona subrayado, negrita y cursiva. Si desea habilitar el sobresaliente, el reverso y el rizado, consulte la sección Rango de tokens a continuación.

4.4. Ejemplos de especificaciones de atributos de resaltado

Operador	Descripción
<code>fbi</code>	Agregar <i>negrita-cursiva</i>
<code>f+bi</code>	Agregar <i>negrita-cursiva</i>
<code>f+bi-u</code>	Agregar <i>negrita-cursiva</i> y elimine el subrayado
<code>f-u+bi</code>	Agregar <i>negrita-cursiva</i> y elimine el subrayado
<code>f=bi</code>	Reemplace cualquier atributo de formato existente con <i>negrita-cursiva</i>
<code>f-,c-</code>	Eliminar todos los atributos de formato y cualquier color de primer plano
<code>fu-b,cb,kr</code>	Agregue <u>subrayado</u> y elimine negrita, haga que el color del texto sea azul y el color de fondo rojo
<code>fu-b cb kr</code>	Agregue <u>subrayado</u> y elimine negrita, haga que el color del texto sea azul y el color de fondo rojo

4.5. Resaltado selectivo (basado en patrones)

Hasta este punto, hemos estado aplicando resaltado a todo el texto en un rango. También es posible apuntar a subregiones específicas dentro del texto seleccionado u operado visualmente. Para aplicar el resaltado de forma selectiva, agregue un `/` a la especificación de resaltado, seguido de una "expresión de patrón de selector": p. Ej.,

`highlighting-spec / selector-pattern`

Ejemplo de uso: suponga que después de resaltar muchas palabras y frases en su documento con `f=b,cr,kg` (**negrita, texto rojo sobre fondo verde**), decide que `f=bi,cb` (**negrita-cursiva, texto azul**) habría sido una mejor elección. Hacer el cambio manualmente sería tedioso; afortunadamente, los patrones de selector proporcionan una mejor manera:

Seleccione un rango de líneas que contenga todo el texto que desea cambiar.

1. Escriba `\h`
2. En el símbolo del sistema, escriba `f+i,cb,k- / fb & cr & kg`.

Nota: `f=bi,cb,k- / fb & cr & kg` habrían logrado lo mismo.

3. Presiona Enter para aplicar el resaltado.

Explicación: La especificación de resaltado agrega cursiva, cambia el color del texto a azul y elimina el resaltado de fondo por completo, pero solo afecta al texto rojo en negrita sobre un fondo verde.

Los patrones de selección son esencialmente expresiones booleanas que combinan especificaciones de formato / color con operadores lógicos estándar:

Operador	Descripción
[]	O lógico (se puede abreviar como)
&[&]	Y lógico (se puede abreviar como &)
!	Negación lógica
(...)	Grupo de subpatrones

Consejo: | y & son equivalentes a || y &&, respectivamente, pero para evitar confusiones con la especial f| y f& primitivas, algunos usuarios pueden preferir las formas largas.

Las especificaciones de color utilizadas en los patrones de selector son idénticas a las utilizadas para resaltar. Al igual que con las especificaciones de resaltado de formato, las especificaciones del selector de formato comienzan con operadores especiales que modifican su comportamiento:

Operador	Descripción
	Coincide con las regiones que contienen cualquiera de los atributos de formato posteriores
&	Coincide con las regiones que contienen todos los atributos de formato posteriores
= (default)	Coincide con las regiones que contienen <i>exactamente</i> (todos y solo) los atributos de formato posteriores

Nota: f&..., f|... y f=... son construcciones atómicas, que no permiten espacios incrustados.

4.6. Ejemplo de patrones de selector

Los siguientes ejemplos ilustran el uso de especificaciones de resaltado con patrones de selector adjuntos:

Especificación/Patrón	Acción	Se aplica a
fu/cr	Agregar subrayado	Texto rojo
cb/cr f&bi	Hacer que el texto sea azul	Texto que es rojo o tiene atributos en negrita y cursiva (y posiblemente otros)
fu,kg/cr&& bi	Agregue subrayado y haga que el fondo sea verde	Texto que es rojo y tiene ya sea atributos en negrita o cursiva (y posiblemente ambos)
fi/cr f=bu	Agregue cursiva	Texto en rojo o subrayado en negrita
Cb,kr,f-i/fbi&! (cr&k-)	Hacer que el texto sea azul sobre un fondo rojo y eliminar	texto en negrita-cursiva y no rojo sobre un fondo incoloro

	el atributo de cursiva	
--	------------------------	--

Sugerencia: `Txtfmt` generalmente ignora los espacios en blanco y se pueden usar para facilitar la lectura de las expresiones.

Nota: Los tipos de resaltado omitidos de la expresión del selector no están restringidos: por ejemplo, una expresión del selector de `fb` coincide con el texto en negrita de *cualquier* color de primer plano/fondo (incluido ninguno).

4.7. Anulación de operador de eliminación "inteligente"

Aunque `Txtfmt` hace que los tokens de resaltado sean invisibles, para `Vim` son simplemente caracteres en su búfer como cualquier otro: por lo tanto, los operadores de eliminación incorporados de Vim no hacen distinción entre texto y tokens. Para ver cómo esto puede ser problemático, considere el siguiente escenario ... Ejecuta `dd` para eliminar la primera línea de un bloque de texto resaltado. Aunque tenía la intención de eliminar solo el texto, el `dd` también eliminó los tokens de resaltado ocultos al comienzo del bloque; por lo tanto, ha eliminado inadvertidamente el resaltado de todo el bloque. Para mitigar este problema, `Txtfmt` proporciona un operador de eliminación "inteligente" (`\d`) tanto en modo Normal como Visual, que agregará, eliminará y reemplazará tokens según sea necesario para evitar cambios en el resaltado que sigue al texto eliminado.

5. Sangría inicial (resaltado vs relleno de página/párrafo)

Si alguna vez ha utilizado un procesador de texto, es posible que haya notado que cuando un bloque de texto tiene sangría, los atributos como el subrayado y el color de fondo no son visibles en el margen. Cuando se utilizan espacios iniciales o tabulaciones para sangrar bloques de texto en `Vim`, generalmente parece mejor no resaltarlos. Pero, ¿qué es exactamente la "sangría inicial"? ¿Todos los espacios en blanco principales? ¿Solo espacios? ¿Solo pestañas? ¿Alguna combinación de pestañas y espacios, según lo determinado por opciones como `'tabstop'` y `'shiftwidth'`? El valor de la opción `'leadindent'` de `Txtfmt` proporciona la respuesta a esta pregunta, como se muestra en la siguiente tabla:

Ajuste	Definición de sangría inicial
<code>none</code>	Sin tratamiento especial de espacios en blanco iniciales, que se resaltarán como cualquier otro texto. Tenga en cuenta que este era el comportamiento predeterminado hasta <code>Txtfmt</code> versión 4.0, que introdujo esta opción.
<code>space</code>	Secuencia más larga de espacios iniciales (ASCII 32).
<code>tab</code>	Secuencia más larga de pestañas iniciales (ASCII 9).
<code>white</code> (<i>default</i>)	Secuencia más larga de espacios en blanco iniciales (espacios y/o tabulaciones).
<code>smart</code>	La lógica compleja utiliza <code>'tabstop'</code> y <code>'shifwidth'</code> (<i>ancho de desplazamiento</i>) para determinar si una secuencia de tabulaciones iniciales y/o espacios se considera sangría inicial. Heurística: cualquier secuencia de espacios en blanco iniciales que podría generarse en una línea vacía mediante una secuencia de <code>CTRL-T</code> en el modo

Insertar o desplazamientos a la derecha (>>) en el modo Normal.

Advertencia: si usa la configuración "inteligente", debe usar constantemente mecanismos como `CTRL-T` y `>>` para insertar la sangría inicial: en particular, la inserción de tabulaciones literales para la sangría inicial puede producir resultados inesperados.

Nota: La lógica que reconoce la sangría inicial considera solo los espacios en blanco reales, ignorando los tokens de resaltado incrustados.

```
:help txtfmt-'leadingindent'
```

5.1. Anulaciones de sangría / mayúsculas "inteligentes"

Debido a que los tokens resaltados son invisibles, pero aparecen como texto normal para Vim, su presencia en la sangría inicial podría llevar a un comportamiento de shift/indent inesperado si Txtfmt no anula los comandos incorporados de shift/indent `<<`, `>>`, `CTRL-T` y `CTRL-D`. Las invalidaciones de Txtfmt comprenden el papel especial de los tokens en un búfer Txtfmt (así como las implicaciones de las distintas configuraciones de opciones 'leadindent'), y harán todo lo posible para asegurar que varios comandos shift/indent "hagan lo correcto".

5.2. "Smart": Volver a tabular comando

Por las razones descritas en la sección anterior, la presencia de tokens resaltados en la sangría inicial tiende a romper el comportamiento del comando builtin `:retab` de Vim. En consecuencia, Txtfmt proporciona un token `:Retab`, invocado como el builtin `:retab`, que intenta hacer que el retabbing funcione de manera sensata cuando la sangría inicial contiene tokens Txtfmt, teniendo en cuenta la opción 'leadindent' del búfer.

5.3. Mapas Manuales

Los mapas manuales se utilizan para insertar tokens de resaltado específicos en ubicaciones específicas del búfer. Cada token determina el color del texto, el color de fondo o los atributos de formato vigentes hasta el siguiente token del mismo tipo (posiblemente el token final ' - '). Con los mapas automáticos, simplemente especifica los cambios de resaltado deseados, y Txtfmt realiza automáticamente las inserciones y eliminaciones de tokens requeridas. Esta tarea es mucho más compleja de lo que parece. Para ver por qué, suponga que después de seleccionar un texto, usa un mapa automático con especificación resaltada `fb,cr,kg` (agregue **negrita**, **texto rojo**, **fondo verde**). Puede suponer que Txtfmt insertaría 6 tokens de la siguiente manera:

```
<kr><cb><fb> <selected text...> <k-><c-><f->
```

Pero esta es solo una de las muchas posibilidades. Tenga en cuenta que puede haber tokens *dentro* del texto seleccionado que deban eliminarse. Dejar un token `<cg>`, por ejemplo, sería perjudicial, ya que terminaría prematuramente la región de texto azul. Un token `<cb>` dentro de la región sería inofensivo pero redundante con el que está al principio de la región, por lo que realmente debería eliminarse. Pero, ¿qué pasa con los tokens de atributo de formato dentro de la región? No se pueden eliminar simplemente, ya que la especificación de resaltado solicitaba solo la adición de negrita, ¡no la eliminación de nada más! Un token `<fi>` dentro de la región, por lo tanto, debería ser reemplazado por un token `<fbi>` para satisfacer la solicitud sin alterar el resaltado existente. De manera similar, si el texto justo después de la región seleccionada originalmente estaba subrayado en cursiva, el mapa automático

necesitaría terminar la región con un token `<fui>` (no `<f->`), para evitar cambiar el resaltado del texto no seleccionado.

Los mapas automáticos proporcionan una abstracción que le permite pensar en términos de un cambio de resaltado deseado, protegiéndolo de los detalles a menudo tediosos involucrados en realizar el cambio. Entonces, ¿por qué querrías usar mapas manuales en lugar de mapas automáticos? La mayoría de los usuarios no lo harán, pero puede haber escenarios en los que el control explícito proporcionado por los mapas manuales sea útil. En consecuencia, aquí hay un tutorial rápido ...

Suponga que desea ingresar texto en verde ... Ejecute `\i`, e ingrese `cg` en el indicador (mnemotécnico: color verde):

El texto que escribe ahora es verde.

Mientras escribe texto verde, desea enfatizar una frase poniéndola en negrita y cursiva. Aún en el modo Insertar, ejecute `CTRL-\CTRL-\` e ingrese `fb` (o `fib`) en el indicador (**mnemotécnico**: formato **negrita cursiva**).

El texto que escribe ahora es **verde, negrita y cursiva**.

Ahora desea ingresar un texto con un fondo azul. Aún en el modo Insertar, ejecute `CTRL-\CTRL-\` nuevamente e ingrese `kb` en el indicador (**mnemotécnico**: fondo azul).

El texto que escribe ahora es **verde, negrita y cursiva sobre un fondo azul**.

En este punto, cualquiera de las regiones de resaltado abiertas se puede terminar con el correspondiente token de terminación (`f-`, `c-` o `k-`). Alternativamente, puede ingresar una especificación de resaltado sin terminador para cambiar el resaltado posterior sin terminar explícitamente la región anterior. Insertar un token `cr`, por ejemplo, cambiaría del color del texto verde al rojo.

Para terminar el ejemplo, terminemos las 3 regiones activas ejecutando `CTRL-\CTRL-\` desde el modo Insertar una última vez e ingresando `c-,f-,k-` (o `c- f- k-`) en el indicador (**mnemotécnico**: sin color, sin formato, sin color de fondo).

El texto que escribe ahora es texto simple y sin resaltar.

Nota: El ejemplo anterior usó insertar mapas de tokens `\i` y `CTRL-\CTRL-\` para insertar tokens resaltados. En realidad, hay 15 mapas de este tipo en total, pero son fáciles de recordar para un usuario de Vim por analogía con los comandos integrados de Vim para ingresar al modo de inserción: por ejemplo, `\i` inserta antes del cursor; `\a` inserta después del cursor; `\I` inserta al principio de la línea; etc ... Además, cada uno de esos comandos tiene una variante que comienza con `\v` (por ejemplo, `\vi`), que regresa al modo Normal después de insertar el token. Finalmente, `CTRL-\CTRL-\` le permite insertar un token sin salir del modo Insertar.

```
:help txtfmt-ins-tok-maps
```

5.4. Posicionamiento del cursor con punto (.)

Al igual que con las especificaciones de resaltado automático de mapas, se pueden concatenar múltiples especificaciones de formato/color en una lista separada por comas o espacios. Además, puede reemplazar una de las comas con un punto (.) Para especificar dónde debe colocarse el cursor en relación con los tokens insertados. Esta función proporciona una manera conveniente de ingresar los tokens de inicio y finalización de una región antes de comenzar a escribir el texto resaltado. Si, por ejemplo, desea ingresar **texto rojo en negrita sobre un fondo amarillo**, puede ingresar `cr fb ky. c-f- k-` para insertar todos los tokens a la vez, dejando el cursor entre los tokens de inicio y final, listo para ingresar el texto resaltado.

5.5. Ir a mapas de fichas

Mientras esté utilizando mapas automáticos, no debería tener que pensar mucho en los tokens de resaltado de `Txtfmt`: los tokens son invisibles siempre que la sintaxis de `Txtfmt` esté activa, y el complemento maneja automáticamente toda la inserción/eliminación de tokens. Sin embargo, si está utilizando mapas manuales, deberá trabajar con los tokens: específicamente, deberá insertarlos, reemplazarlos y eliminarlos. `Txtfmt` proporciona mapas para facilitar la inserción (`\i`, `\I`, `\a`, `\A`, `\o`, `\O`) y reemplazo (`\s`). Los operadores de eliminación integrados de `Vim` (por ejemplo, `x`, `X`, etc.) pueden usarse para eliminar tokens, pero dado que los tokens en sí son invisibles, `Txtfmt` proporciona un puñado de mapas para ayudarlo a **encontrar** los tokens que desea eliminar/cambiar. En conjunto, estos mapas se denominan mapas "Jump to Token" y todos se pueden utilizar tanto en modo normal como en modo pendiente de operador. Hay bastantes variantes que le permiten especificar la dirección del salto, el tipo de token buscado (color del texto, color de fondo, atributos de formato) ya sea en el token de destino o junto a él.

Mapa	Descripción
<code>]f</code>	reenviar al siguiente token de formato
<code>]c</code>	hacia atrás al token de color de texto anterior
<code>]a</code>	reenviar al siguiente token de cualquier tipo
<code>]tf</code>	reenviar a char antes del siguiente token de formato (mnemotécnico "till", hasta)
<code>[ea</code>	hacia atrás al <i>final</i> del token anterior de cualquier tipo (es decir, <code><f-></code> , <code><c-></code> o <code><k-></code>)
<code>]bf</code>	reenviar al siguiente <i>inicio</i> de formato de token (es decir, token que especifica atributos de formato, not <code><f-></code>)

```
:help txtfmt-jump-to-tok
```

5.6. Inspección de tokens

Si alguna vez necesita saber el tipo de un token, simplemente coloque el cursor sobre él (por ejemplo, usando uno de los mapas de "saltar al token" descritos anteriormente) y ejecute `... \ga`. Como `ga` y `g8` integrados, este comando muestra información en la línea de estado sobre el carácter debajo del cursor, pero en lugar de un código de carácter, muestra una cadena corta que indica el tipo de token.

Línea de estado	Significado
<code>fibu</code>	Dar formato a cursiva-negrita-subrayado.
<code>c1</code>	Color de primer plano #1.
<code>k3</code>	Color de fondo #3,
<code>f-</code>	Dar formato al token final.
<code>[ea</code>	hacia atrás al <i>final</i> del token anterior de cualquier tipo (es decir, <code><f-></code> , <code><c-></code> o <code><k-></code>)
<code>]bf</code>	reenviar al siguiente <i>inicio</i> de formato de token (es decir, token que especifica atributos de formato, not <code><f-></code>)

```
:help txtfmt-get-tok-info
```

6. Configuración

6.1. Opciones

Aunque la configuración predeterminada de la mayoría de las opciones de `Txtfmt` estará bien para el usuario promedio, existen 3 mecanismos distintos para la personalización:

Mecanismo	Descripción	Finalidad	Ejemplo
<code>Txtfmt</code> “modelines”	Análogas a los “modelines” de <code>Vim</code>	Asociar opciones con archivos.	<code>txtfmt: li=smart nested</code> <code>vim: ft=txtfmt</code>
Variables locales de búfer	La configuración se aplica solo al búfer actual.	Establecido desde un autocomando para un tipo de archivo específico.	<code>let b:txtfmtLeadingindent = 'white'</code>
Variables globales	La configuración se aplica a todos los búferes <code>Txtfmt</code> .	Establecido desde <code>vimrc</code> .	<code>let g:txtfmtTokrange = '180X'</code>

```
:help txtfmt-options
```

Para una lista de opciones soportadas:

```
:help txtfmt-opt-list
```

6.2. Configuración de color

Color	Color#	Abreviatura	Terminal	GUI	¿Disponible en segundo plano?
Black	1	k	dark	#000000	No
Blue	2	b	darkblue	#0000FF	Si
Green	3	g	darkgreen	#00FF00	Si
Turquoise	4	t	darkcyan	#00FFFF	No
Red	5	r	darkred	#FF0000	Si
Violet	6	v	darkmagenta	#FF00FF	No
Yellow	7	y	darkyellow	#FFFF00	No
White	8	w	white	#FFFFFF	No

Pero todo lo relacionado con estos valores predeterminados es configurable. Por ejemplo, si tiene una computadora relativamente moderna, es posible que desee activar los 8 colores para resaltar el fondo. Puede hacerlo con la siguiente anulación de 'bgcolormask' en su vimrc:

```
" Unmask background colors disabled by default
```

```
let g:txtfmtBgcolormask = "11111111"
```

Puede lograr lo mismo para un solo archivo agregando la siguiente "Txtfmt Linemode" al principio o al final del archivo: `txtfmt:bcm=11111111`.

Nota: Los colores de primer plano y de fondo se pueden habilitar/deshabilitar de forma independiente.

```
:help txtfmt-'fgcolormask'
:help txtfmt-'bgcolormask'
```

Las propias definiciones de color también se pueden personalizar. Si el negro es el color de texto predeterminado en su esquema de colores preferido, es posible que no desee desperdiciar 1 de los 8 espacios de color en él. Puede cambiar el color n.º 1 de negro a gris pizarra agregando la siguiente línea a su vimrc:

```
let g:txtfmtColor{1} = '^G$,c:Gray,g:#708090'
```

De manera similar, si el valor predeterminado para el color #6 (violeta) parece demasiado brillante en la GUI, puede atenuarlo y cambiar su abreviatura de "v" (violeta) a "p" (púrpura) agregando lo siguiente a su vimrc:

```
let g:txtfmtColor{6} = '^p$,c:DarkMagenta,g:#800080'
```

Nota: En realidad, no especifica un nombre para un color, sino una expresión regular que coincide con cualquier cadena que pueda usar para especificarlo en una especificación de resaltado. La construcción `\% []` se usa en las expresiones regulares predeterminadas para permitir que los nombres sean abreviados o escritos en parte o en su totalidad, pero la mayoría de los usuarios probablemente querrán usar abreviaturas de una sola letra.

Nota: La `c:` y `g:` en las cadenas de definición de color se refieren a `color terminal` (*color de terminal*) y `gui`, respectivamente. La sintaxis de cadena a todo color le permite definir diferentes colores para diferentes tipos de terminal/GUI.

```
:help txtfmt-color-config
```

7. Rango de token

`Txtfmt` requiere un bloque de caracteres contiguos para usar como tokens de resaltado. Tanto la ubicación como el tamaño del bloque están determinados por el valor de la opción '`tokrange`', que se puede configurar mediante la variable `Txtfmt option` o la línea de `Txtfmt modeline`. La opción '`tokrange`' es una cadena que consta de un número decimal o hexadecimal con un sufijo final: por ejemplo, "`0xE000X`" o "`180S`". El valor numérico es el código de carácter del primer carácter del bloque y el sufijo (que no distingue entre mayúsculas y minúsculas) determina el resaltado disponible, como se muestra en la siguiente tabla:

Sufijo tokrange	Resaltado admitido	Número de tokens en el rango
s (short)	underline,bold,italic + fg colors	17
s (extended) default	underline,bold,italic + fg colors + bg colors	26

L (Long)	<code>underline,bold,italic,standout,reverse</code> (and optionally <code>undercurl</code>) + fg colors	41 (73 si <code>undercurl</code> está habilitado)
----------	--	---

```
:help txfmt-'tokrange'
:help txfmt-'undercurl'
```

Si no configura la opción `'tokrange'` explícitamente, `Txfmt` elegirá un valor predeterminado que tenga sentido para la codificación en uso. Este valor predeterminado debería estar bien para la mayoría de los usuarios, pero puede cambiarlo fácilmente agregando algo como esto a su `vimrc` ...

```
" Place Txfmt tokens at 0xE100 (256 bytes past start of Unicode 'Private Use Area')
" and enable 'Long' formats for standout, reverse, etc.
let g:txfmtTokrange = '0xE100L'
```

Alternativamente, puede personalizar `'tokrange'` para un solo archivo usando una línea de modo `Txfmt`: por ejemplo,

```
txfmt:tokrange=0xE100X
```

Advertencia: los cambios en `g:txfmtTokrange` impactan el resaltado en **todos** los archivos `Txfmt` existentes sin un `modeline`; por lo tanto, cualquier cambio en `g:txfmtTokrange` globalmente debería ir acompañado de una migración de los archivos existentes. `Txfmt` proporciona el comando `:MoveStartTok` para ayudar con esto.

8. Visualización del rango de token

El comando `:ShowTokenMap` proporciona una ilustración tabular del `'tokrange'` en vigor para el búfer actual. Cada fila de la tabla enumera el código de carácter de un token junto con su significado (por ejemplo, `underline-bold` (*subrayado en negrita*) o `Color3`). Se muestra información adicional para un token de color: su patrón de nombre, valor RGB (ya sea como valor hexadecimal o nombre de color específico del terminal) y estado habilitado/deshabilitado.

```
:help txfmt-:ShowTokenMap
```

9. Instalación

`Txfmt` se puede instalar usando cualquiera de los mecanismos estándar de complementos de `Vim`: por ejemplo, `Pathogen`, `Vundle`, etc. Si no está utilizando un administrador de complementos, simplemente descomprima la distribución `Txfmt` en algún lugar de su `'runtimepath'` (por ejemplo, `~/vim`) y ejecuta `:helptags ALL` para preparar la ayuda de `Txfmt`.

10. Carga del complemento

Advertencia: como la mayoría de los complementos, `Txfmt` requiere que se habiliten tanto la detección del tipo de archivo como el resaltado de sintaxis, pero ninguno de los dos está habilitado en la configuración predeterminada de `Vim`; por lo tanto, a menos que haya copiado los comandos del ejemplo `vimrc` (o haya obtenido el ejemplo `vimrc` de su propio `vimrc`), probablemente necesitará agregar algo como lo siguiente a su `vimrc`:

```
syntax on
filetype plugin on
```

Suponiendo que el complemento se haya instalado correctamente, puede cargarlo manualmente ejecutando lo siguiente desde la línea de comandos de Vim:

```
:set ft=txtfmt
```

Sin embargo, un mejor enfoque es asegurarse de que Txtfmt se cargue automáticamente cada vez que abra ciertos archivos o tipos de archivos. A continuación se describen varios enfoques ...2

10.1. Tipo de archivo.vim personalizado

Si aún no existe, cree su propio filetype.vim en el primer directorio de 'runtimepath' (p. Ej., ~/.Vim) y agregue un autocomando que asocie txtfmt con las extensiones de archivo deseadas.

Ejemplo: -- filetype.vim -

```
augroup filetypedetect
```

```
" Treat files with the following extensions as Txtfmt files: .txt, .tf, .jnl
```

```
au! BufRead,BufNewFile *.txt,*.tf,*.jnl setfiletype txtfmt
```

```
augroup END
```

10.2. Archivo específico del complemento en ftdetect

Si aún no existe, cree su propia carpeta "ftdetect" en el primer directorio de 'runtimepath' (por ejemplo, ~/.vim). En ftdetect, cree un archivo llamado "txtfmt.vim" que contenga un comando automático que asocie txtfmt con las extensiones de archivo deseadas.

Ejemplo:

```
-- ftdetect/txtfmt.vim --
```

```
" Treat files with the following extensions as Txtfmt files: .txt, .tf, .jnl
```

```
au! BufRead,BufNewFile *.txt,*.tf,*.jnl set filetype=txtfmt
```

Nota: Cuando se usa ftdetect, no es necesario envolver sus autocomandos en un bloque agrupado, ya que su script proviene de un grupo de autocomandos.

```
:help new-filetype
```

10.3. Modeline

Modelines se pueden usar para configurar 'filetype=txtfmt' cuando los archivos que desea resaltar no comparten una extensión común. Simplemente agregue una línea como la siguiente al principio o al final de un archivo que desea resaltar: vim:ft=txtfmt

Nota: Deberá cerrar y volver a abrir el archivo para que se procese modline.

10.4. Combinando Txtfmt con otros tipos de archivos

Es muy posible utilizar el resaltado Txtfmt en un archivo asociado con otro complemento de tipo de archivo. Por ejemplo, si está utilizando un complemento que facilita la toma de notas o el diario, es posible que desee utilizar Txtfmt para resaltar las entradas. Incluso es posible "anidar" las regiones de resaltado de Txtfmt dentro de las regiones de sintaxis definidas por otros complementos, pero el anidamiento debe habilitarse explícitamente configurando la opción txtfmt-'nested'.

```
:help txtfmt-combining  
:help txtfmt-nesting
```

Columna1	Columna2
s (<i>default</i>)	Agregar a los atributos de formato existentes Agregar a los atributos de formato existentes
-	Eliminar de los atributos de formato existentes
=	Reemplazar/sobrescribir atributos de formato existentes
s (short)	

<https://github.com/bpstahlman/txtfmt>