

Manual do KatePart

**Thad McGinnis
Anne-Marie Mahfouf
Anders Lund
T.C. Hollingsworth
Christoph Cullmann
Lauri Watts**

**Tradução: Marcus Gama
Tradução: André Marcelo Alvarenga**



Manual do KatePart

Conteúdo

1	Introdução	8
2	Alguns Fundamentos	9
2.1	Arrastar e Soltar	9
2.2	Atalhos	9
3	Trabalhando com o Editor KatePart	13
3.1	Visão geral	13
3.2	Navegando no Texto	14
3.3	Trabalhando com Seleção	14
3.3.1	Usando a Seleção de Blocos	15
3.3.2	Usando Seleção de Sobrescrita	15
3.3.3	Usando Seleção Persistente	15
3.4	Copiando e Colando Texto	16
3.5	Procurando e Substituindo Texto	16
3.5.1	As Barras para Pesquisar e Substituir	16
3.5.2	Procurando Texto	17
3.5.3	Substituindo Texto	17
3.6	Usando os favoritos	18
3.7	Quebra de Texto Automática	18
3.8	Usando o recuo automático	19
3.9	Indicadores de modificação de linhas	20
3.10	O minimapa na barra de rolagem	20
3.11	Cursors múltiplos	21
3.11.1	Criando cursores múltiplos	21
3.11.2	Trabalhando com cursores múltiplos	21
4	Entradas de Menu	22
4.1	Menu Arquivo	22
4.2	Menu Editar	23
4.3	Menu Seleção	25
4.4	Menu Exibir	27
4.5	Menu Ir	28
4.6	Menu Ferramentas	29
4.7	Menus Configurações e Ajuda	32

5	Ferramentas Avançadas de Edição	33
5.1	Comentar/Descomentar	33
5.2	A Linha de Comando do Componente de Edição	33
5.2.1	Comandos Padrão da Linha de Comando	34
5.2.1.1	Comandos para configurar o Editor	34
5.2.1.2	Comandos de edição	36
5.2.1.3	Comandos de navegação	41
5.2.1.4	Comandos das funções básicas do editor (dependem do aplicativo em que o componente de edição for usado)	41
5.3	Usando o Desdobramento de Código	42
6	Ampliando o KatePart	43
6.1	Introdução	43
6.2	Trabalhando com Realce de Sintaxe	43
6.2.1	Introdução	43
6.2.2	O Sistema de Realce de Sintaxe do KatePart	44
6.2.2.1	Como funciona	44
6.2.2.2	Regras	45
6.2.2.3	Estilos de Contexto e Palavras-Chave	45
6.2.2.4	Estilos Padrão	45
6.2.3	O Formato XML de Definição de Realce	46
6.2.3.1	Introdução	46
6.2.3.2	As Seções em detalhes	49
6.2.3.3	Estilos Padrão Disponíveis	51
6.2.4	Regras de Detecção de Realce	52
6.2.4.1	As regras em detalhes	54
6.2.4.2	Dicas & Truques	58
6.3	Trabalhando com temas de cores	59
6.3.1	Introdução	59
6.3.2	O temas de cores do KSyntaxHighlighting	60
6.3.3	Formato JSON dos temas de cores	62
6.3.3.1	Introdução	62
6.3.3.2	A estrutura JSON	62
6.3.3.3	Seções principais dos arquivos de tema de cores JSON	63
6.3.3.4	Metadados	65
6.3.4	Cores em detalhes	65
6.3.4.1	Cores do editor	65
6.3.4.2	Estilos de texto padrão	72
6.3.4.3	Estilos de texto de realce personalizados	75
6.3.5	GUI dos temas de cores	75
6.3.5.1	Criar um novo tema	76
6.3.5.2	Importar ou exportar arquivos de tema JSON	76

6.3.5.3	Editando temas de cores	77
6.3.5.3.1	Cores	77
6.3.5.3.2	Estilos de texto padrão	77
6.3.5.3.3	Estilos de texto de realce	77
6.3.6	Dicas & Truques	78
6.3.6.1	Contraste das cores do texto	78
6.3.6.2	Sugestões de consistência com o realce de sintaxe	78
6.4	Scripts com JavaScript	78
6.4.1	Scripts de recuo	78
6.4.1.1	O cabeçalho do script de recuo	79
6.4.1.2	O código-fonte do recuador	79
6.4.2	Scripts de Linha de Comando	81
6.4.2.1	O Cabeçalho do Script de Linha de Comando	81
6.4.2.2	O Código-Fonte do Programa	82
6.4.2.2.1	Associar combinações de teclas	82
6.4.3	API de Programação	83
6.4.3.1	Cursors e Intervalos	84
6.4.3.1.1	O Protótipo do Cursor	84
6.4.3.1.2	O Protótipo do Intervalo	85
6.4.3.2	Funções globais	86
6.4.3.2.1	Arquivos de leitura e inclusão	86
6.4.3.2.2	Depuração	86
6.4.3.2.3	Tradução	86
6.4.3.3	A API do View	87
6.4.3.4	API do Document	89
6.4.3.5	A API do editor	94
7	Configurar o KatePart	95
7.1	Configuração do Componente de Edição	95
7.1.1	Aparência	95
7.1.1.1	Geral	95
7.1.1.2	Bordas	97
7.1.2	Temas de cores	98
7.1.3	Edição	98
7.1.3.1	Geral	98
7.1.3.2	Navegação de texto	99
7.1.3.3	Recuo	100
7.1.3.4	Complementação automática	101
7.1.3.5	Verificação ortográfica	102
7.1.3.6	Modo de entrada Vi	102
7.1.4	Abrir/Salvar	102

Manual do KatePart

7.1.4.1	Geral	102
7.1.4.2	Avançado	103
7.1.4.3	Modos & Tipos de Arquivos	104
7.2	Configurando com as Variáveis do Documento	105
7.2.1	Como o KatePart usa Variáveis	106
7.2.2	Variáveis Disponíveis	106
7.2.3	Opções estendidas nos arquivos <code>.kateconfig</code>	109
8	Créditos e licença	110
9	O modo de entrada do VI	111
9.1	Modo de entrada do VI	111
9.1.1	Incompatibilidades com o Vim	111
9.1.2	Alternando modos	112
9.1.3	Integração com as funcionalidades do Kate	112
9.1.4	Comandos suportados nos modos normal/visual	113
9.1.5	Movimentos suportados	114
9.1.6	Objetos de texto suportados	116
9.1.7	Comandos de modo de inserção suportados	116
9.1.8	O Objeto de Texto Vírgula	117
9.1.9	Funcionalidades que faltam	117
A	Expressões Regulares	118
A.1	Introdução	118
A.2	Padrões	119
A.2.1	Escapando caracteres	119
A.2.2	Classes de Caracteres e abreviações	119
A.2.2.1	Caracteres com significado especial dentro das classes	121
A.2.3	Alternativas: combinando ‘um deles’	121
A.2.4	Subpadrões	121
A.2.4.1	Especificando alternativas	121
A.2.4.2	Capturando texto relacionado (referências anteriores)	121
A.2.4.3	Afirmações Adiante	122
A.2.4.4	Afirmações atrás	122
A.2.5	Caracteres com significado especial dentro de padrões	122
A.3	Quantificadores	123
A.3.1	Guloso	123
A.3.2	Exemplos do Contexto	124
A.4	Afirmações	124
B	Índice	126

Resumo

O KatePart é um componente de edição completo do KDE.

Capítulo 1

Introdução

KatePart é um componente de edição completo, usado por muitos aplicativos do Qt™ e do KDE. O KatePart é mais que um editor de texto, tem como objetivo ser o editor do programador e poderia ser considerado pelo menos como uma alternativa parcial para a maioria dos editores mais poderosos. Uma das principais funcionalidades do KatePart é a sintaxe colorida, personalizada para muitas das diferentes linguagens de programação, como por exemplo: C/C++, Java™, Python, Perl, Bash, Modula 2, HTML e Ada.

KWrite é um editor de texto simples baseado no KatePart. Tem uma interface de documentos individuais (SDI) que permite modificar apenas um arquivo por janela. Pelo fato de o KWrite ser uma implementação muito simples do KatePart, não precisa ter uma documentação própria. Se você souber usar o KWrite, poderá usar o KatePart em qualquer lugar!

Capítulo 2

Alguns Fundamentos

O KWrite e várias outras utilizações do KatePart são muito simples de usar. Qualquer pessoa que já usou um editor de texto não deverá ter problemas.

2.1 Arrastar e Soltar

O KatePart usa o protocolo Arrastar e Soltar do KDE. Os arquivos podem ser arrastados e depois soltos no KatePart, a partir da Área de Trabalho, do gerenciador de arquivos Dolphin ou de algum site FTP remoto que estiver aberto em uma das janelas do Dolphin.

2.2 Atalhos

Muitos dos atalhos de teclado são configuráveis através do menu [Configurações](#). Por padrão, o KatePart respeita os seguintes atalhos de teclado:

Ins	Alterna entre o modo Inserir e Sobrescrever. Quando no modo inserir o editor adicionará qualquer caractere digitado ao texto empurrando qualquer dado à direita do cursor do texto. O modo sobrescrever faz com que a entrada de cada caractere elimine o caractere imediatamente à direita do cursor de texto.
Esquerda	Move o cursor um caractere à esquerda.
Direita	Move o cursor um caractere à direita.
Acima	Move o cursor uma linha acima.
Abaixo	Move o cursor uma linha abaixo.
Ctrl+E	Ir para a linha de edição anterior.
Ctrl+Shift+E	Ir para o próximo local de edição no documento.
Alt+Shift+Acima	Move o cursor para o recuo correspondente anterior.
Alt+Shift+Abaixo	Move o cursor para o recuo correspondente anterior.
Ctrl+6	Move até o parênteses correspondente.

Page Up	Move o cursor uma página acima.
Page Down	Move o cursor uma página abaixo.
Home	Move o cursor para o início da linha.
End	Move o cursor para o fim da linha.
Ctrl+Home	Move até o início do documento.
Ctrl+End	Move até o fim do documento.
Ctrl+Acima	Rolar uma linha acima.
Ctrl+Abaixo	Rolar uma linha abaixo.
Ctrl+Direita	Mover palavra à direita.
Ctrl+Esquerda	Mover palavra à esquerda.
Ctrl+Shift+Acima	Mover as linhas para cima.
Ctrl+Shift+Abaixo	Mover as linhas para baixo.
Ctrl+.	Duplicar as linhas selecionadas para baixo.
Ctrl+B	Definir um favorito.
Alt+Page Up	Favorito anterior.
Alt+Page Down	Próximo favorito.
Del	Apaga o caractere à direita do cursor (ou o texto selecionado).
Backspace	Apaga o caractere à esquerda do cursor.
Ctrl+Del	Apagar palavra à direita.
Ctrl+Backspace	Apagar palavra à esquerda.
Ctrl+K	Excluir linha.
Shift+Enter	Insere uma nova linha incluindo os caracteres iniciais da linha atual, que não forem letras ou números. Isso é útil, por exemplo, para escrever comentários no código. No fim da linha '// algum texto', pressione esse atalho para que a próxima linha inicie com '// '. Assim, você não terá que inserir esses caracteres no início de cada nova linha com comentários.
Ctrl+Shift+Enter	Cria uma nova linha abaixo da linha atual.
Ctrl+Alt+Enter	Cria uma nova linha acima da linha atual.
Shift+Esquerda	Marca o texto que estiver um caractere à esquerda.
Shift+Direita	Marca o texto que estiver um caractere à direita.
Ctrl+F	Localizar.
F3	Localizar próxima.
Shift+F3	Localizar anterior.
Ctrl+H	Localizar selecionado.
Ctrl+Shift+H	Localizar selecionado para trás.
Ctrl+Shift+Direita	Selecionar palavra à direita.
Ctrl+Shift+Esquerda	Selecionar palavra à esquerda.
Shift+Home	Selecionar até o início da linha.
Shift+End	Selecionar até o fim da linha.
Shift+Acima	Selecionar até a linha anterior.
Shift+Abaixo	Selecionar até a linha seguinte.
Ctrl+Shift+6	Selecionar até o parênteses correspondente.
Ctrl+Shift+Page Up	Selecionar até o topo da visão.
Ctrl+Shift+Page Down	Selecionar até o fundo da visão.
Shift+Page Up	Selecionar página acima.
Shift+Page Down	Selecionar página abaixo.

Ctrl+Shift+Home	Selecionar até o início do documento.
Ctrl+Shift+End	Selecionar até o fim do documento.
Ctrl+Home	Selecionar tudo.
Ctrl+Shift+A	Desmarcar.
Ctrl+Shift+B	Modo de seleção em bloco.
Ctrl+C / Ctrl+Ins	Copia o texto marcado para a área de transferência.
Ctrl+D	Comentar.
Ctrl+Shift+D	Descomentar.
Ctrl+G	Vai para linha...
Ctrl+I	Recuar seleção.
Ctrl+Shift+I	Retirar o recuo da seleção.
Ctrl+J	Juntar linhas.
Ctrl+P	Imprimir.
Ctrl+R	Substituir.
Ctrl+S	Invoca o comando Salvar.
Ctrl+Shift+S	Salvar como.
Ctrl+U	Maiúsculas.
Ctrl+Shift+U	Minúsculas.
Ctrl+Alt+U	Iniciais maiúsculas.
Ctrl+V / Shift+Ins	Cola o texto da área de transferência na linha de edição.
Ctrl+X / Shift+Ins	Apaga o texto marcado e copia-o para a área de transferência.
Ctrl+Z	Desfazer.
Ctrl+Shift+Z	Refazer.
Ctrl+-	Diminuir fonte.
Ctrl++Ctrl+=	Aumentar fonte.
Ctrl+Shift+-	Dobrar os nós superiores.
Ctrl+Shift++	Desdobrar os nós superiores.
Ctrl+Espaço	Invocar completamento de código.
F5	Recarregar.
F6	Mostrar/Ocultar a borda dos ícones.
F7	Alternar para linha de comando.
F9	Mostrar/Ocultar as marcas de desdobramento.
F10	Quebra de linha dinâmica.
F11	Mostrar/Ocultar números de linha.
Ctrl+T	Transpor caracteres.
Ctrl+Shift+O	Verificação ortográfica automática.
Ctrl+Shift+V	Mudar para o próximo modo de entrada.
Ctrl+8	Reutilizar a palavra acima.
Ctrl+9	Reutilizar a palavra abaixo.
Ctrl+Alt+#	Expandir a abreviatura.
Ctrl+Alt+Acima	Adicionar um cursor acima do cursor atual.
Ctrl+Alt+Abaixo	Adicionar um cursor abaixo do cursor atual.
Shift+Alt+I	Criar um cursor no final de cada linha da seleção.
Alt+J	Localiza a próxima ocorrência da palavra sob o cursor e selecione-a.
Ctrl+Alt+Shift+J	Localiza todas as ocorrências da palavra sob o cursor e selecione-as.

Capítulo 3

Trabalhando com o Editor KatePart

Anders Lund
Dominik Haumann
Tradução: Lisiane Sztoltz

3.1 Visão geral

O editor KatePart é a área de edição da janela do KatePart. Este editor também é usado pelo Kate e pelo KWrite, e ele pode também ser usado pelo Konqueror para exibir arquivos texto de seu computador local ou da rede.

O editor é composto pelos seguintes componentes:

A Área do Editor

Esta é a área onde o texto de seu documento está localizado.

As barras de rolagem

Indicam a posição da parte visível do texto do documento, e pode ser usada para mover-se pelo documento. Arrastar as barras não fará com que o cursor de inserção seja movido.

As barras de rolagem podem ser exibidas e ocultas, caso necessário.

A Borda do Ícone

A borda do ícone é um pequeno painel do lado esquerdo do editor, exibindo um pequeno ícone próximo às linhas marcadas.

Você pode ajustar ou remover um [favorito](#) em uma linha visível clicando com o botão esquerdo do mouse na borda do ícone, próximo àquela linha.

A exibição da borda do ícone pode ser alternada usando o item de menu **Ver** → **Mostrar Borda do Ícone**.

O Painel dos Números de Linha

O painel de números de linha exibem os números de linha de todas as linhas visíveis no documento.

A exibição do Painel de Números de linha pode ser alternado usando o item de menu **Ver** → **Mostrar Números da Linha**.

Painel Dobragem

A área de dobragem/desdobramento permite-lhe expandir ou recolher blocos de linhas de código. O cálculo das regiões dobráveis é feito de acordo com regras na definição de realce de sintaxe do documento.

TAMBÉM NESTE CAPÍTULO:

- [Navegando no texto](#)
- [Trabalhando com seleção](#)
- [Copiando e colando o texto](#)
- [Procurando e substituindo texto](#)
- [Usando os favoritos](#)
- [Quebra de Texto Automática](#)
- [Usando o recuo automático](#)

3.2 Navegando no Texto

A movimentação pelo texto no KatePart é como na maioria dos editores de texto gráficos. O usuário poderá mover o cursor com as teclas dos cursores e com as teclas **Page Up**, **Page Down**, **Home** e **End**, em conjunto com as teclas modificadoras **Ctrl** e **Shift**. A tecla **Shift** é sempre usada para gerar uma seleção, enquanto a tecla **Ctrl** pode ter efeitos diferentes nas diferentes teclas:

- Para as teclas **Acima** e **Abaixo**, isto significa deslocar o cursor, em vez de movê-lo.
- Para as teclas **Esquerda** e **Direita**, isto significa saltar as palavras em vez de simples caracteres.
- Para as teclas **Page Up** e **Page Down**, isto significa ir para o limite visível da janela, em vez de navegar.
- Para as teclas **Home** e **End**, isto significa ir para o início ou fim do documento, em vez de ir para o início ou fim da linha.

O KatePart também lhe oferece uma forma de saltar rapidamente para um parênteses correspondente: Coloque o cursor no interior de um parênteses e pressione **Ctrl+6** para saltar para o parênteses correspondente.

Além disso, você poderá colocar ou retirar um [favorito](#), para saltar para posições definidas por você.

3.3 Trabalhando com Seleção

Existem duas maneiras básicas de selecionar texto no KatePart: usando o mouse e usando o teclado.

Para selecionar usando o mouse, pressione e mantenha pressionado o botão esquerdo do mouse enquanto arrasta o cursor do mouse de onde a seleção deve iniciar, até o ponto final desejado. O texto fica selecionado quando você arrasta.

O clique duplo em uma palavra seleciona a mesma.

O clique triplo em uma linha seleciona a linha inteira.

Se o **Shift** estiver pressionado enquanto se clica, o texto será selecionado:

- Se nenhum texto já estiver selecionado, da posição do cursor do texto até a posição do cursor do mouse.
- Se existe uma seleção, deste essa seleção até a posição do ponteiro do mouse.

NOTA

Ao selecionar o texto arrastando o mouse, o texto selecionado é copiado para a área de transferência, e pode ser colado clicando-se com o botão do meio do mouse no editor, ou em qualquer outro aplicativo para o qual você deseja colar o texto.

Para selecionar usando o teclado, pressione mantenha pressionada a tecla **Shift** usando as teclas de navegação (as setas, **Page Up**, **Page Down**, **Home** e **End**, possivelmente em combinação com o **Ctrl** para aumentar o movimento do cursor de texto).

Veja também a seção [Navegando no Texto](#), neste capítulo.

Para Copiar a seleção atual, use o item de menu **Editar** → **Copiar** ou o atalho de teclado (o padrão é **Ctrl+C**).

Para Desmarcar a seleção atual, use o item de menu **Editar** → **Desmarcar** ou o atalho de teclado (o padrão é **Ctrl+Shift+A**), ou clique com o botão esquerdo do mouse no editor.

3.3.1 Usando a Seleção de Blocos

Quando a Seleção de Bloco estiver habilitada, você pode fazer ‘seleções verticais’ no texto, o que significa selecionar colunas limitadas de múltiplas linhas. Isto é útil em trabalhos com linhas separadas com tabulações, por exemplo.

A Seleção de Bloco pode ser alternada usando-se o item de menu **Editar** → **Modo de Seleção de Bloco**. O atalho padrão no teclado é **Ctrl+Shift+B**.

3.3.2 Usando Seleção de Sobrescrita

Se a Seleção de Sobrescrita estiver habilitada, a digitação ou colagem de texto dentro da seleção causará a substituição do texto selecionado. Se não estiver habilitada, o novo texto será adicionado na posição do cursor do texto.

A Seleção de Sobrescrita está habilitada, por padrão.

Para alterar a configuração desta opção, veja a página [Cursor & Seleção](#) da [Janela de Configuração](#).

3.3.3 Usando Seleção Persistente

Quando a Seleção Persistente estiver habilitada, a digitação de caracteres ou a movimentação do cursor não fará com que a seleção desapareça. Isto significa que você pode mover o cursor para fora da seleção e digitar o texto.

A Seleção Persistente está desabilitada, por padrão.

A Seleção Persistente pode ser ativada na página [Cursor & Seleção](#) da [Janela de Configuração](#).

ATENÇÃO

Se a Seleção Persistente e a Seleção de Sobrescrita estiverem ambas habilitadas, a digitação e a colagem de texto, quando o cursor estiver dentro da seleção, farão com que ela seja substituída ou desmarcada.

3.4 Copiando e Colando Texto

Para copiar texto, selecione-o e use a opção do menu **Editar** → **Copiar**. Adicionalmente, a seleção de texto com o mouse fará com que o texto selecionado seja copiado para a área de transferência.

Para colar o texto contido atualmente na área de transferência, use o item de menu **Editar** → **Colar**.

Adicionalmente, o texto selecionado com o mouse pode ser colando clicando-se com o botão do meio do mouse na posição desejada.

DICA

Se você está usando o ambiente de trabalho KDE, você pode recuperar textos copiados recentemente de qualquer aplicativo, usando o ícone do Klipper na área de notificação.

3.5 Procurando e Substituindo Texto

3.5.1 As Barras para Pesquisar e Substituir

O KatePart possui uma barra de pesquisa incremental e uma barra avançada de pesquisa e substituição, que possibilita a inserção de um texto de substituição, juntamente com algumas opções adicionais.

As barras oferecem as seguintes opções comuns:

Procurar

Aqui é onde se digita a string de pesquisa. A interpretação da string depende de algumas das opções escritas a seguir.

Diferenciar maiúsculas de minúsculas

Se estiver ativo, a pesquisa será limitada aos itens que correspondam a nível de capitalização (maiúsculas ou minúsculas) de cada um dos caracteres no padrão de pesquisa.

A barra de pesquisa e substituição avançadas oferece algumas opções adicionais:

Texto Simples

Corresponde literalmente a qualquer ocorrência do texto de pesquisa.

Palavras Completas

Se estiver selecionado, a busca somente combinará se existir uma palavra com os mesmos finais da string encontrada, o que significa que nem caracteres alfanuméricos, nem algum outro caractere visível ou um final de linha.

Sequências de Escape

Se estiver selecionada, o botão **Adicionar** no topo de menu de contexto das caixas de texto ficará ativo permitindo que você adicione sequências de escape para o padrão de pesquisa, usando uma lista pré-definida.

Expressão regular

Se estiver assinalada, o texto a procurar é interpretado como uma expressão regular. O botão **Adicionar** no topo do menu de contexto das caixas de texto ficará ativo permitindo que você crie itens de expressão regular para o padrão de pesquisa, usando como base uma lista pré-definida.

Veja em [Expressões regulares](#) para mais informações.

Procura apenas na área selecionada

Se estiver selecionada, a pesquisa e substituição será feita apenas sobre o texto selecionado.

Procurar todos

Ao clicar neste botão, serão realçadas todas as ocorrências no documento e será mostrado o número de ocorrências encontradas numa pequena área.

3.5.2 Procurando Texto

Para procurar texto, invoque a barra de pesquisa incremental com a combinação de teclas **Ctrl+F** ou com a opção do menu **Editar** → **Procurar...**

Isto abre a barra de pesquisa incremental no fundo da janela do editor. Do lado esquerdo da barra, existe um botão com ícone para fechar a barra, seguido de um pequeno campo de texto para inserir o padrão a pesquisar.

Quando você começar a inserir os caracteres no seu padrão de pesquisa, a busca inicia imediatamente. Se existir uma ocorrência no texto, ela fica realçada e a cor de fundo do campo muda para verde-claro. Se o padrão de pesquisa não corresponder a nenhum texto, isto é indicado por uma cor de fundo vermelha-clara no campo.

Use os botões  or  para ir para a ocorrência anterior ou seguinte no documento.

As correspondências no documento são realçadas mesmo quando você fecha a barra de pesquisa. Para limpar este realce, pressione a tecla **Esc**.

Você poderá escolher aqui se a pesquisa faz a distinção entre minúsculas e maiúsculas. Se selecionar a opção , irá limitar as pesquisas para os itens que correspondam às letras (maiúsculas ou minúsculas) de cada um dos caracteres do padrão de pesquisa.

Clique no botão  à direita da barra de pesquisa incremental, para mudar para a barra mais poderosa de pesquisa e substituição.

Para repetir a última operação de pesquisa, se existir alguma e sem ter que invocar a barra de pesquisa incremental, use a opção **Editar** → **Procurar Próximo (F3)** ou **Editar** → **Procurar Anterior (Shift+F3)**.

3.5.3 Substituindo Texto

Para substituir o texto, invoque a barra avançada de pesquisa e substituição com a opção do menu **Editar** → **Substituir** ou com a combinação de teclas **Ctrl+R**.

No canto superior esquerdo da barra existe um botão com um ícone para fechar a barra, seguido de um pequeno menu suspenso para inserir o padrão de pesquisa. O menu se lembra dos padrões recentemente usados.

Você poderá controlar o modo de pesquisa selecionando as opções **Texto simples**, **Palavras completas**, **Sequências de escape** ou **Expressão regular** da caixa suspensa.

Se **Sequências de escape** ou **Expressão regular** estiver selecionado, o item de menu **Adicionar...** na base do menu de contexto de caixas de texto será habilitado e lhe permitirá adicionar sequências de escape ou expressões regulares para pesquisa ou substituição de padrões a partir de listas pré-definidas.

Use os botões  or  para ir para a ocorrência anterior ou seguinte no documento.

Insira o texto de substituição no campo de texto **Substituir** e clique no botão **Substituir**, para substituir apenas o texto realçado, ou o botão **Substituir Tudo** para substituir o texto de pesquisa em todo o documento.

Você poderá modificar o comportamento da pesquisa e da substituição ao selecionar as diferentes opções no lado direito da barra. Se selecionar a opção , irá limitar as pesquisas apenas aos itens com capitalização correspondente (maiúsculas ou minúsculas) para cada um dos caracteres no padrão de pesquisa. A opção  irá procurar e substituir apenas dentro do texto selecionado no momento. A opção **Procurar tudo** realça todas as ocorrências no documento e mostra o número de correspondências encontradas em uma pequena janela de contexto.

Clique no botão  à direita da barra avançada de pesquisa e substituição, para mudar para a barra de pesquisa incremental.

DICA

Se estiver usando uma expressão regular para encontrar um texto a ser substituído, você pode empregar referências regressivas, para reutilizar o texto capturado em subpadrões da expressão, com parênteses.

Veja em [Expressões regulares](#) para mais informações.

DICA

Você poderá **procurar, substituir e procurar de forma incremental** na [linha de comando](#).

3.6 Usando os favoritos

A característica de favoritos permite a você marcar certas linhas, para ser capaz de encontrá-las facilmente de novo.

Você pode ajustar ou remover um favorito de uma linha de duas maneiras:

- Movendo o cursor de inserção para essa linha e ativando o comando **Favoritos** → **Definir Favorito** (Ctrl+B).
- Carregando na Borda do Ícone junto a essa linha.

Os Favoritos estão disponíveis no menu **Favoritos**. Os favoritos individuais estão disponíveis como itens de menu, rotulados com o número da linha que contém o favorito, e os primeiros caracteres da linha. Para mover o cursor para o início da linha marcada, abra o menu e selecione o favorito.

Para saltar rapidamente entre favoritos ou para o favorito anterior/seguinte, use as opções **Favoritos** → **Seguinte** (Alt+Page Down) ou **Favoritos** → **Anterior** (Alt+Page Up).

3.7 Quebra de Texto Automática

Esta característica permite a você ter o texto formatado de uma maneira muito simples: o texto será quebrado, e assim nenhuma linha excederá o número máximo de caracteres por linha, a menos que exista uma string mais longa sem espaços em branco.

Para desabilitar/habilitar isto, desmarque/marque a caixa de verificação **Quebra de Linha Estática** na [página de edição](#) do [diálogo de configuração](#).

Para definir o tamanho máximo da linha (número máximo de caracteres por linha), use a opção **Quebrar Linha Em** na página de [Edição](#) da [janela de configuração](#).

Se habilitada, ele terá os seguintes efeitos:

- Ao digitar, o editor inserirá automaticamente uma quebra de linha após o último espaço em branco na posição antes que o comprimento máximo de linha seja alcançado.
- Ao carregar um documento, o editor quebrará o texto de modo similar, e assim nenhuma linha será maior que o comprimento máximo por linha, se elas contiverem quaisquer espaços em branco pertencentes a ela.

NOTA

Não existe atualmente um modo de ajustar a quebra de linha por tipo de documento, ou até mesmo habilitar ou desabilitar a característica em um nível de documento. Isto será corrigido em versões futuras do KatePart.

3.8 Usando o recuo automático

O componente de edição do KatePart suporta uma variedade de modos de recuo automático, desenhado para os diferentes formatos de texto. Você poderá escolhê-los, entre os modos disponíveis, com a opção **Ferramentas** → **Recuar**. Os módulos de recuo automático também oferecem uma função **Ferramentas** → **Formatar recuo**, que irá calcular novamente o recuo da linha atual ou a selecionada. Assim, você poderá mudar o recuo do seu documento inteiro, selecionando o texto e ativando essa opção.

Todos os modos de recuo usam a configuração de recuo no documento ativo.

DICA

Você poderá definir todos os tipos de variáveis de configuração, incluindo aqueles relacionados com o recuo com as [Variáveis do documento](#) e os [Tipos de arquivo](#).

MODOS DE RECUO AUTOMÁTICO DISPONÍVEIS

Nenhum

Se selecionar este modo, irá desativar o recuo automático.

Normal

Este sistema de recuo simplesmente mantém o recuo semelhante ao da linha anterior, com o conteúdo que não corresponda a espaços em branco. Você poderá combinar isto com a utilização das ações de recuo e remoção de recuo, para poder recuar de acordo com a sua vontade.

Estilo C

Um sistema de recuo para o C e para as linguagens semelhantes, como o C++, C#, Java™, JavaScript, entre outros. Este sistema não irá funcionar para as linguagens de 'scripting', como o Perl ou o PHP.

Haskell

Um sistema de recuo específico para a linguagem de programação Haskell.

Lilypond

Um sistema de recuo específico para a linguagem de notação musical Lilypond.

Lisp

Um sistema de recuo específico para a linguagem de programação Lisp e respectivos dialetos.

Python

Um sistema de recuo específico para a linguagem de programação Python.

Estilo XML

Um sistema de recuo específico para linguagens semelhantes ao XML.

3.9 Indicadores de modificação de linhas

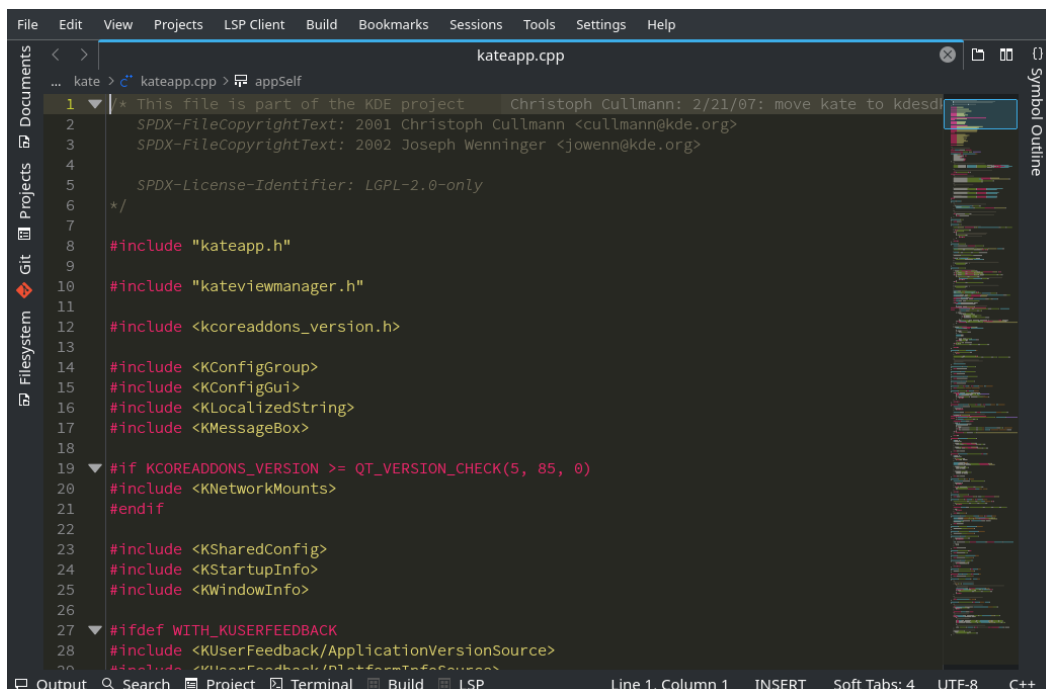
Os indicadores de modificação de linhas do KatePart permitem-lhe ver facilmente o que você alterou recentemente em um arquivo. Por padrão, as alterações gravadas são indicadas com uma barra verde à esquerda de um documento, enquanto as alterações não gravadas são indicadas com uma barra laranja.

Os indicadores de modificação de linhas em ação.

Você poderá alterar as cores usadas no [painel de configuração das Cores & Fontes](#) ou poderá desativar esta funcionalidade por completo na página [Bordas](#) do painel de configuração da [Aparência](#).

3.10 O minimapa na barra de rolagem

O minimapa na barra de rolagem exibe uma miniatura dos documentos no local da barra de rolagem. A parte visível do documento será realçada.



O minimapa na barra de rolagem mostra uma miniatura do código-fonte do Kate.

Você pode ativar ou desativar o minimapa temporariamente selecionando **Exibir → Mostrar minimapa na barra de rolagem** ou de forma permanente na seção [Aparência, da configuração do KatePart](#).

3.11 Cursores múltiplos

O suporte a múltiplos cursores foi introduzido na versão 5.93 do katepart.

3.11.1 Criando cursores múltiplos

- Para criá-los com o mouse, use **Alt** + botão esquerdo do mouse. O modificador é configurável. Consulte [Configurar modificador multicursor](#)
- Para criar usando o teclado, pressione **Ctrl+Alt+Acima** para criar o cursor acima do cursor principal e **Ctrl+Alt+Abaixo** para criar o cursor abaixo. Esses atalhos também são configuráveis
- Para criar cursores a partir de uma seleção, primeiro selecione algum texto e pressione **Shift+Alt+I**. Isso criará um cursor no final de cada linha da seleção.
- Use **Alt+J** para encontrar a próxima ocorrência da palavra sob o cursor e selecioná-la + criar um cursor. Se quiser pular a palavra atual sob o cursor, pressione **Alt+K** e a palavra selecionada será marcada como pulada. Ao pressionar **Alt+J** novamente, a palavra atual será desmarcada e o cursor passará para a próxima palavra.
- Use **Ctrl+Alt+Shift+J** para encontrar todas as ocorrências da palavra sob o cursor e selecioná-las com o cursor no final de cada seleção. Você pode usar **Alt+J** para percorrer as palavras selecionadas e usar **Alt+K** para desmarcar qualquer palavra, como mencionado no parágrafo anterior.

3.11.2 Trabalhando com cursores múltiplos

Depois de criar alguns cursores, você pode realizar a maioria das operações de edição neles como faria em um único cursor. Por exemplo, digitar uma letra a digitará para cada cursor. Da mesma forma, você pode realizar transformações de texto por exemplo, capitalização para todas as posições ou seleções.

Às vezes, você desejará remover os cursores. Para isso, você pode usar **Alt** + botão esquerdo do mouse no cursor que deseja remover. Se você quiser remover os cursores apenas em linhas vazias, existe uma ação pronta para isso para você. Para invocar a ação, abra a barra de comandos usando **Ctrl+Alt+I** e procure por **Remover cursores de linhas vazias** e pressione **Enter**. Você também pode configurar um atalho para esta ação.

Capítulo 4

Entradas de Menu

4.1 Menu Arquivo

Arquivo → Novo (Ctrl+N)

Isto inicia um novo documento numa janela de edição nova e independente.

Arquivo → Nova janela

Cria outra janela que contém o documento atual. Todas as modificações feitas no documento de uma janela serão refletidos na outra janela e vice-versa.

Arquivo → Abrir... (Ctrl+O)

Exibe o diálogo padrão do KDE **Abrir Arquivo**. Use o visualizador de arquivo para selecionar o arquivo que você deseja abrir, e clique em **Abrir** para abri-lo.

Arquivo → Abrir recente

Este é um atalho para os documentos recentemente salvos. Clicar neste item abrirá uma lista ao lado do menu com vários dos arquivos mais recentemente salvos. Clicar em um arquivo específico irá abri-lo no KatePart - se o arquivo ainda estiver na mesma localização.

Arquivo → Salvar (Ctrl+S)

Isto salva o documento atual. Se o documento já tiver sido salvo então isto sobrescreverá o arquivo anteriormente salvo sem pedir pela permissão do usuário. Se esta é a primeira vez que salva um novo documento a diálogo salvar como (descrito abaixo) será invocado.

Arquivo → Salvar como... (Ctrl+Shift+S)

Permite que um documento seja salvo com um novo nome de arquivo. Isso é feito com a caixa de diálogo de arquivo descrita acima na seção [Abrir](#) deste arquivo de ajuda.

Arquivo → Salvar como com codificação...

Salva um documento com um novo nome de arquivo e com outra codificação.

Arquivo → Salvar cópia como

Salva uma cópia do documento com um novo nome de arquivo e continua a edição do documento original.

Arquivo → Atualizar (F5)

Recarrega o arquivo ativo do disco. Este comando é útil, se outro programa ou processo tiver alterado o arquivo enquanto você o manipula no KatePart.

Arquivo → Imprimir... (Ctrl+P)

Abre uma janela de impressão que permite ao usuário especificar o quê, onde e como imprimir.

Arquivo → Exportar como HTML

Salva o documento aberto como um arquivo HTML, que será formatado de acordo com as configurações atuais de realce de sintaxe e esquema de cores.

Arquivo → Fechar (Ctrl+W)

Fecha o arquivo ativo com este comando. Se você tiver feito alterações não salvas, será questionado se deseja salvar o arquivo antes do KatePart fechá-lo.

Arquivo → Sair (Ctrl+Q)

Isto fechará a janela do editor. Se você tiver mais de uma instância do KatePart em execução, através dos itens de menu **Nova** ou **Nova Janela**, estas instâncias não serão fechadas.

4.2 Menu Editar

Editar → Desfazer (Ctrl+Z)

Desfaz o último comando de edição (digitação, cópia, colagem etc.)

NOTA

Isto pode desfazer vários comandos de edição do mesmo tipo, como digitação.

Editar → Refazer (Ctrl+Shift+Z)

Isto reverte a mudança mais recente (se houver) feita usando o Desfazer.

Editar → Recortar (Ctrl+X)

Apaga a seleção atual e a coloca na área de transferência. A área de transferência é uma funcionalidade invisível e oferece uma forma de transferir dados entre os aplicativos.

Editar → Copiar (Ctrl+C)

Copia o texto que estiver selecionado para a área de transferência, para que possa ser colado em outro local. A área de transferência é uma funcionalidade invisível e oferece uma forma de transferir dados entre os aplicativos.

Editar → Copiar como HTML

Copia a seleção como HTML, formatada de acordo com as configurações atuais de realce de sintaxe e esquema de cores.

Editar → Colar (Ctrl+V)

Insere o primeiro item da área de transferência na posição do cursor. A área de transferência é uma funcionalidade invisível e oferece uma forma de transferir dados entre os aplicativos.

NOTA

Se a Seleção de Sobrescrita estiver habilitada, o texto colado sobrescreverá a seleção, se existir.

Editar → Colar seleção (Ctrl+Shift+Ins)

Isso colará o conteúdo da [seleção do mouse](#) que foi escolhido anteriormente. Marque algum texto com o ponteiro do mouse para colá-lo no arquivo atualmente aberto usando este item de menu.

Editar → Substituir pelo conteúdo da área de transferência

Isso irá trocar o texto selecionado com o conteúdo da [área de transferência](#).

Editar → Colar histórico da área de transferência

Isso abrirá uma janela para selecionar e colar uma entrada do histórico da área de transferência.

Editar → Modos de entrada

Alterna entre os modos Normal e Estilo VI, de edição modal. O modo Estilo VI tem suporte aos comandos e sequências mais usados nos modos normal e visual do 'vim', tendo uma barra de estado opcional do modo VI. Esta barra de status mostra os comandos, à medida que vão sendo inseridos, o resultado dos comandos e o modo atual. O comportamento deste modo poderá ser configurado na aba [Modo de Entrada VI](#) da página de [Edição](#), na janela de configuração do KatePart.

Editar → Modo de sobreposição (Ins)

Ativa ou desativa os modos de Inserção/Sobrescrita. Quando o modo for o **INS**, você poderá inserir caracteres no local onde se encontra o cursor. Quando o modo for o **SOB**, os caracteres inseridos irão substituir os caracteres atuais, se o seu cursor tiver texto após ele. A barra de estado mostra a informação sobre o Modo de Sobreposição como **INS** ou **SOB**.

Editar → Modo somente leitura

Configura o documento atual no modo Somente Leitura. Isto evita que seja adicionado qualquer texto ou formatação do documento.

Editar → Localizar... (Ctrl+F)

Isto abre a barra de pesquisa incremental no fundo da janela do editor. Do lado esquerdo da barra, existe um botão com um ícone para fechar a barra, seguido de um pequeno campo de texto para introduzir o padrão a pesquisar.

Quando você começar a inserir caracteres no seu padrão de pesquisa, a busca começa imediatamente. Se existir uma ocorrência no texto, ela fica realçada e a cor de fundo do campo muda para verde-claro. Se o padrão de pesquisa não corresponder a nenhum texto, isto é indicado por uma cor de fundo vermelho-claro no campo.

Use os botões  or  para ir para a ocorrência anterior ou seguinte no documento.

As correspondências no documento são realçadas mesmo quando você fecha a barra de pesquisa. Para limpar este realce, pressione a tecla **Esc**.

Você poderá escolher aqui se a pesquisa faz a distinção entre minúsculas e maiúsculas. Se selecionar a opção , irá limitar as pesquisas para os itens que correspondam às letras (maiúsculas ou minúsculas) de cada um dos caracteres do padrão de pesquisa.

Clique no botão  à direita da barra de pesquisa incremental, para mudar para a barra mais poderosa de pesquisa e substituição.

Editar → Localizar variantes → Localizar próxima (F3)

Isto repete a última operação de busca, se houver, sem chamar a barra de pesquisa incremental, e procura no documento para frente, iniciando a partir da posição do cursor.

Editar → Localizar variantes → Localizar anterior (Shift+F3)

Isto repete a última operação de busca, se houver, sem chamar a barra de procura, e busca no documento para trás ao invés de para frente.

Editar → Localizar variantes → Localizar selecionada (Ctrl+H)

Procura a próxima ocorrência do texto selecionado.

Editar → Localizar variantes → Procurar selecionado para trás (Ctrl+Shift+H)

Procura a ocorrência anterior do texto selecionado.

Editar → Substituir... (Ctrl+R)

Este comando abre a barra avançada de pesquisa e substituição. No lado superior esquerdo da barra, existe um botão com um ícone para fechar a barra, seguido de um pequeno campo de texto para inserir o padrão a pesquisar.

Você pode controlar o modo de pesquisa selecionando as opções **Texto simples**, **Palavras completas**, **Sequências de escape** ou **Expressão regular** na lista.

Se **Sequências de escape** ou **Expressão regular** estiver selecionado, o item de menu **Adicionar...** na base do menu de contexto de caixas de texto será habilitado e lhe permitirá adicionar sequências de escape ou expressões regulares para pesquisa ou substituição de padrões a partir de listas pré-definidas.

Use os botões  or  para ir para a ocorrência anterior ou seguinte no documento.

Insira o texto de substituição no campo de texto **Substituir** e clique no botão **Substituir**, para substituir apenas o texto realçado, ou o botão **Substituir Tudo** para substituir o texto de pesquisa no documento inteiro.

Você poderá modificar o comportamento da pesquisa e da substituição ao selecionar as

diferentes opções no lado direito da barra. Se selecionar a opção , irá limitar as pesquisas apenas aos itens com capitalização correspondente (maiúsculas ou minúsculas) para

cada um dos caracteres no padrão de pesquisa. A opção  irá procurar e substituir apenas dentro do texto selecionado no momento. A opção **Procurar tudo** realça todas as ocorrências no documento e mostra o número de correspondências encontradas em uma pequena janela de contexto.

Clique no botão  à direita da barra avançada de pesquisa e substituição, para mudar para a barra de pesquisa incremental.

4.3 Menu Seleção

Seleção → Selecionar tudo (Ctrl+A)

Isto irá selecionar todo o documento. Isto pode ser muito útil para copiar um arquivo inteiro para outro aplicativo.

Seleção → Desmarcar (Ctrl+Shift+A)

Retira a seleção de texto no editor, se existir.

Seleção → Modo de Seleção em Bloco (Ctrl+Shift+B)

Ativa ou desativa o Modo de Seleção. Quando o Modo de Seleção for **BLOCO**, a barra de estado contém o texto **BLOCO** e você poderá fazer seleções verticais, por exemplo selecionar as colunas de 5 a 10 nas linhas 9 a 15.

Seleção → Comentar (Ctrl+D)

Isto adiciona um espaço ao início da linha onde o cursor de texto está localizado ou ao início de quaisquer linhas selecionadas.

Seleção → Descomentar (Ctrl+Shift+D)

Isto remove um espaço (se existir) do início da linha onde o cursor de texto está localizado ou do início de quaisquer linhas selecionadas.

Seleção → Juntar linhas (Ctrl+J)

Junta as linhas selecionadas, ou a linha atual e a seguinte, separando-as apenas com um espaço. Os espaços finais/iniciais das linhas reunidas são removidas dos extremos colados.

Seleção → Capitalização (Ctrl+Alt+U)

Coloca em maiúsculo a primeira letra do texto selecionado ou da palavra atual.

Seleção → Maiúsculas (Ctrl+U)

Coloca o texto selecionado ou a letra após o cursor em maiúsculo.

Seleção → Minúsculas (Ctrl+Shift+U)

Coloca o texto selecionado ou a letra após o cursor em minúsculo.

Seleção → Limpar recuo

Isto limpa o recuo da seleção atual ou da linha onde o cursor se encontra. Limpar o recuo, garante que todo o seu texto selecionado segue o modo de recuo que você escolher.

Seleção → Formatar recuo

Faz um novo alinhamento da linha atual ou das linhas selecionadas, usando o modo de recuo e a configuração de recuo no documento.

Seleção → Alinhar em...

Este comando alinha as linhas no bloco selecionado ou em todo o documento na coluna fornecida por uma expressão regular que será solicitada.

Se você fornecer um padrão vazio, ele se alinhará ao primeiro caractere não vazio por padrão.

Se o padrão tiver uma captura, ele será recuado na correspondência capturada.

Exemplos:

Com `'-'`, serão inseridos espaços antes do primeiro `'-'` de cada linha para alinhá-las todas na mesma coluna.

Com `'alinhar em :\\s+(.)'`, serão inseridos espaços antes do primeiro caractere não em branco que ocorre após dois pontos para alinhá-los todos na mesma coluna.

Seleção → Aplicar quebra de linha

Aplica quebra de linha estática em todo o documento. Isto significa que uma linha nova de texto irá iniciar automaticamente quando a linha atual exceder o tamanho definido na opção **Quebrar linha em** da aba de Edição do menu **Configurações → Configurar Editor...**

Seleção → Adicionar cursor acima do cursor (Ctrl+Alt+Acima)

Adiciona um novo cursor acima do cursor atual. O cursor é posicionado na linha diretamente acima e na mesma coluna que o cursor atual.

Seleção → Adicionar cursor abaixo do cursor (Ctrl+Alt+Abaixo)

Adiciona um novo cursor abaixo do cursor atual. O cursor é posicionado na linha diretamente abaixo e na mesma coluna que o cursor atual.

Seleção → Adicionar cursores no fim das linhas (Alt+Shift+I)

Adiciona um cursor a cada linha selecionada no momento.

Seleção → Procurar e selecionar a próxima ocorrência (Alt+J)

Encontra a próxima ocorrência da palavra atualmente sob o cursor, seleciona-a e adiciona um cursor.

Seleção → Procurar e selecionar todas as ocorrências (Ctrl+Alt+Shift+J)

Encontra todas as ocorrências da palavra atualmente sob o cursor, seleciona-as e adiciona um cursor para cada uma delas.

4.4 Menu Exibir

Exibir → Aumentar a fonte (Ctrl++)

Isto aumenta o tamanho da fonte usada para exibição.

Exibir → Diminuir a fonte (Ctrl+-)

Isto diminui o tamanho da fonte usada para exibição.

Exibir → Quebra de linha → Quebra de linha dinâmica (F10)

Ativa ou desativa a quebra de linha dinâmica para a janela atual. A quebra de linha dinâmica faz com que todo o texto de uma janela fique visível, sem a necessidade de se deslocar na horizontal, repartindo uma linha em várias linhas visuais, de acordo com as necessidades.

Exibir → Quebra de linha → Indicadores de quebra de linhas dinâmica

Escolhe se e quando deverão ser mostrados os indicadores de quebra de linha. Isto só fica disponível se a opção **Quebra de Linha Dinâmica** estiver assinalada.

Exibir → Quebra de linha → Mostrar marcador de quebra de linhas estática

Se esta opção estiver assinalada, será desenhada uma linha vertical na coluna de quebra de linha, como está definido na opção **Configurações → Configurar Editor...** na página de Edição. Lembre-se que o marcador de mudança de linha só é desenhado se usar uma fonte monoespaçada.

Exibir → Bordas → Mostrar borda de ícones (F6)

Isto é um item de comutação. Se o ligar, irá ativar a Borda dos Ícones do lado esquerdo do editor ativo, e vice-versa. A borda dos ícones indica as posições das linhas marcadas no editor.

Exibir → Bordas → Mostrar número das linhas (F11)

Isto é um item de comutação. Se ligá-lo, irá aparecer uma área que mostra os números de linha do documento à esquerda e vice-versa.

Exibir → Bordas → Mostrar marcas na barra de rolagem

Se esta opção estiver assinalada, a janela irá mostrar algumas marcas na barra de rolagem vertical. As marcas são equivalentes às marcas na [Borda do ícone](#).

Exibir → Bordas → Mostrar minimapa na barra de rolagem

Isto irá substituir a barra de rolagem com a visualização do documento atual. Para obter mais informações sobre o minimapa na barra de rolagem, veja [Seção 3.10](#).

Exibir → Dobradura de código

Estas opções são referentes à [dobradura de código](#):

Exibir → Bordas → Dobradura de código → Mostrar marcadores de dobra

Ativa ou desativa a visualização da área de marcadores de dobragem e desdobramento do lado esquerdo da janela.

Exibir → Dobradura de código → Dobrar nó atual

Recolhe a região que contém o cursor.

Exibir → Dobradura de código → Desdobrar nó atual

Expande a região que contém o cursor.

Exibir → Dobradura de código → Dobrar nós de nível superior (Ctrl+Shift++)

Recolhe todos os níveis superiores do documento. Clique no triângulo que aponta para a direita para expandir todos os níveis superiores.

Exibir → Dobradura de código → Desdobrar nós de nível superior (Ctrl+Shift++)

Expande todas as regiões de nível superior no documento.

Mostrar os espaços não-visíveis

Mostrar/ocultar uma área em torno dos espaços não-visíveis.

4.5 Menu Ir

Ir → Ir para linha... (Ctrl+G)

Isto abre a barra "ir para linha" no fundo da janela, que é usada para fazer com que o cursor pule para uma linha em particular (especificada pelo número) no documento. O número da linha pode ser inserido diretamente na caixa de texto ou graficamente clicando na seta para cima e para baixo do controle de contagem ao lado da caixa de texto. A pequena seta para cima aumentará o número da linha e a seta para baixo diminuirá. Feche a barra com um clique no botão com um ícone existente no lado esquerdo da barra.

Ir → Ir para linha de edição anterior (Ctrl+E)

Esta ação pula a linha de edição anterior na [configuração de múltiplos cursores](#).

Ir → Ir para a próxima linha de edição (Ctrl+Shift+E)

Esta ação pula a próxima linha de edição na [configuração de múltiplos cursores](#).

Ir → Ir para a linha modificada anterior

As linhas que foram alteradas desde a abertura do arquivo são chamadas de linhas modificadas. Esta ação move o cursor para a linha modificada anterior.

Ir → Ir para a próxima linha modificada

As linhas que foram alteradas desde a abertura do arquivo são chamadas de linhas modificadas. Esta ação move o cursor para a próxima linha modificada.

Ir → Ir para o parênteses correspondente (Ctrl+6)

Move o cursor para o parêntese de abertura ou fechamento associado.

Ir → Selecionar parênteses correspondente (Ctrl+Shift+6)

Seleciona o texto entre os parênteses de abertura e de fechamento associados.

Ir → Favoritos (Ctrl+Shift+6)

Abaixo das entradas descritas aqui, uma entrada para cada favorito no documento ativo estará disponível. O texto será as primeiras palavras da linha marcada. Escolha um item para mover o cursor para o início daquela linha. O editor rolará o necessário para tornar aquela linha visível.

Ir → Favoritos → Definir favorito (Ctrl+B)

Ajusta ou remove um favorito da linha atual do documento ativo (se já existe, é removido, caso contrário o favorito é configurado).

Ir → Favoritos → Limpar todos os favoritos

Este comando removerá todos os marcadores do documento bem como a lista de marcadores que é anexada na base deste item de menu.

Ir → Favoritos → Anterior (Alt+Page Up)

Isto irá mover o cursor para o início da linha imediatamente acima que contiver um favorito. O texto do item do menu irá incluir o número da linha e o primeiro pedaço de texto da linha. Este item só está disponível quando houver um favorito numa linha acima do cursor.

Ir → Favoritos → Próximo (Alt+Page Down)

Isto irá mover o cursor para o início da linha imediatamente abaixo que contiver um favorito. O texto do item do menu irá incluir o número da linha e o primeiro pedaço de texto da linha. Este item só está disponível quando houver um favorito numa linha acima do cursor.

4.6 Menu Ferramentas

Ferramentas → Modo

Escolhe o esquema de tipo de arquivos que preferir para o documento ativo. Isto substitui o modo de tipo de arquivos global definido na opção **Configurações → Configurar Editor...** na página de Tipos de Arquivos, somente para o seu documento atual.

Ferramentas → Realce

Escolhe o esquema de realce de sintaxe que preferir para o documento ativo. Isto substitui o modo de realce global definido na opção **Configurações → Configurar Editor...**, somente para o seu documento atual.

Ferramentas → Recuar

Escolhe o estilo de recuo que preferir para o documento ativo. Isto substitui o modo de recuo global definido na opção **Configurações → Configurar editor...**, somente para o seu documento atual.

Ferramentas → Codificação

Você poderá sobrepor a codificação predefinida na opção **Configurações → Configurar Editor...**, na página **Abrir/Salvar**, para definir uma codificação diferente para o seu documento atual. A codificação que definir aqui será válida apenas para o seu documento atual.

Ferramentas → Adicionar o Marcador da Ordem de 'Bytes' (BOM)

Se assinalar esta opção, você poderá adicionar explicitamente um marcador de ordem dos 'bytes' para os documentos codificados em Unicode. O marcador da ordem de 'bytes' é um caractere em Unicode que é usado para assinalar a ordem dos 'bytes' ('endianness') de um arquivo ou sequência de texto. Para mais informações, veja o [Marcador da Ordem de 'Bytes'](#).

Ferramentas → Fim de linha

Seleciona o modo de fim de linha que preferir para o documento ativo. Isto substitui o modo de fim de linha global definido na opção **Configurações → Configurar o Editor...**, somente para o seu documento.

Ferramentas → Scripts

Este submenu contém uma lista para todas as ações de script. A lista pode ser facilmente modificada ao se [escrever seus próprios scripts](#). Desta maneira, o KatePart pode ser estendido por ferramentas definidas pelo usuário.

Ferramentas → Scripts → Navegação

Ferramentas → Scripts → Navegação → Mover o cursor para o recuo correspondente anterior (Alt+Shift+Acima)

Move o cursor para a primeira linha acima da atual, que esteja no mesmo nível de recuo desta.

Ferramentas → Scripts → Navegação → Mover o cursor para o recuo correspondente seguinte (Alt+Shift+Abaixo)

Move o cursor para a primeira linha abaixo da atual, que esteja no mesmo nível de recuo desta.

Ferramentas → Scripts → Edição

Ferramentas → Scripts → Edição → Ordenar o texto selecionado

Ordena o texto selecionado ou o documento inteiro em ordem crescente.

Ferramentas → Scripts → Edição → Mover as linhas para baixo (Ctrl+Shift+Abaixo)

Movimenta as linhas selecionadas para baixo.

Ferramentas → Scripts → Edição → Mover as linhas para cima (Ctrl+Shift+Acima)

Movimenta as linhas selecionadas para cima.

Ferramentas → Scripts → Edição → Duplicar as linhas selecionadas para baixo (Ctrl+Alt+Abaixo)

Duplica as linhas selecionadas para baixo.

Ferramentas → Scripts → Edição → Duplicar as linhas selecionadas para cima (Ctrl+Alt+Acima)

Duplica as linhas selecionadas para cima.

Ferramentas → Scripts → Edição → Codificar em URI o texto selecionado

Codifica o texto selecionado, de forma que possa ser usado como parte de um texto de pesquisa em uma URL, substituindo a seleção pelo texto codificado.

Ferramentas → Scripts → Edição → Descodificar em URI o texto selecionado

Se parte do texto de pesquisa de uma URL estiver selecionado, isto irá decodificá-lo e substituir a seleção pelo texto original.

Ferramentas → Scripts → Emmet

Ferramentas → Scripts → Emmet → Expandir a abreviatura

Converte o texto selecionado para um par de marcas HTML ou XML abertas e fechadas. Por exemplo, se selecionar **div**, este item irá substituir isso por `<div>`
`</div>`.

Ferramentas → Scripts → Emmet → Envolver com a marca

Envolve o texto selecionado com a marca indicada na [linha de comando](#).

Ferramentas → Scripts → Emmet → Mover o cursor para a marca correspondente

Se o cursor estiver dentro de uma marca aberta HTML/XML, esta opção irá movê-lo para a marca de fechamento. Se o cursor estiver dentro da marca de fechamento, irá movê-lo para a marca de abertura.

Ferramentas → Scripts → Emmet → Selecionar o conteúdo interior da marca HTML/XML

Quando o cursor estiver dentro de um par de marcas HTML/XML, este item irá alterar a seleção para incluir o conteúdo dessas marcas de HTML/XML, sem selecionar as marcas em si.

Ferramentas → Scripts → Emmet → Selecionar o conteúdo exterior da marca HTML/XML

Quando o cursor estiver dentro de um par de marcas de HTML/XML, este item irá alterar a seleção para incluir o conteúdo dessas marcas HTML/XML, sem selecionar as marcas em si.

Ferramentas → Scripts → Emmet → Alternar comentário

Se o bloco selecionado não for um comentário, esta opção irá envolver esse bloco em comentários de HTML/XML (por exemplo, `<!-- texto selecionado -->`). Se já for um comentário, as marcas do comentário serão removidas.

Ferramentas → Scripts → Emmet → Excluir a marca sob o cursor

Se o cursor estiver dentro de uma marca de HTML/XML, este item irá excluir a marca inteira.

Ferramentas → Scripts → Emmet → Decrementar 1 ao número

Este item irá subtrair uma unidade ao texto selecionado, se for um número. Por exemplo, se tiver selecionado um **5**, irá passar para **4**.

Ferramentas → Scripts → Emmet → Decrementar 10 ao número

Este item irá subtrair dez unidades ao texto selecionado, se for um número. Por exemplo, se tiver selecionado um **15**, irá passar para **5**.

Ferramentas → Scripts → Emmet → Decrementar 0,1 ao número

Este item irá subtrair 0,1 ao texto selecionado, se for um número. Por exemplo, se tiver selecionado um **4, 5**, irá passar para **4, 4**.

Ferramentas → Scripts → Emmet → Incrementar 1 ao número

Este item irá adicionar uma unidade ao texto selecionado, se for um número. Por exemplo, se tiver selecionado um **5**, irá passar para **6**.

Ferramentas → Scripts → Emmet → Incrementar 10 ao número

Este item irá adicionar dez unidades ao texto selecionado, se for um número. Por exemplo, se tiver selecionado um **5**, irá passar para **15**.

Ferramentas → Scripts → Emmet → Incrementar 0,1 ao número

Este item irá adicionar 0,1 ao texto selecionado, se for um número. Por exemplo, se tiver selecionado um **4, 5**, irá passar para **4, 6**.

Ferramentas → Alternar para linha de comando (F7)

Mostra a linha de comando da componente do KatePart no fundo da janela. Na linha de comando, digite **help** para obter ajuda e **help list** para obter uma lista de comandos. Para mais informações sobre a linha de comando, veja a [Linha de comando do componente de edição](#).

Ferramentas → Invocar o completamento de código (Ctrl+Espaço)

Invocar manualmente o completamento de comando, normalmente usando um atalho associado a esta ação.

Ferramentas → Completamento de palavras

As opções **Reutilizar a palavra abaixo (Ctrl+9)** e **Reutilizar a palavra acima (Ctrl+8)** terminam o texto escrito no momento, procurando por palavras semelhantes antes ou depois da posição atual do cursor. O **Completamento por linha de comando** mostra uma janela com os itens correspondentes.

Ferramentas → Ortografia → Verificação ortográfica automática (Ctrl+Shift+O)

Quando a **Verificação ortográfica automática** estiver ativa, o texto escrito incorretamente ficará sublinhado no documento.

Ferramentas → Ortografia → Ortografia...

Isto inicia o programa de verificação ortográfica - um programa que é desenhado para ajudar o usuário a descobrir e a corrigir os erros ortográficos. Ao clicar nesta opção, irá iniciar o verificador ortográfico e abrir a janela de verificação ortográfica. Existem quatro campos alinhados verticalmente no centro da janela com as suas legendas correspondentes à sua esquerda. Começando a partir do topo, elas são:

Palavra desconhecida:

Aqui, o verificador ortográfico indica a palavra atualmente em análise. Isto acontece quando o verificador ortográfico encontra uma palavra que não se encontra no seu dicionário - um arquivo que contém uma lista com as palavras corretas com as quais compara cada palavra no editor.

Substituir por:

Se o verificador possuir alguma palavra semelhante no seu dicionário a primeira será listada aqui. O usuário pode aceitar a sugestão, digitar nesta caixa sua correção, ou escolher uma sugestão diferente a partir da próxima caixa.

Idioma:

Se você possuir vários dicionários instalados, poderá selecionar aqui o dicionário/idioma a ser usado.

Do lado direito da janela, existem 6 botões que permitem ao usuário controlar o processo de verificação ortográfica. São estes:

Adicionar ao Dicionário

Pressionar este botão adiciona a **Palavra Desconhecida** ao dicionário do verificador. Isto significa que no futuro o verificador sempre considerará esta palavra como ortograficamente correta.

Sugerir

O verificador poderá enumerar aqui uma quantidade de substituições possíveis para a palavra em consideração. Clicar em qualquer uma das sugestões, fará com que ela seja introduzida no campo **Substituir por** acima.

Substituir

Este botão faz o verificador substituir a palavra em consideração no documento com a palavra na caixa **Substituir por**.

Substituir Todas

Este botão faz com que o verificador ortográfico substitua não somente a **Palavra desconhecida**: atual, mas automaticamente faça a mesma substituição para qualquer outra ocorrência desta **Palavra desconhecida** no documento.

Ignorar

Ativar este botão fará com que o verificador ortográfico avance sem realizar nenhuma alteração.

Ignorar Todas

Este botão faz com que o verificador não faça nada com a **Palavra desconhecida**: atual e em nenhuma das outras ocorrências da mesma palavra.

NOTA

Isto só se aplica a esta rodada de verificação ortográfica. Se o verificador for executado de novo, mais tarde, ele irá parar na mesma palavra.

Existem mais três botões no fundo da janela de verificação ortográfica. Estes são:

Ajuda

Isto invoca o sistema de ajuda do KDE com a página de ajuda para esta janela.

Terminado

Este botão cancela o processo de verificação ortográfica e volta para o documento.

Cancelar

Este botão cancela o processo de verificação ortográfica; todas as modificações são anuladas e você irá voltar ao seu documento.

Ferramentas → Ortografia → Ortografia (do cursor)...

Isto inicia o programa de verificação ortográfica, mas inicia a partir do cursor, em vez de começar no início do documento.

Ferramentas → Ortografia → Verificação ortográfica da seleção...

Faz uma verificação ortográfica da seleção atual.

Ferramentas → Ortografia → Alterar dicionário

Mostra uma lista com todos os dicionários disponíveis para verificação ortográfica no fundo da janela de edição. Isto permite a mudança rápida do dicionário de verificação ortográfica, por exemplo para a verificação automática do texto em vários idiomas.

4.7 Menus Configurações e Ajuda

Configurações → Editar tema de cores

Este menu apresenta os esquemas de cor disponíveis. Você poderá alterar aqui o esquema para a janela atual; para alterar o esquema padrão, você deve usar a página [Fontes & Cores](#) da janela de configuração.

O KatePart possui os itens dos menus **Configurações** e **Ajuda** comuns do KDE. Para mais informações, consulte as seções sobre o [Menu Configurações](#) e o [Menu Ajuda](#) dos Fundamentos do KDE.

Capítulo 5

Ferramentas Avançadas de Edição

Anders Lund
Dominik Haumann
Tradução: Lisiane Sztoltz

5.1 Comentar/Descomentar

Os comandos **Comentar** e **Descomentar**, disponíveis no menu **Ferramentas** permitem-lhe adicionar ou remover marcações de comentários à seleção ou à linha atual, se não existir texto selecionado, se os comentários são suportados pelo formato do texto que você está editando.

As regras de como o comentário é feito são feitas nas definições de sintaxe; assim, se o realce de sintaxe não é usado, as ações de comentar/descomentar não são possíveis.

Alguns formatos definem marcas de comentários para linhas únicas, outros suportam marcas de comentários para várias linhas e alguns suportam ambas. Se não estiverem disponíveis os comentários para várias linhas, o ato de comentar uma seleção que não inclui por completo a sua última linha não é possível.

Se um marcador de linha única estiver disponível, é preferível comentar linhas simples onde for aplicável, pois isto ajuda a evitar problemas com comentários aninhados.

Ao remover marcadores de comentários, textos sem comentários devem ser selecionados; ao remover marcadores de comentários de múltiplas linhas de uma seleção, qualquer espaço em branco fora do marcador é ignorado.

Para colocar marcações de comentários, use a opção do menu **Ferramentas** → **Comentar** ou o atalho de teclado relacionado, que por padrão é **Ctrl+D**.

Para remover marcações de comentários, use a opção do menu **Ferramentas** → **Descomentar** ou o atalho de teclado relacionado, que por padrão é o **Ctrl+Shift+D**.

5.2 A Linha de Comando do Componente de Edição

O componente de edição do KatePart possui uma linha de comando interna, que lhe permite efetuar várias ações a partir de uma GUI mínima. A linha de comando é um campo de texto no fundo da área de edição onde, para mostrá-la, você poderá selecionar a opção **Ver** → **Alternar para Linha de Comando** ou usar o atalho (que é, por padrão, o **F7**). O editor oferece um conjunto de comandos, tal como está documentado abaixo, sendo oferecidos mais comandos através de plugins.

Para executar um comando, pressione a tecla Return. A linha de comando irá indicar se foi bem sucedida e, provavelmente, irá mostrar uma mensagem. Se você inseriu na linha de comando pressionando F7, ele ocultar-se-á automaticamente, ao fim de alguns segundos. Para limpar a mensagem e inserir um comando novo, pressione F7 de novo.

A linha de comando possui um sistema de ajuda embutido, podendo inserir o comando **help** para iniciá-lo. Para ver uma lista com todos os comandos disponíveis, invoque o comando **help list**; para ver a ajuda sobre um determinado comando, faça **help comando**.

A linha de comando possui um histórico embutido, como tal, o usuário poderá reutilizar os comandos já digitados. Para navegar no histórico, use as teclas **Acima** e **Abaixo**. Ao mostrar os comandos do histórico, a parte do argumento do comando será selecionada, o que lhe permite sobrepor facilmente os argumentos.

5.2.1 Comandos Padrão da Linha de Comando

TIPOS DE ARGUMENTO

BOOLEANO

Isto é usado com os comandos que ativam ou desativam certas coisas. Os valores válidos são o **on**, **off**, **true**, **false**, **1** ou **0**.

INTEIRO

Um número inteiro.

TEXTO

Uma string separada por aspas simples (') ou aspas (") quando conter espaços.

5.2.1.1 Comandos para configurar o Editor

Estes comandos são oferecidos pelo componente do editor e permitem-lhe configurar apenas o documento e a janela ativos. Isto poderá ser útil se quiser usar uma configuração diferente da pré-definida como, por exemplo, no recuo.

set-tab-width INTEIRO largura

Configura a largura da tabulação como sendo **largura**.

set-indent-width INTEIRO largura

Configura a largura do recuo como **largura**. Só é usado se o usuário recuar com espaços.

set-word-wrap-column INTEIRO largura

Configura a largura da linha para uma quebra como tendo **largura** colunas. Isto é usado se você tiver o seu texto sendo quebrado automaticamente.

set-icon-border BOOLEANO ativar

Configura a visibilidade da borda dos ícones.

set-folding-markers BOOLEANO ativar

Configura a visibilidade da área de marcadores de dobra do código.

set-line-numbers BOOLEANO ativar

Configura a visibilidade da área de números de linha.

set-replace-tabs BOOLEANO ativar

Se estiver ativo, as tabulações são substituídas por espaços, à medida que você vai digitando.

set-remove-trailing-space **BOOLEANO** **ativar**

Se estiver ativo, os espaços finais serão removidos quando o cursor abandonar uma determinada linha.

set-show-tabs **BOOLEANO** **ativar**

Se estiver ativo, os caracteres de tabulação e espaços estarão visíveis através de um pequeno ponto.

set-show-indent **BOOLEANO** **ativar**

Se estiver ativo, o recuo será visualizado por uma linha tracejada vertical.

set-indent-spaces **BOOLEANO** **ativar**

Se estiver ativo, o editor irá recuar com `largura-recuo` espaços por cada nível de recuo, em vez de usar o caractere TAB.

set-mixed-indent **BOOLEANO** **ativar**

Se estiver ativo, o KatePart irá usar uma mistura de TABs e espaços para o recuo. Cada nível de recuo irá ter um tamanho de `largura-recuo`, enquanto os restantes níveis de recuo serão otimizados para usar tantos caracteres TAB quantos possíveis.

Quando for executado, este comando irá ativar o recuo com espaços e, se a largura de recuo não estiver definida, corresponderá a metade da `largura-tabulação` do documento, no momento da execução.

set-word-wrap **BOOLEANO** **ativar**

Ativa a quebra de linha dinâmica, de acordo com o parâmetro **ativar**.

set-replace-tabs-save **BOOLEANO** **ativar**

Quando estiver ativo, as tabulações serão substituídas por espaços em branco, sempre que o documento for salvo.

set-remove-trailing-space-save **BOOLEANO** **ativar**

Quando estiver ativo, os espaços finais serão removidos de cada linha, sempre que o documento for salvo.

set-indent-mode **TEXTO** **nome**

Configura o modo de recuo automático como **nome**. Se o **nome** não for conhecido, o modo é configurado como 'none' (nenhum). Os modos válidos são 'none', 'normal', 'cstyle', 'askell', 'lilypond', 'lisp', 'python', 'ruby' e 'xml'.

set-auto-indent **BOOLEANO** **script**

Ativa ou desativa o recuo automático.

set-highlight **TEXTO** **realce**

Configura o sistema de realce de sintaxe para o documento. O argumento deverá ser um nome de realce válido, como pode ser visto no menu **Ferramentas** → **Realce**. Este comando oferece uma lista de completção automática para o seu argumento.

reload-scripts

Recarrega todos os [scripts do JavaScript](#) usados pelo Kate, incluindo scripts de indentação e de linha de comando.

set-mode **STRING** **modo**

Seleciona o esquema de tipo de arquivo para o documento atual.

nn[oremap] **TEXTO** **original** **TEXTO** **mapeado**

Mapeia a sequência **original** para a **mapeada**.

5.2.1.2 Comandos de edição

Estes comandos modificam o documento atual.

indent

Recua as linhas selecionadas ou a linha atual.

unindent

Remove um nível de recuo das linhas selecionadas ou da linha atual.

cleanindent

Limpa o recuo das linhas selecionadas ou da linha atual, de acordo com a configuração de recuo no documento.

comment

Insere marcações de comentários para fazer da seleção, linhas selecionadas ou da linha atual um comentário, de acordo com o formato de texto definido pela definição do realce de sintaxe para o documento.

uncomment

Remove os marcadores de comentários da seleção, linhas selecionadas ou da linha atual, de acordo com o formato de texto definido pela definição do realce de sintaxe do documento.

kill-line

Remove a linha atual.

replace TEXTO padrão TEXTO substituição

Substitui o texto correspondente ao **padrão** por **substituição**. Se você quiser incluir espaços em branco no **padrão**, deverá colocar tanto o **padrão** como a **substituição** entre aspas simples ou duplas. Se os argumentos não estiverem entre estes caracteres, a primeira palavra é usada como **padrão** e o resto como a **substituição**. Se a **substituição** estiver em branco, cada ocorrência do **padrão** será removida.

Você poderá configurar opções para modificar a procura, adicionando dois pontos (:), seguidos de uma ou mais letras, onde cada uma representa uma configuração, o que corresponderá ao formato **replace:opções padrão substituição**. As opções disponíveis são:

b

Pesquisar para trás.

c

Procura a partir da posição do cursor.

e

Procura apenas na área selecionada.

r

Faz uma pesquisa por uma determinada expressão regular. Se estiver ativo, você poderá usar o **\N**, onde o N é um número que representa as capturas no texto de substituição.

s

Faz uma pesquisa com distinção entre maiúsculas e minúsculas.

p

Pede permissão para substituir a próxima ocorrência.

w

Faz uma correspondência apenas com palavras inteiras.

date TEXTO formato

Insere um texto de data/hora, tal como está definido pelo **formato** indicado, ou com o formato 'aaaa-MM-dd hh:mm:ss', se não for indicada qualquer formatação. As traduções a seguir são feitas quando for interpretado o **formato**:

d	O dia, como um número sem um zero inicial (1-31).
dd	O dia, como um número com um zero inicial (01-31).
ddd	O nome do dia abreviado, no formato regional (por exemplo 'Seg'..'Dom').
dddd	O nome do dia completo, no formato regional (por exemplo 'Segunda'..'Domingo').
M	O mês, como um número sem um zero inicial (1-12).
MM	O mês, como um número com um zero inicial (01-12).
MMMM	O nome completo do mês localizado (por exemplo 'Janeiro'..'Dezembro').
MMM	O nome do mês abreviado, no formato regional (por exemplo 'Jan'..'Dez').
aa	O ano, como um número de dois algarismos (00-99).
aaaa	O ano, como um número de quatro algarismos (1752-8000).
h	As horas sem um zero inicial (0..23 ou 1..12 no caso do formato AM/PM).
hh	As horas com um zero inicial (00..23 ou 01..12 no caso do formato AM/PM).
m	Os minutos sem um zero inicial (0..59).
mm	Os minutos com um zero inicial (00..59).
s	Os segundos sem um zero inicial (0..59).
ss	Os minutos com um zero inicial (00..59).
z	Os milissegundos sem um zero inicial (0..999).
zzz	Os milissegundos com zeros iniciais (000..999).
AP	Usar o formato de horas AM/PM. O AP será substituído por "AM" ou por "PM".
ap	Usar o formato de horas am/pm. O ap será substituído por "am" ou por "pm".

char TEXTO identificador

Este comando permite que você insira caracteres literais através de suas identificações numéricas, em decimal, octal ou hexadecimal. Para usá-lo, abra o diálogo de Edição de Comando e digite **char**: [número] na caixa de entrada e clique em **OK**.

Example 5.1 exemplos do comando char

Entrada: **char**: 234

Resultado: ê

Entrada: **char**: 0x1234

Resultado: jué

s///[ig] %s///[ig]

Este comando faz um sed como operação de procura/substituição na linha atual, ou em um arquivo inteiro (%s///).

Em resumo, é procurado o texto que combine com o *padrão de busca*, a expressão regular entre a primeira e a segunda barra, e quando é encontrado, a parte que combina com o texto é substituída pela expressão entre a segunda e a última barra. Os parênteses no padrão de busca criam *referências anteriores*, isto é, o comando lembra qual parte encontrada do texto combinou no parênteses; estas strings podem ser reutilizadas no padrão de substituição, referidas como \1 para o primeiro conjunto de parênteses, \2 para o segundo e assim por diante.

Para procurar por um caractere literal (ou), você precisa usar o caractere de *escape*, que é o caractere de barra invertida: \ (\)

Se você puser um **i** no fim da expressão, a correspondência não fará distinção entre maiúsculas ou minúsculas. Se, por outro lado, puser um **g** no fim, todas as ocorrências do padrão serão substituídas, caso contrário, só será substituída a primeira ocorrência.

Example 5.2 Substituindo texto na linha atual

Seu compilador parou, dizendo a você que a classe `minhaClasse`, mencionada na linha 3902 do seu arquivo fonte, não está definida.

"Droga!" você pensa, é claro que é `MinhaClasse`. Você vai para a linha 3902, e em vez de tentar encontrar a palavra no texto, você lança o Diálogo de Edição de Comando, digita **s/minhaclasse/MinhaClasse/i**, clica no botão **OK**, salva o arquivo e compila – com sucesso, sem erros.

Example 5.3 Substituindo texto no arquivo inteiro

Imagine que você tem um arquivo, no qual está mencionada a string 'Senhorita Jensen' várias vezes, quando aparece alguém e lhe diz que ela casou com o 'Senhor Jones'. Você quer, é claro, substituir todas as ocorrências de 'Senhorita Jensen' por 'Senhora Jones'.

Entre na linha de comando e digite o comando **%s/Senhorita Jensen/Senhora Jones/**, pressionando RETURN para terminar.

Example 5.4 Um Exemplo Mais Avançado

Este exemplo tira partido das *referências anteriores* assim como das *classes de palavras* (se você não souber o que isso é, baseie-se na documentação referida abaixo).

Suponha que você possui a seguinte linha:

```
void MinhaClasse::DoStringOps( String      &foo, String &bar, String *p, ←
    int  &a, int &b )
```

Agora, você pode perceber que este não é um código bom, e decidir que deseja usar a palavra-chave `const` para todos os argumentos ‘address of’, aqueles caracterizados pelo operador `&` em frente ao nome do argumento. Você poderia também simplificar o espaço em branco, pois assim existiria somente um caractere de espaço em branco entre cada palavra.

Abra o Diálogo de Edição de Comando, e digite `s/\s+(\w+)\s+(\&)/ const \1 \2/g` e clique no botão **OK**. O caractere `g` no final da expressão faz com que a expressão regular recompila cada combinação, para recuperar as *referências de volta*.

Saída `void MinhaClasse::DoStringOps( const String &foo, const String &bar, String *p, const int &a, const int &b )`

Missão completa! Agora, o que aconteceu? Bem, procuramos por espaços em branco (`\s+`) seguidos por um ou mais caracteres alfabéticos (`\w+`) seguidos por mais espaços em branco (`\s+`) seguidos por um `E` comercial, e no processo é salvo o pedaço alfabético e o `E` comercial para reutilização na operação de substituição. Quando substituímos a parte relacionada da nossa linha, com um espaço em branco, seguido por ‘`const`’, seguido por um espaço em branco, seguido por nosso pedaço alfabético (`\1`) seguido por um espaço em branco, em por fim, seguido por nosso `E` comercial salvo (`\2`)

Em alguns casos, a parte alfabética foi ‘`String`’, em alguns ‘`int`’, assim, usar a classe de caractere `\w` e o quantificador `+` provê uma vantagem considerável.

sort

Ordena o texto selecionado ou todo o documento.

natsort

Ordena as linhas selecionadas ou todo o documento naturalmente.

Example 5.5 sort x natsort

sort(a10, a1, a2) resulta em a1, a10, a2

natsort(a10, a1, a2) resulta em a1, a2, a10

moveLinesDown

Move as linhas selecionadas para baixo.

moveLinesUp

Move as linhas selecionadas para cima.

uniq

Remove linhas duplicadas do texto selecionado ou do documento inteiro.

rtrim

Remove o espaço à direita do texto selecionado ou do documento inteiro.

ltrim

Remove o espaço à esquerda do texto selecionado ou do documento inteiro.

join [TEXTO separador]

Mescla as linhas selecionadas ou o documento inteiro. Opcionalmente, recebe um parâmetro definindo um separador, por exemplo: `join ', '`

rmblank

Remove todos os espaços em branco do texto selecionado ou do documento inteiro.

alignon

Este comando alinha as linhas no bloco selecionado ou em todo o documento na coluna especificada por uma expressão regular fornecida como argumento.

Se você fornecer um padrão vazio, ele se alinhará ao primeiro caractere não vazio por padrão.

Se o padrão tiver uma captura, ele será recuado na correspondência capturada.

Exemplos:

alignon - irá inserir espaços antes do primeiro '-' de cada linha para alinhá-las na mesma coluna.

alignon :\\s+(.) irá inserir espaços antes do primeiro caractere não em branco que ocorrer após dois pontos para alinhá-los todos na mesma coluna.

unwrap

Remove as quebras de linha do texto selecionado ou do documento inteiro.

each TEXTO script

Fornecida uma função do JavaScript como argumento, chama pela lista de linhas selecionadas e substitui-as com o valor de retorno desta chamada.

Example 5.6 Mescla as linhas selecionadas

```
each 'function(lines){return lines.join(", ")}'
```

Ou, de maneira resumida:

```
each 'lines.join(", ").'
```

filter TEXTO script

Fornecida uma função do JavaScript como argumento, chama pela lista de linhas selecionadas e remove-as onde a chamada retorna falso.

Example 5.7 Remover linhas em branco

```
filter 'function(l){return l.length > 0;}'
```

Ou, de maneira resumida:

```
filter 'line.length > 0'
```

map TEXTO script

Fornecida uma função do JavaScript como argumento, chama pela lista de linhas selecionadas e substitui a linha com o valor de retorno da chamada.

Example 5.8 Remover linhas em branco

```
map 'function(line){return line.replace(/^s+/, "");}'
```

Ou, de maneira resumida:

```
map 'line.replace(/^s+/, "")'
```

duplicateLinesUp

Duplica as linhas selecionadas acima da seleção atual.

duplicateLinesDown

Duplica as linhas selecionadas abaixo da seleção atual.

5.2.1.3 Comandos de navegação

goto INT linha

Este comando vai para a linha indicada.

grep TEXTO padrão

Pesquisa o documento pela expressão regular **padrão**. Para mais informações, veja as apêndice [A](#).

find TEXTO padrão

Este comando vai para a primeira ocorrência do **padrão**, de acordo com a configuração. As ocorrências a seguir poderão ser encontradas com a opção **Editar** → **Procurar Próximo** (o atalho por padrão é o **F3**).

O comando de procura poderá ser configurado adicionando dois-pontos, seguidos de uma ou mais opções, sendo o formato **find:opções padrão**. São suportadas as seguintes opções:

- b** Pesquisar para trás.
- c** Procura a partir da posição do cursor.
- e** Procura apenas na área selecionada.
- r** Faz uma pesquisa por uma determinada expressão regular. Se estiver ativo, você poderá usar o **\N**, onde o N é um número que representa as capturas no texto de substituição.
- s** Faz uma pesquisa com distinção entre maiúsculas e minúsculas.
- w** Faz uma correspondência apenas com palavras inteiras.

ifind TEXTO padrão

Este comando oferece uma procura ‘à-medida-que-digita’. Você poderá configurar o comportamento da procura adicionando dois-pontos (:), seguidos de uma ou mais opções, como se segue: **ifind:opções padrão**. As opções permitidas são:

- b** Pesquisar para trás.
- r** Faz uma procura por uma expressão regular.
- s** Faz uma pesquisa com distinção entre maiúsculas e minúsculas.
- c** Procura a partir da posição do cursor.

5.2.1.4 Comandos das funções básicas do editor (dependem do aplicativo em que o componente de edição for usado)

w
Salvar o documento atual.

wa
Salvar todos os documentos abertos.

q	Fechar o documento atual.
qa	Fechar todos os documentos abertos.
wq	Salvar e fechar o documento atual.
wqa	Salvar e fechar todos os documentos abertos.
x	Salvar e fechar o documento atual somente se ele foi modificado.
x	Salvar e fechar todos os documentos abertos somente se eles foram modificados.
bp	Ir para o documento anterior na lista de documentos.
bn	Ir para o próximo documento na lista de documentos.
new	Abrir um novo documento em uma divisão de visão horizontal.
vnew	Abrir um novo documento em uma divisão de visão vertical.
e	Recarregar o documento atual se ele tiver sido modificado em disco.
enew	Editar um novo documento.
print	Abrir a janela de impressão para imprimir o documento atual.

5.3 Usando o Desdobramento de Código

O desdobramento ou dobragem de código permite-lhe ocultar partes de um documento no editor, tornando mais simples analisar globalmente arquivos grandes. No KatePart, as regiões desdobráveis são calculadas com base nas regras das definições de realce de sintaxe; como tal só está disponível em alguns formatos - tipicamente em código-fonte de programas, código em XML e semelhantes. A maioria das definições de realce que suportam o desdobramento de código também lhe permitem definir manualmente regiões desdobráveis, tipicamente com as palavras-chave **BEGIN** e **END**.

Para usar a funcionalidade de desdobramento de código, ative os marcadores com a opção **Exibir** → **Mostrar as Marcações de Dobragem**, se não estiverem já visíveis. A área de Marcações de Dobragem, no lado esquerdo da tela, mostra uma visão gráfica sobre as regiões que poderão ser recolhidas ou expandidas, com símbolos triangulares para indicar a operação possível numa determinada região: um triângulo para baixo significa que a região é expandida; clicar sobre ele irá recolher a região, clicar em um triângulo para direita irá expandi-la.

Estão disponíveis três comandos para manipular o estado das regiões dobradas; para tal, veja a [documentação do menu](#).

As linhas recolhidas serão lembradas quando um arquivo é fechado, assim quando você reabri-lo, os nós que estiverem recolhidos permanecerão dobrados. Isto se aplica a operações de recarregamento também.

Se você não quiser usar a funcionalidade de desdobramento do código, poderá desativar a opção **Mostrar as marcações de dobragem (se disponíveis)** na página [Aparência](#) do editor de configuração.

Capítulo 6

Ampliando o KatePart

T.C. Hollingsworth

Tradução: Lisiane Sztoltz

Tradução: André Marcelo Alvarenga

Tradução: Marcus Gama

6.1 Introdução

Como em qualquer componente de editor de texto avançado, o KatePart oferece uma grande variedade de formas para ampliar suas funcionalidades. Você pode [criar scripts simples para adicionar funcionalidades com o JavaScript](#) Finalmente, assim que tiver ampliado as funcionalidades do KatePart, sintá-se a vontade para [se juntar a nós](#) e compartilhar as suas melhorias com o mundo!

6.2 Trabalhando com Realce de Sintaxe

6.2.1 Introdução

O Realce de Sintaxe é o que faz com que o editor exiba automaticamente o texto em diferentes cores/estilos, dependendo da função da string em questão para o propósito do arquivo. No código-fonte do programa, por exemplo, declarações de controle pode ser renderizadas em negrito, enquanto tipos de dados e comentários ficam com cores diferentes do restante do texto. Isto aumenta consideravelmente a legibilidade do texto e assim, o autor pode ser mais eficiente e produtivo.

```
Theme Repository::defaultTheme(Repository::DefaultTheme t)
{
    if (t == DarkTheme)
        return theme(QLatin1String("Breeze Dark"));
    return theme(QLatin1String("Default"));
}
```

Uma função C++, representada com realce de sintaxe.

```
Theme Repository::defaultTheme(Repository::DefaultTheme t)
{
    if (t == DarkTheme)
        return theme(QLatin1String("Breeze Dark"));
    return theme(QLatin1String("Default"));
}
```

A mesma função C++, sem realce de sintaxe.

Dos dois exemplos, qual é o mais fácil de ler?

O KatePart vem com um sistema flexível, configurável e capaz de fazer realce de sintaxe; a distribuição padrão oferece definições para um vasto conjunto de linguagens de programação, de manipulação e de 'scripting', bem como para outros formatos de texto. Além disso, você pode criar as suas próprias definições em arquivos XML simples.

O KatePart detectará automaticamente regras de realce de sintaxe quando você abrir um arquivo, baseado no tipo MIME do arquivo, determinado pela extensão ou, se não existir, pelo conteúdo. Se você não conseguir, ajuste manualmente a sintaxe para o uso no menu **Ferramentas** → **Realçar**.

Os estilos e cores usados por cada definição de realce de sintaxe podem ser configurados usando a página de [Estilos de Realce de Texto](#) da [Janela de Configuração](#); por outro lado, os tipos MIME e as extensões de arquivos para os quais deve ser usada, podem ser configurados usando a página de [Modos & Tipos de Arquivo](#).

NOTA

O realce de sintaxe existe para aumentar a legibilidade do texto correto, mas não se pode confiar nisto para validar seu texto. Marcar o texto para sintaxe é difícil, dependendo do formato que você está usando e, em alguns casos, os autores das regras de sintaxe ficarão orgulhosos se 98% do texto é renderizado corretamente, embora muito frequentemente você precise de um estilo raro para ver os 2% incorretos.

6.2.2 O Sistema de Realce de Sintaxe do KatePart

Esta seção irá discutir o realce de sintaxe do KatePart em detalhes. É para você, caso deseje saber mais sobre esta funcionalidade ou se quiser criar ou alterar as definições de sintaxe.

6.2.2.1 Como funciona

Sempre que você abrir um arquivo, uma das primeiras coisas que o editor KatePart faz é detectar qual definição de sintaxe deve ser usada para o arquivo. Ao ler o texto do arquivo, e enquanto você digita no arquivo, o sistema de realce de sintaxe analisará o texto usando as regras definidas pela definição de sintaxe, e marcará no texto onde contexto e estilos diferentes iniciarem e finalizarem.

Quando você escrever no documento, o novo texto será analisado e marcado na hora, pois se você remover um caractere que está marcado como início ou fim de um contexto, o estilo em volta do texto modifica de acordo com ele.

As definições de sintaxe usadas pelo sistema de realce de sintaxe do KatePart são arquivos XML que contém

- Regras para a detecção do texto inteiro, organizado em blocos de contexto
- Listas de palavras-chave
- Definições de Item de Estilo

Ao analisar o texto, as regras de detecção serão avaliadas na ordem em que foram definidas, e se o início da string atual coincidir com uma regra, o contexto relacionado será usado. O ponto de partida no texto é movido para o ponto final no local onde aquela regra coincide, e um novo loop de regras inicia, começando no contexto configurado pela regra relacionada.

6.2.2.2 Regras

As regras de detecção são o “coração” do sistema de detecção de sintaxe. Uma regra é uma string, um caractere ou uma **expressão regular** com a qual se faz a correspondência do texto a analisar. Contém informações sobre o estilo a ser usado na parte correspondente do texto. Pode mudar do contexto atual do sistema para outro contexto explícito ou para o contexto anterior usado pelo texto.

As regras são organizadas em grupos de contexto, sendo que este é usado pelos conceitos principais do texto dentro de um formato, como por exemplo strings dentro de aspas ou blocos de comentário do código-fonte de um programa. Isto garante que o sistema de realce não precisa ficar procurando todas as regras quando não for necessário, e também que sequências de algum caractere no texto podem ser tratadas de modo diferente, dependendo do contexto atual.

As regras são organizadas em grupos de contexto, sendo que este é usado pelos conceitos principais do texto dentro de um formato, como por exemplo strings dentro de aspas ou blocos de comentário do código-fonte de um programa. Isto garante que o sistema de realce não precisa ficar procurando todas as regras quando não for necessário, e também que sequências de algum caractere no texto podem ser tratadas de modo diferente, dependendo do contexto atual.

6.2.2.3 Estilos de Contexto e Palavras-Chave

Em algumas linguagens de programação, os números inteiros são tratados diferentemente dos números de ponto flutuante pelo compilador (o programa que converte o código-fonte para um binário executável), e podem existir caracteres que possuem significado especial dentro de uma string. Em tais casos, faz sentido renderizá-los de modo diferente dos outros, assim, são mais fáceis de identificar durante a leitura do texto. Mesmo que eles não sejam representados em contextos especiais, pode ser vistos no sistema de realce de sintaxe, e assim, podem ser marcados com uma renderização diferente.

Uma definição de sintaxe pode conter tantos estilos quanto forem necessários para cobrir os conceitos do formato no qual serão usados.

Em muitos formatos, existem listas de palavras, que representam um conceito específico; por exemplo, em linguagens de programação, as declaração de controle são um conceito, nomes de tipos de dados outro conceito, e funções pré-integradas na linguagem um terceiro conceito. O Sistema de Realce de Sintaxe do KatePart pode usar estas listas para detectar e marcar palavras no texto para enfatizar os conceitos dos formatos.

6.2.2.4 Estilos Padrão

Se você abrir um arquivo de código em C++, um arquivo de Java™ e um documento em HTML no KatePart, irá ver que, ainda que os formatos sejam diferentes e, por isso, sejam selecionadas palavras diferentes para um tratamento especial, as cores usadas são as mesmas. Isto deve-se ao fato do KatePart ter uma lista pré-definida de Estilos Padrão, os quais são usados pelas definições de sintaxe individuais.

Isto faz com que fique mais fácil reconhecer conceitos similares em diferentes formatos. Por exemplo, os comentários estão presentes na maioria das linguagens de programação, script e marcação, e quando são renderizados utilizando-se o mesmo estilo em todas as linguagens, você não precisa parar e pensar para identificá-los dentro do texto.

DICA

Todos os estilos de uma definição de sintaxe usam um dos estilos padrão. Algumas definições de sintaxe usam mais estilos além dos pré-definidos, por isso se você usar um formato frequentemente, pode ser útil abrir a janela de configuração para ver se alguns conceitos usam o mesmo estilo. Por exemplo, só existe um estilo padrão para as cadeias de caracteres, mas como a linguagem de programação Perl lida com dois tipos de cadeias de caracteres, você pode melhorar o realce se configurar esses dois tipos de uma forma ligeiramente diferente. Todos os [estilos padrão disponíveis](#) serão explicados mais tarde.

6.2.3 O Formato XML de Definição de Realce

6.2.3.1 Introdução

O pacote KatePart utiliza o framework de realce de sintaxe do pacote KDE Frameworks. Os arquivos XML de realce padrão fornecidos com o pacote KatePart são compilados na biblioteca de realce de sintaxe por padrão.

Esta seção é uma introdução ao formato XML de Definição de Realce. Baseado em um pequeno exemplo, ele irá descrever as componentes principais, bem como o seu significado e utilização. A próxima seção colocará em detalhes as regras de detecção.

A definição formal, também conhecida como XSD, você encontra no [Repositório de realce de sintaxe](#) no arquivo `language.xsd`

Os arquivos de definição de realce personalizados `.xml` estão localizados em `org.kde.syntax-highlighting/syntax/` na sua pasta de usuário, encontrada com **`qtpaths --paths GenericDataLocation`** que geralmente são `$HOME /.local/share/` e `/usr/share/`.

Em pacotes Flatpak e Snap, o diretório acima não funcionará, pois o local dos dados é diferente para cada aplicativo. Em um aplicativo Flatpak, o local dos arquivos XML personalizados geralmente é `$HOME /.var/app/ nome-pacote-flatpak /data/org.kde.syntax-highlighting/syntax/` e em um aplicativo Snap, esse local é `$HOME /snap/ nome-pacote-snap /current/.local/share/org.kde.syntax-highlighting/syntax/`.

No Windows, esses arquivos estão localizados em `%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\syntax`. `%USERPROFILE%` geralmente se expande para `C:\Users\user`.

Em resumo, para a maioria das configurações, o diretório de arquivos XML personalizados é o seguinte:

Para o usuário local	<code>\$HOME /.local/share/org.kde.syntax-highlighting/syntax/</code>
Para todos os usuários	<code>/usr/share/org.kde.syntax-highlighting/syntax/</code>
Para pacotes Flatpak	<code>\$HOME /.var/app/ nome-pacote-flatpak /data/org.kde.syntax-highlighting/syntax/</code>
Para pacotes Snap	<code>\$HOME /snap/ nome-pacote-snap /current/.local/share/org.kde.syntax-highlighting/syntax/</code>
No Windows®	<code>%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\syntax</code>
No macOS®	<code>\$HOME /Library/Application Support/org.kde.syntax-highlighting/syntax/</code>

Se existirem vários arquivos para a mesma linguagem, o arquivo com o atributo de versão mais alto no elemento **language** será carregado.

SEÇÕES PRINCIPAIS DOS ARQUIVOS DE DEFINIÇÕES DE REALCE DO KATEPART

Um arquivo de realce contém um cabeçalho que define a versão do XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

A raiz do arquivo de definição é o elemento **language**. Os atributos disponíveis são:

Atributos necessários:

O **name** define o nome da linguagem. Ele aparece nos respectivos menus e janelas.

O **section** indica a categoria.

O **extensions** define as extensões dos arquivos, como por exemplo, *"*.cpp;*.h"*

O **version** especifica a revisão atual do arquivo de definição em termos de um número inteiro. Sempre que você alterar um arquivo de definição de realce, certifique-se de aumentar esse número.

O **kateversion** indica a última versão suportada pelo KatePart.

Atributos opcionais:

O **mimetype** associa os arquivos do tipo MIME.

O **casesensitive** define se as palavras-chave fazem distinção entre maiúsculas e minúsculas.

O **priority** é necessário se outro arquivo de definições de realce usar as mesmas extensões. Ganhará o que tiver maior prioridade.

O **author** contém o nome do autor e o seu endereço de e-mail.

O **license** contém a licença, que é normalmente a licença MIT para novos arquivos de realce de sintaxe.

O **style** contém a linguagem fornecida e é usado pelos sistemas de recuo para o atributo *required-syntax-style*.

O **indenter** define qual indentador será usado por padrão. Os indentadores disponíveis são: *ada, normal, cstyle, cmake, haskell, latex, lilypond, lisp, lua, pascal, python, replicode, ruby e xml*.

O **hidden** define se o nome deverá aparecer nos menus do KatePart.

Assim, a próxima linha se parece com o seguinte:

```
<language name="C++" version="1" kateversion="2.4" section="Sources" ←  
extensions="*.cpp;*.h" />
```

A seguir vem o elemento **highlighting**, que contém o elemento opcional **list** e os elementos obrigatórios **contexts** e **itemDatas**.

O elemento **list** contém uma lista de palavras-chave. Neste caso, as palavras-chave são a *class* e a *const*. Você poderá adicionar tantas listas quanto desejar.

Desde o KDE Frameworks 5.53, uma lista pode incluir palavras-chave de outra lista ou linguagem/arquivo, usando o elemento **include**. O **##** é usado para separar o nome da lista e o nome da definição da linguagem, da mesma forma que na regra **IncludeRules**. Isso é útil para evitar a duplicação de listas de palavras-chave, caso você precise incluir as palavras-chave de outra linguagem/arquivo. Por exemplo, a lista *othername* contém a palavra-chave *str* e todas as palavras-chave da lista *types*, que pertence à linguagem *ISO C++*.

O elemento **contexts** contém todos os contextos. O primeiro contexto é, por padrão, o início do realce. Existem duas regras no contexto *Normal Text* (Texto Normal), que correspondem à lista de palavras-chave com o nome *um_nome* e uma regra que detecta aspas e

muda o contexto para *string* (cadeia de caracteres). Para aprender mais sobre as regras, leia o próximo capítulo.

A terceira parte é o elemento **itemDatas**. Contém todas as cores e estilos de fonte necessários pelos contextos e regras. Neste exemplo, são usados o **itemData** de *Normal Text* (Texto Normal), *String* (Cadeia de Caracteres) e *Keyword* (Palavra-Chave).

```
<highlighting>
  <list name="somename">
    <item>class</item>
    <item>const</item>
  </list>
  <list name="othername">
    <item>str</item>
    <include>types##ISO C++</include>
  </list>
  <contexts>
    <context attribute="Normal Text" lineEndContext="#pop" name=" ←
      Normal Text" >
      <keyword attribute="Keyword" context="#stay" String="somename" ←
        />
      <keyword attribute="Keyword" context="#stay" String="othername" ←
        />
      <DetectChar attribute="String" context="string" char="&quot;" ←
        />
    </context>
    <context attribute="String" lineEndContext="#stay" name="string" ←
      >
      <DetectChar attribute="String" context="#pop" char="&quot;" />
    </context>
  </contexts>
  <itemDatas>
    <itemData name="Normal Text" defStyleNum="dsNormal" />
    <itemData name="Keyword" defStyleNum="dsKeyword" />
    <itemData name="String" defStyleNum="dsString" />
  </itemDatas>
</highlighting>
```

A última parte de uma definição de realce é a seção opcional **general**. Ela poderá conter informações sobre as palavras-chave, expansão/recolhimento de código, comentários, recuo, linhas vazias e verificação ortográfica.

A seção **comment** define com que texto é introduzido um comentário para uma única linha. Você poderá também definir comentários multilinha, usando o *multiLine* com o atributo adicional *end*. Isto é usado se o usuário pressionar o atalho correspondente para *comentar/descomentar*.

A seção **keywords** define se as listas de palavras-chave fazem distinção entre maiúsculas e minúsculas ou não. Os outros atributos serão explicados mais tarde.

As outras seções, **folding**, **emptyLines** e **spellchecking**, geralmente não são necessárias e serão explicadas posteriormente.

```
<general>
  <comments>
    <comment name="singleLine" start="##"/>
    <comment name="multiLine" start="####" end="####" region=" ←
      CommentFolding"/>
  </comments>
  <keywords casesensitive="1"/>
  <folding indentationsensitive="0"/>
```



```

<emptyLines>
  <emptyLine regexp="\s+"/>
  <emptyLine regexp="\s*#.*/>
</emptyLines>
<spellchecking>
  <encoding char="á" string="\`a"/>
  <encoding char="à" string="\`a"/>
</spellchecking>
</general>
</language>

```

6.2.3.2 As Seções em detalhes

Esta parte irá descrever todos os atributos disponíveis para o 'contexts', o 'itemDatas', o 'keywords', o 'comments', a expansão de código e o recuo.

O elemento **context** pertence ao grupo **contexts**. Um contexto, por si só, define as regras específicas do contexto, como o que deve acontecer se o sistema de realce chegar ao fim de uma linha. Os atributos disponíveis são:

O **name** declara o nome do contexto. As regras irão usar esse nome para indicar o contexto para onde mudar, se a regra corresponder.

O **attribute** identifica o estilo a ser usado para um caractere quando nenhuma regra corresponde ou quando uma regra não especifica um atributo. Neste último caso, o *attribute* do contexto especificado no *context* da regra será usado.

O **lineEndContext** define o contexto para onde o sistema de realce salta, se atingir o fim de uma linha. Poderá ser o nome de outro contexto, o **#stay** para não mudar de contexto (por exemplo, não fazer nada) ou o **#pop** que fará com que saia deste contexto. É possível usar, por exemplo, **#pop#pop#pop** para sair de dentro de três contextos, ou ainda **#pop#pop!OutroContexto** para sair duas vezes e mudar para o contexto **OutroContexto**. Também é possível alternar para um contexto que pertence a outra definição de linguagem, da mesma forma que nas regras **IncludeRules**, por exemplo, **AlgumContexto##JavaScript**. As mudanças de contexto também são descritas em Seção 6.2.4. Padrão: **#stay**.

O **lineEmptyContext** define o contexto, se for encontrada uma linha em branco. A nomenclatura das mudanças de contexto é a mesma que a descrita anteriormente em *lineEndContext*. Padrão: **#stay**.

O **fallthroughContext** especifica o próximo contexto para o qual alternar caso nenhuma regra corresponda. A nomenclatura das mudanças de contexto é a mesma descrita anteriormente em *lineEndContext*. Padrão: **#stay**.

O **fallthrough** define se o sistema de realce alterna para o contexto especificado em **fallthroughContext** caso nenhuma regra corresponda. Observe que, desde o KDE Frameworks 5.62, este atributo está obsoleto em favor de **fallthroughContext**, pois, se o atributo **fallthroughContext** estiver presente, subentende-se implicitamente que o valor de **fallthrough** é **true**. Padrão: **false**.

O **noIndentationBasedFolding** desativa o recolhimento baseado em indentação no contexto. Se o recolhimento baseado em indentação não estiver ativado, este atributo é inútil. Ele é definido no elemento *folding* do grupo *general*. Padrão: **false**.

O elemento **itemData** está no grupo **itemDatas**. Define o estilo e as cores da fonte. Assim, é possível definir os seus próprios estilos e cores. Contudo, recomenda-se que usar os estilos predefinidos, para que o usuário veja sempre as mesmas cores usadas em várias linguagens. Todavia, existem casos em que não existe outra forma e, assim, é necessário mudar os atributos de cores e tipos de fonte. Os atributos 'name' e 'defStyleNum' são obrigatórios, enquanto os outros são opcionais. Os atributos disponíveis são:

O **name** define o nome do 'itemData'. Os contextos e regras irão usar este nome no seu atributo *attribute*, para referenciar um 'itemData'.

O **defStyleNum** define qual o estilo padrão usar. Os estilos pré-definidos disponíveis são explicados mais tarde em detalhes.

O **color** define uma cor. Os formatos válidos são o '#rrggbb' ou 'rgb'.

O **selColor** define a cor da seleção.

O **italic**, se for *true* (verdadeiro), irá colocar o texto em itálico.

O **bold**, se for *true* (verdadeiro), irá colocar o texto em negrito.

O **underline**, se for *true* (verdadeiro), irá colocar o texto sublinhado.

O **strikeOut**, se for *true* (verdadeiro), o texto ficará traçado.

O **spellChecking**, se for *true* (verdadeiro), será verificada a ortografia do texto.

O elemento **keywords**, no grupo **general**, define as propriedades das palavras-chave. Os atributos disponíveis são:

O **casesensitive** poderá ser *true* (verdadeiro) ou *false* (falso). Se for *true*, todas as palavras-chave farão distinção entre maiúsculas e minúsculas.

O **weakDelimiter** é uma lista de caracteres que não irão atuar como separadores de palavras. Por exemplo, o ponto '.' é um separador de palavras. Assuma que uma palavra-chave num **list** contém um ponto; nesse caso, só irá corresponder se indicar que o ponto é um delimitador fraco.

O **additionalDelimiter** define os delimitadores ou separadores adicionais.

O **wordWrapDelimiter** define os caracteres após os quais poderá ocorrer uma mudança de linha.

Os delimitadores pré-definidos e de mudança de linha são os caracteres . () : ! + , - < = > % & * / ; ? [] ^ { | } ~ \, o espaço (' ') e a tabulação ('\t').

O elemento **comment** no grupo **comments** define propriedades de comentário que são usadas para: Ferramentas → Comentar, Ferramentas → Descomentar e Ferramentas → Alternar comentário. Os atributos disponíveis são:

O **name** tanto poderá ser *singleLine* como *multiLine*. Se escolher o *multiLine*, serão necessários os atributos *end* e *region*. Se escolher *singleLine*, poderá adicionar o atributo opcional *position*.

O **start** define o texto usado para iniciar um comentário. No C++, este será o `"/**` em comentários de várias linhas. Este atributo é obrigatório para os tipos *multiLine* e *singleLine*.

O **end** define o texto usado para fechar um comentário. No C++, será o `*/`. Este atributo está disponível apenas e é obrigatório para comentários do tipo *multiLine*.

O **region** deverá ser o nome do comentário multi-linhas que poderá expandir ou recolher. Assuma que tem o *beginRegion*=`"Comment"` ... *endRegion*=`"Comment"` nas suas regras; nesse caso, deverá usar o *region*=`"Comment"`. Desta forma, a remoção de comentários funciona, mesmo que não tenha selecionado todo o texto do comentário multi-linhas. O cursor só precisa estar dentro deste comentário. Este atributo está disponível apenas para o tipo *multiLine*.

O **position** define onde o comentário de linha única é inserido. Por padrão, o comentário de linha única é colocado no início da linha na coluna 0, mas se você usar *position*=`"afterwhitespace"` o comentário é inserido após os espaços em branco iniciais à direita, antes do primeiro caractere que não seja um espaço em branco. Isso é útil para colocar comentários corretamente em linguagens onde a indentação é importante, como Python ou YAML. Este atributo é opcional e o único valor possível é *afterwhitespace*. Isso só está disponível para o tipo *singleLine*.

O elemento **folding**, no grupo **general**, define as propriedades de dobragem/desdobramento do código. Os atributos disponíveis são:

O **indentationsensitive**, se for *true*, aplicará os marcadores de dobragem de código com base no recuo, como acontece na linguagem de programação Python. Normalmente você não terá que definir isto, uma vez que o valor padrão é *false*.

O elemento **emptyLine** no grupo **emptyLines** define quais linhas devem ser tratadas como linhas vazias. Isso permite modificar o comportamento do atributo *lineEmptyContext* nos elementos **context**. Os atributos disponíveis são:

O **regexpr** define uma expressão regular que será tratada como uma linha em branco. Por padrão, linhas em branco não contêm nenhum caractere; portanto, isso adiciona linhas em branco adicionais, por exemplo, se você quiser que linhas com espaços também sejam consideradas linhas em branco. No entanto, na maioria das definições de sintaxe, você não precisa definir esse atributo.

O elemento **encoding** no grupo **spellchecking** define uma codificação de caracteres para a verificação ortográfica. Atributos disponíveis:

O **char** é um caractere codificado.

O **string** é uma sequência de caracteres que será codificada como o caractere *char* na verificação ortográfica. Por exemplo, na linguagem LaTeX, a string `\~{A}` representa o caractere Ä.

6.2.3.3 Estilos Padrão Disponíveis

Os estilos padrão [já foram explicados](#), em resumo: Os estilos padrão são os estilos de cores e fontes pré-definidos.

Estilos padrão gerais:

dsNormal, quando não é necessário nenhum realce em especial.

dsKeyword, para as palavras-chave do idioma incorporadas.

dsFunction, nas chamadas e definições de funções.

dsVariable, se aplicável: nomes das variáveis (por exemplo `$algumaVariavel` em PHP/-Perl).

dsControlFlow, palavras-chave de controle de fluxo, como o `'if'`, `'else'`, `'switch'`, `'break'`, `'return'`, `'yield'`, ...

dsOperator, operadores como o `+`, `-`, `*`, `/`, `::`, `<`, `>`

dsBuiltIn, nas funções, classes e objetos incorporados.

dsExtension, extensões comuns, como as classes e funções/macros do Qt™ em C++ e Python.

dsPreprocessor, para as instruções do pré-processador ou para as definições de macros.

dsAttribute, anotações como o `@override` e o `__declspec(...)`.

Estilos padrão relacionados com as strings:

dsChar, para caracteres únicos, como o `'x'`.

dsSpecialChar, caracteres com significados especiais nas strings, como as sequências de escape, substituições ou operadores de expressões regulares.

dsString, para strings do tipo "olá mundo".

dsVerbatimString, strings literais como o `'raw \backslash'` do Perl, CoffeeScript e das linhas de comando, assim como o `r'\raw'` do Python.

dsSpecialString, SQL, expressões regulares, documentação, modo matemático do L^AT_EX, ...

dsImport, importação, inclusão ou requisição de módulos.

Estilos padrão relacionados com números:

dsDataType, para os tipos de dados incorporados, como o 'int', 'void', 'u64'.

dsDecVal, nos valores decimais.

dsBaseN, nos valores com uma base diferente de 10.

dsFloat, nos valores de ponto flutuante.

dsConstant, nas constantes incorporadas e definidas pelo usuário, como o PI.

Estilos padrão de comentários e relacionados com a documentação:

dsComment, para comentários.

dsDocumentation, para `/**` Comentários de documentação `*/` ou `"""` strings de documentação `"""`.

dsAnnotation, para os comandos de documentação como o `@param`, `@brief`.

dsCommentVar, os nomes das variáveis usadas nos comandos acima, como o "foobar" no `@param foobar`.

dsRegionMarker, para marcadores de região, como o `//BEGIN`, `//END` nos comentários.

Outros estilos padrão:

dsInformation, notas e dicas do tipo `@note` no doxygen.

dsWarning, para os avisos do tipo `@warning` no doxygen.

dsAlert, para palavras especiais como o `TODO`, `FIXME`, `XXXX`.

dsError, para realçar erros e sintaxes inválidas.

dsOthers, quando nada mais se aplica.

6.2.4 Regras de Detecção de Realce

Esta seção descreve as regras de detecção de sintaxe.

Cada regra pode corresponder a zero ou mais caracteres no início do texto que é testado. Se a regra corresponder, é atribuído o estilo ou *atributo* definido pela regra aos caracteres correspondentes; uma regra poderá perguntar se o contexto atual será alterado.

As regras se parecem com isto:

```
<NomeRegra attribute="(identificador)" context="(identifier)" [atributos ←  
    específicos da regra] />
```

O *attribute* identifica o estilo a usar para os caracteres correspondentes pelo nome ou índice; o *context* identifica, como esperado, o contexto a usar a partir daqui.

O *context* pode ser identificado por:

- Um *identificador*, que é o nome do outro contexto.
- Uma *ordem* diz ao mecanismo para ficar no contexto atual (**#stay**), ou voltar a usar um contexto usado anteriormente na string (**#pop**). Um contexto vazio ou ausente é equivalente a **#stay**.

Para voltar mais passos, a palavra-chave `#pop` pode ser repetida: **#pop#pop#pop**

- Uma *ordem* seguida de um ponto de exclamação (!) e um *identificador*, que fará com que o mecanismo siga primeiro a ordem e depois mude para o outro contexto, por exemplo **#pop#pop!OtherContext**.

- Um *identificador*, que é um nome de contexto, seguido por dois hashes (##) e outro *identificador*, que é o nome de uma definição de linguagem. Essa nomenclatura é similar à usada nas regras **IncludeRules** e permite que você alterne para um contexto pertencente a outra definição de realce de sintaxe, por exemplo, **AlgunContexto##JavaScript**.

Os atributos específicos da regra variam e estão descritos nas seções a seguir.

ATRIBUTOS COMUNS

- *attribute*: Um atributo mapeia-se para um *itemData* definido. Padrão: *attribute* do contexto especificado no atributo *context*.
- *context*: Indica o contexto para onde muda o sistema de realce, se a regra corresponder. Padrão: *#stay*.
- *beginRegion*: Inicia um bloco de dobragem de código. Padrão: não definido.
- *endRegion*: Fecha uma região de dobragem de código. Padrão: não definido.
- *lookAhead*: Se for *true* (verdadeiro), o sistema de realce não irá processar o tamanho da correspondência. Padrão: *false*.
- *firstNonSpace*: Corresponder apenas se o texto for o primeiro não-espaco em branco da linha. Padrão: *false*.
- *column*: Corresponder apenas se a coluna corresponder. Padrão: não definido.

REGRAS DINÂMICAS

- *dynamic*: poderá ser (*true* | *false*).

Como isto funciona:

Nas [expressões regulares](#) das regras **RegExpr**, todo o texto entre colchetes curvos simples (**PADRÃO**) é capturado e armazenado. Essas capturas podem ser usadas no contexto em que são alternadas, nas regras com o atributo **dynamic true**, por meio de *%N* (em *String*) ou *N* (em *char*).

É importante mencionar que um texto capturado em uma regra **RegEx** é armazenado apenas para o contexto alterado, especificado em seu atributo **context**.

DICA

- Se as capturas não forem usadas, tanto por regras dinâmicas quanto na mesma expressão regular, grupos não capturantes devem ser usados: **(?:PADRÃO)**
Os grupos *lookahead* ou *lookbehind*, como **(?=PADRÃO)**, **(?!PADRÃO)** ou **(?<=PADRÃO)**, não são capturados. Consulte [Expressões regulares](#) para obter mais informações.
- Os grupos de captura podem ser usados dentro da mesma expressão regular, usando *\N* em vez de *%N* respectivamente. Para mais informações, consulte [Capturando texto correspondente \(referências anteriores\)](#) em [Expressões regulares](#).

Exemplo 1:

Neste exemplo simples, o texto correspondente à expressão regular **=*** é capturado e inserido em *%1* na regra dinâmica. Isso permite que o comentário termine com a mesma quantidade de **=** que no início. Isso corresponde a textos como: **[[comentário]]**, **[=[comentário]=]** ou **[===== [comentário]=====]**.

Além disso, as capturas estão disponíveis apenas no contexto alternado para *Comentário de várias linhas*.

Manual do KatePart

```
<context name="Normal" attribute="Normal Text" lineEndContext="#stay">
  <Regex context="Multi-line Comment" attribute="Comment" String="\[ (=*) ↵
    \[" beginRegion="RegionComment"/>
</context>
<context name="Multi-line Comment" attribute="Comment" lineEndContext="# ↵
  stay">
  <StringDetect context="#pop" attribute="Comment" String="]\%1]" dynamic=" ↵
    true" endRegion="RegionComment"/>
</context>
```

Exemplo 2:

Na regra dinâmica, `%1` corresponde à captura que corresponde a `#+`, e `%2` a `"+;`. Isso corresponde ao texto como: `#label~~~~dentro do contexto~~~~#`.

Essas capturas não estarão disponíveis em outros contextos, como *OtherContext*, *FindEscapes* ou *SomeContext*.

```
<context name="SomeContext" attribute="Normal Text" lineEndContext="#stay">
  <Regex context="#pop!NamedString" attribute="String" String="(#+) (?:[\\w ↵
    -]|[^[[:ascii:]])(&quot;+;)" />
</context>
<context name="NamedString" attribute="String" lineEndContext="#stay">
  <Regex context="#pop!OtherContext" attribute="String" String="%2(?:%1 ↵
    ?" dynamic="true"/>
  <DetectChar context="FindEscapes" attribute="Escape" char="\" />
</context>
```

Exemplo 3:

Isso corresponde a textos como: **Class::function<T>(...)**.

```
<context name="Normal" attribute="Normal Text" lineEndContext="#stay">
  <Regex context="FunctionName" lookAhead="true"
    String="\b([a-zA-Z_][\w-]*) (?:) ([a-zA-Z_][\w-]*) (?:&lt;[\\w-\\ ↵
    s]*&gt;)?(\()" />
</context>
<context name="FunctionName" attribute="Normal Text" lineEndContext="#pop">
  <StringDetect context="#stay" attribute="Class" String="%1" dynamic="true ↵
    "/>
  <StringDetect context="#stay" attribute="Operator" String="%2" dynamic=" ↵
    true"/>
  <StringDetect context="#stay" attribute="Function" String="%3" dynamic=" ↵
    true"/>
  <DetectChar context="#pop" attribute="Normal Text" char="4" dynamic="true ↵
    "/>
</context>
```

DELIMITADORES LOCAIS

- *weakDelimiter*: lista de caracteres que não atuam como delimitadores de palavras.
- *additionalDelimiter*: define os delimitadores ou separadores adicionais.

6.2.4.1 As regras em detalhes

DetectChar

Detecta um caractere específico. Comumente usado, por exemplo, para encontrar o final das strings citadas.

```
<DetectChar char="(caractere)" (atributos comuns) (dinâmico) />
```

O atributo **char** define o caractere a ser procurado.

Detect2Chars

Detecta dois caracteres específicos, em uma ordem definida.

```
<Detect2Chars char="(caractere)" char1="(caractere)" (atributos comuns) ↔ />
```

O atributo **char** define o primeiro caractere a ser procurado, e o atributo **char1** o segundo. Esta regra existe por razões históricas e, para facilitar a leitura, é preferível usar **StringDetect**.

AnyChar

Detecta um caractere de um conjunto de caracteres especificados.

```
<AnyChar String="(string)" (atributos comuns) />
```

O atributo **String** define o conjunto de caracteres.

StringDetect

Detecta uma string exata.

```
<StringDetect String="(string)" [insensitive="true|false;"] (atributos ↔ comuns) (dinâmico) />
```

O atributo **String** define a sequência a encontrar. O atributo **insensitive** é por padrão *false* e é passado à função de comparação de cadeias de caracteres. Se o valor for *true* a comparação não faz distinção entre maiúsculas e minúsculas.

WordDetect

Detecta um texto exato, mas obriga adicionalmente a que esteja rodeado por limites de palavras, como um ponto '.' ou um espaço em branco no início e no fim da palavra. Pense em **\b<texto>\b** em termos de uma expressão regular, só que é mais rápido que a regra **Exp Reg**.

```
<WordDetect String="(string)" [insensitive="true|false;"] (atributos ↔ comuns) (delimitadores locais) />
```

O atributo **String** define a sequência a encontrar. O atributo **insensitive** é por padrão *false* e é passado à função de comparação de cadeias de caracteres. Se o valor for *true* a comparação não faz distinção entre maiúsculas e minúsculas.

Desde: Kate 3.5 (KDE 4.5)

RegExp

Procura por uma expressão regular.

```
<RegExp String="(texto)" [insensitive="true|false"] [minimal="true| ↔ false"] (atributos comuns) (dinâmico) />
```

O atributo **String** define a expressão regular.

O **insensitive** é, por padrão, *false* e é passado ao motor de expressões regulares.

O **minimal** é, por padrão, *false* e é passado ao motor de expressões regulares.

Pelo motivo que as regras estão sempre iniciando a busca no início da string atual, uma expressão regular iniciada com um acento circunflexo (^) indica que a regra deve ser coincidente somente no início da linha.

Veja em [Expressões Regulares](#) mais informações sobre o assunto.

keyword

Detecta uma palavra-chave de uma lista especificada.

```
<keyword String="(nome da lista)" (atributos comuns) (delimitadores locais) />
```

O atributo **String** identifica a lista de palavras-chave pelo nome. Uma lista com aquele nome, portanto, deve existir.

O sistema de realce processa as regras da palavra-chave de forma muito otimizada. Isto a torna uma necessidade absoluta em que quaisquer palavras-chave a corresponder sejam envolvidas por separadores definidos, sejam eles implícitos (os separadores padrão) ou explícitos, com a propriedade *additionalDelimiter* da marca *keywords*.

Se uma palavra-chave a corresponder conter um caractere separador, este caractere respectivo deverá ser adicionado à propriedade *weakDelimiter* da marca *keywords*. Este caractere irá então perder a sua propriedade de separador em todas as regras *keyword*. Também é possível usar o atributo *weakDelimiter* de *keyword* para que esta modificação se aplique apenas a esta regra.

Int

Detecta um número inteiro (como a expressão regular: **\b[0-9]+**).

```
<Int (atributos comuns) (delimitadores locais) />
```

Esta regra não possui atributos específicos.

Float

Detecta um número de ponto flutuante (como a expressão regular: **(\b[0-9]+\.[0-9]*|\.[0-9]+)([eE][-+]?[0-9]+)?**).

```
<Float (atributos comuns) (delimitadores locais) />
```

Esta regra não possui atributos específicos.

HILOct

Detecta uma representação de número de ponto octal (como a expressão regular: **\b0[0-7]+**).

```
<HILOct (atributos comuns) (delimitadores locais) />
```

Esta regra não possui atributos específicos.

HIHex

Detecta uma representação numérica hexadecimal (como uma expressão regular: **\b0[xX][0-9a-fA-F]+**).

```
<HIHex (atributos comuns) (delimitadores locais) />
```

Esta regra não possui atributos específicos.

HLCStringChar

Detecta um caractere de escape.

```
<HLCStringChar (atributos comuns) />
```

Esta regra não possui atributos específicos.

Corresponde a representações literais dos caracteres usados normalmente no código do programa como, por exemplo, o `\n` (nova linha) ou o `\t` (TAB).

Os seguintes caracteres irão corresponder se estiverem após uma barra invertida (`\`): **a b e f n r t v " ' ? **. Além disso, os números escapados em hexadecimal como, por exemplo, o `\xff` e os números octais escapados, como o `\033`, irão corresponder.

HLCChar

Detecta um caractere do C.

```
<HLCChar (atributos comuns) />
```

Esta regra não possui atributos específicos.

Corresponde aos caracteres em C colocados dentro de um tique (Exemplo: `'c'`). Os tiques podem ser um único caractere ou um caractere de escape. Veja o `'HLCStringChar'` para ver as sequências de caracteres de escape correspondentes.

RangeDetect

Detecta uma string com os caracteres de início e fim definidos.

```
<RangeDetect char="(caractere)" char1="(caractere)" (atributos comuns) ←  
/>
```

O **char** define o caractere de início e o **char1** o caractere que termina o intervalo.

Útil para detectar, por exemplo, pequenas cadeias de caracteres entre aspas e semelhantes, mas repare que, uma vez que o motor de realce de sintaxe funciona com uma linha de cada vez, isto não irá encontrar as cadeias de caracteres que se prolonguem por mais de uma linha.

LineContinue

Corresponde um caractere específico indicado no fim de uma linha.

```
<LineContinue (atributos comuns) [char="\"] />
```

O atributo **char** define o caractere opcional a ser encontrado, sendo a barra invertida (`'\'`). o padrão. Novidade da versão 4.13 do KDE.

Esta regra é útil para mudar de contexto no fim da linha. Isso é necessário no C/C++, por exemplo, para continuar as macros ou cadeias de caracteres.

IncludeRules

Inclui as regras de outro contexto ou linguagem/arquivo.

```
<IncludeRules context="ligacao_contexto" [includeAttrib="true|false"] ←  
/>
```

O atributo **context** define o contexto a incluir.

Se for texto simples, inclui todas as regras definidas no contexto atual, como por exemplo:

```
<IncludeRules context="outroContexto" />
```

Se o texto conter um **##**, o sistema de realce irá procurar por um contexto de outra definição de linguagem com o nome indicado, por exemplo,

```
<IncludeRules context="String##C++" />
```

iria incluir o contexto *String* da definição de realce C++.

Se o atributo **includeAttrib** for *true*, muda o atributo de destino para o da origem. Isto é necessário para fazer, por exemplo, funcionar os comentários, se o texto correspondente ao contexto incluído for de um realce diferente do contexto-anfitrião.

DetectSpaces

Detecta espaços em branco.

```
<DetectSpaces (atributos comuns) />
```

Esta regra não possui atributos específicos.

Use esta regra se souber que poderão existir vários espaços em branco à frente como, por exemplo, no início das linhas recuadas. Esta regra irá ignorar todos os espaços em branco, em vez de testar várias regras e ignorar uma de cada vez, devido a uma falta de correspondência.

DetectIdentifier

Detecta os textos dos identificadores (como a expressão regular: **[a-zA-Z_][a-zA-Z0-9_]***).

```
<DetectIdentifier (atributos comuns) />
```

Esta regra não possui atributos específicos.

Use esta regra para ignorar uma sequência de caracteres de palavras de uma vez, em vez de testar com várias regras e ignorar uma de cada vez, por falta de correspondência.

6.2.4.2 Dicas & Truques

- As expressões regulares são fáceis de usar mas, normalmente, existe outra forma muito mais rápida de obter o mesmo resultado. Assuma que só deseja corresponder com o caractere '#' se for o primeiro caractere da linha. Uma solução baseada em expressões regulares seria semelhante à seguinte:

```
<RegExpr attribute="Macro" context="macro" String="^\s*#" />
```

Você poderá obter o mesmo se usar:

```
<DetectChar attribute="Macro" context="macro" char="#" firstNonSpace="↔
true" />
```

Se quiser corresponder à expressão regular '**^#**', poderá usar ainda o **DetectChar** com o atributo **column="0"**. O atributo **column** conta como caracteres; assim, uma tabulação conta como se fosse apenas um caractere.

- Nas regras **RegExpr**, use o atributo **column="0"** se o padrão **^PADRÃO** for usado para encontrar correspondências no início de uma linha. Isso melhora o desempenho, pois evita a busca por correspondências nas demais colunas.
- Em expressões regulares, use grupos não capturantes (**?:PADRÃO**) em vez de grupos capturantes (**PADRÃO**), se as capturas não forem usadas na mesma expressão regular ou em regras dinâmicas. Isso evita o armazenamento desnecessário de capturas.
- Você poderá mudar de contextos sem processar os caracteres. Assuma que deseja mudar de contexto quando encontrar o texto ***/**, mas necessita de processar essa sequência no próximo contexto. A regra abaixo irá corresponder e o atributo **lookAhead** fará com que o sistema de realce mantenha o texto correspondente no próximo contexto.

```
<StringDetect attribute="Comment" context="#pop" String="*/" lookAhead="↔"
true" />
```

- Use o **DetectSpaces** se você souber que irão ocorrer vários espaços em branco.
- Use o **DetectIdentifier** em vez da expressão regular '`[a-zA-Z_]\w*`'.
- Use os estilos padrão sempre que puder. Desta forma, o usuário irá encontrar um ambiente familiar.
- Procure em outros arquivos XML para ver como as outras pessoas implementam as regras mais complicadas.
- Você pode validar todos os arquivos XML usando o comando **validatehl.sh mySyntax.xml**. O arquivo `validatehl.sh` usa `language.xsd` que estão disponíveis no [Repositório de realce de sintaxe](#).
- Se repetir algumas expressões regulares complexas com frequência, você poderá usar as **ENTIDADES**. Por exemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE language
[
    <!ENTITY myref      "[A-Za-z_:] [\w.:_-]*">
]>
```

Agora, você poderá usar o *&referencia;* em vez da expressão regular.

- No editor do Kate, você pode recarregar as sintaxes usando a linha de comando integrada (atalho **F7** por padrão) e o comando **reload-highlighting**.
- Você pode usar o utilitário de linha de comando chamado **ksyntaxhighlighter6** (**kate-syntax-highlighter** em versões mais antigas) para testar uma sintaxe e exibir o estilo e as regiões associadas a cada parte de um texto.

Resultado de `ksyntaxhighlighter6 --output-format=ansi --syntax-trace=format test.cpp`.

Use `ksyntaxhighlighter6 -h` para mais opções.

6.3 Trabalhando com temas de cores

6.3.1 Introdução

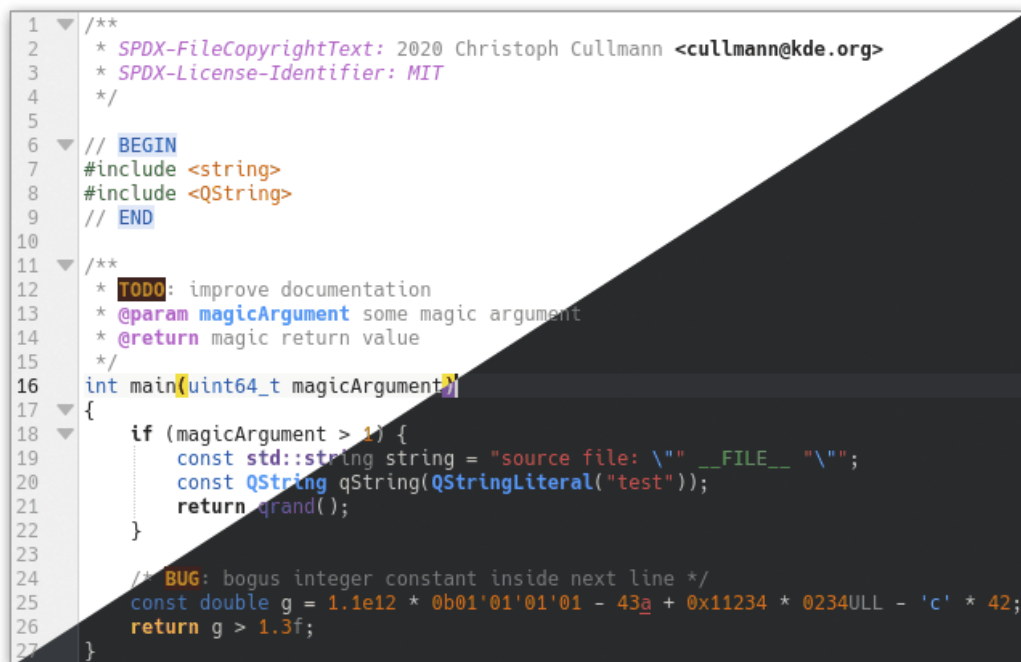
Os temas de cores definem as cores da *área de edição de texto* e do *realce de sintaxe*. Um tema de cores engloba o seguinte:

- O estilo do texto, usado para realce de sintaxe através dos *atributos de estilo padrão*. Por exemplo, a cor do texto e a cor do texto selecionada.

- O plano de fundo da área de edição de texto, incluindo a seleção de texto e a linha atual.
- A borda do ícone da área de texto: seu fundo, a linha separadora, os números de linha, os marcadores de quebra de linha, as marcas de linha modificadas e o recolhimento de código.
- Decoradores de texto, como marcadores de pesquisa, recuo e marcas de tabulação/espço, a correspondência de parênteses e a verificação ortográfica.
- Favoritos e trechos de código.

Para evitar confusão, o seguinte está fora do escopo:

- O tipo e o tamanho da fonte.
- As cores do aplicativo de edição de texto, como o mapa da barra de rolagem, os menus, a barra de abas, a cor da janela, etc. Em aplicativos do KDE, como o Kate ou o KDevelop, essas cores são definidas pelo **esquema de cores global do KDE Plasma**, que é configurado no módulo **Cores em Configurações do sistema** ou no próprio aplicativo, no menu **Configurações** → **Esquema de cores**.



```

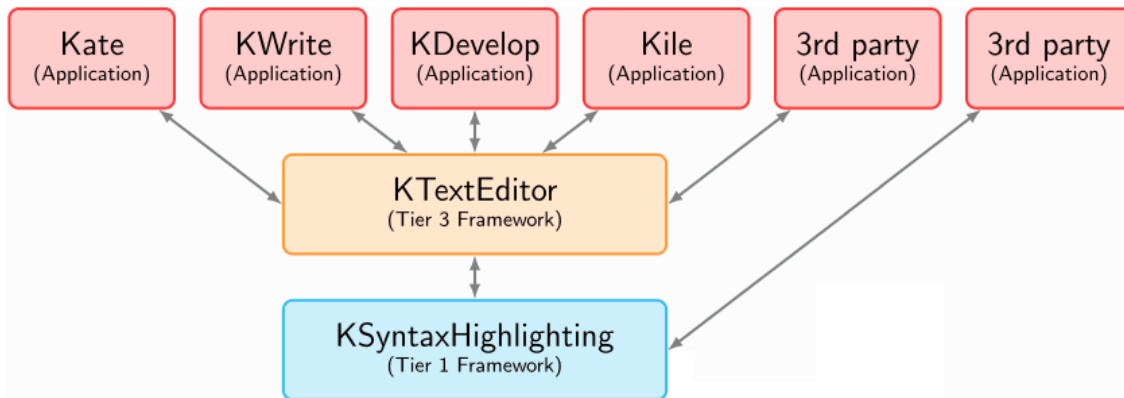
1  /**
2   * SPDX-FileCopyrightText: 2020 Christoph Cullmann <cullmann@kde.org>
3   * SPDX-License-Identifier: MIT
4   */
5
6  // BEGIN
7  #include <string>
8  #include <QString>
9  // END
10
11 /**
12  * TODO: improve documentation
13  * @param magicArgument some magic argument
14  * @return magic return value
15  */
16 int main(uint64_t magicArgument)
17 {
18     if (magicArgument > 1) {
19         const std::string string = "source file: \"__FILE__\"";
20         const QString qString(QStringLiteral("test"));
21         return qrand();
22     }
23
24     /* BUG: bogus integer constant inside next line */
25     const double g = 1.1e12 * 0b01'01'01'01 - 43a + 0x11234 * 0234ULL - 'c' * 42;
26     return g > 1.3f;
27 }

```

Temas de cores 'Breeze Light' e 'Breeze Dark' com realce de sintaxe para 'C++'.

6.3.2 O temas de cores do KSyntaxHighlighting

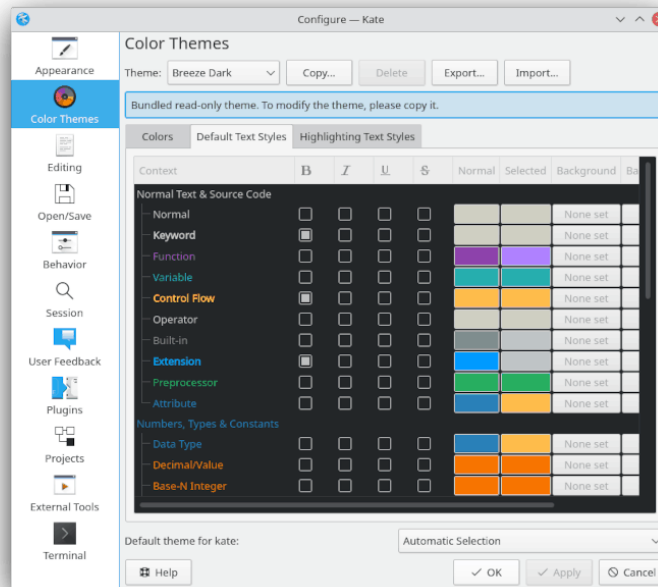
O framework **KSyntaxHighlighting**, que é o **mecanismo de realce de sintaxe**, é a biblioteca que **fornece e gerencia os temas de cores**. Isso faz parte do KDE Frameworks e é usado no editores de texto do KDE como o **Kate**, **KWrite**, **Kile** e **KDevelop**. Essa dependência se parece com o seguinte:



Dependência das bibliotecas do KDE Frameworks em editores de texto.

O KSyntaxHighlighting inclui uma variedade de temas integrados que são exibidos na página [Temas de cores do site do editor Kate](#).

O framework [KTextEditor](#), que é o mecanismo de edição de texto, fornece uma interface de usuário para criar e editar temas de cores, incluindo uma ferramenta para importar e exportar temas. Esta é a maneira mais simples de criar e editar temas; você pode acessá-la através da janela ‘[Configurar](#)’ do editor de texto. Mais detalhes em Seção 6.3.5.



A GUI para gerenciar temas de cores nas configurações do Kate.

É importante mencionar que, nos editores de texto do KDE, como o Kate ou o KDevelop, os temas de cores do KSyntaxHighlighting são usados ​​ desde o [KDE Frameworks 5.75](#), lançado em 10 de outubro de 2020. Anteriormente, os esquemas de cores do Kate (configuração de esquema baseada em KConfig) eram usados ​​e agora estão obsoletos. No entanto, é possível converter os antigos esquemas do Kate para os temas de cores do KSyntaxHighlighting. O [repositório do KSyntaxHighlighting](#) inclui o script `utils/kateschema_to_theme_converter.py` e o utilitário `utils/schema-converter/` para essa finalidade.

6.3.3 Formato JSON dos temas de cores

6.3.3.1 Introdução

Os temas de cores são armazenados em arquivos no formato JSON, com a extensão **.theme**.

No código-fonte do [KSyntaxHighlighting](#), os arquivos JSON dos temas integrados estão localizados no diretório **data/themes/**. Observe que nos editores de texto, os temas integrados são compilados na biblioteca KSyntaxHighlighting; portanto, a maneira de acessá-los é por meio do código-fonte ou [exportando-os da GUI para gerenciar temas do KTextEditor](#).

Também é possível adicionar facilmente temas adicionais ou personalizados, que são carregados do sistema de arquivos. Os arquivos de tema personalizados pelo usuário estão localizados na pasta `org.kde.syntax-highlighting/themes/` na sua pasta de usuário, que você pode encontrar com o comando **qtpaths --paths GenericDataLocation** e geralmente são `$HOME/.local/share/` e `/usr/share/`.

Em pacotes Flatpak e Snap, a pasta acima não funcionará, pois o local dos dados é diferente para cada aplicativo. Em um aplicativo Flatpak, o local dos arquivos de tema personalizados geralmente é `$HOME/.var/app/nome-pacote-flatpak/data/org.kde.syntax-highlighting/themes/` e em um aplicativo Snap, esse local é `$HOME/snap/nome-pacote-snap/current/.local/share/org.kde.syntax-highlighting/themes/`.

No Windows®, esses arquivos estão localizados em `%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\themes`. `%USERPROFILE%` geralmente se expande para `C:\Users\nome-de-usuário`.

Em resumo, para a maioria das configurações, a pasta de temas personalizados é a seguinte:

Para o usuário local	<code>\$HOME/.local/share/org.kde.syntax-highlighting/themes/</code>
Para todos os usuários	<code>/usr/share/org.kde.syntax-highlighting/themes/</code>
Para pacotes Flatpak	<code>\$HOME/.var/app/nome-pacote-flatpak/data/org.kde.syntax-highlighting/themes/</code>
Para pacotes Snap	<code>\$HOME/snap/nome-pacote-snap/current/.local/share/org.kde.syntax-highlighting/themes/</code>
No Windows®	<code>%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\themes</code>
No macOS®	<code>\$HOME/Library/Application Support/org.kde.syntax-highlighting/themes/</code>

Se existirem vários arquivos de tema com o mesmo nome, o arquivo com a **revisão** mais alta será carregado.

6.3.3.2 A estrutura JSON

A estrutura de um arquivo JSON é explicada em [seu site](#). Basicamente, um arquivo no formato JSON consiste em:

- Coleções de pares chave/valor, separados por vírgulas e agrupados em `{ }` que chamaremos de 'objetos'.
- Listas ordenadas de valores, separados por vírgulas e agrupados em `[ ]` que chamaremos de 'array'.

A nomenclatura ‘chave’, ‘valor’, ‘objeto’ e ‘array’ será usada neste artigo. Se esta for a sua primeira vez trabalhando com arquivos JSON, entendê-los é tão simples quanto observar os exemplos abaixo.

6.3.3.3 Seções principais dos arquivos de tema de cores JSON

O objeto raiz do arquivo JSON do tema de cores contém o seguinte esquema de chaves:

- **metadata:** É obrigatória. O valor é um objeto com os metadados do tema, como nome, revisão e licença.
Isto é detalhado em Seção 6.3.3.4.
- **editor-colors:** É obrigatória. O valor é um objeto com as cores da área de edição de texto, como o fundo, a borda do ícone e a decoração do texto.
Isto é detalhado em Seção 6.3.4.1.
- **text-styles:** É obrigatória. O valor é um objeto com os atributos de *estilo de texto padrão* da sintaxe realce de sintaxe. Cada atributo define sua *cor do texto*, sua *cor do texto selecionada* ou se está em *negrito* ou *itálico*, por exemplo. Os estilos de texto podem ser referenciados a partir dos [arquivos XML de definição de sintaxe](#).
Isto é detalhado em Seção 6.3.4.2.
- **custom-styles:** É opcional. Define estilos de texto para os atributos de definições específicas de realce de sintaxe. Por exemplo, em uma definição de realce como **Python** ou **Markdown** você pode especificar um estilo de texto diferente que substitui o padrão definido em **text-styles**.
Isto é detalhado em Seção 6.3.4.3.

A linguagem JSON não suporta comentários. No entanto, você pode usar a chave opcional **_comments** no objeto raiz para escrever comentários. Por exemplo, se você estiver adaptando um tema existente, pode inserir a URL do repositório original. A maneira mais prática é usar um array de strings.

Abaixo está um arquivo de exemplo para o tema ‘Breeze Light’. Você pode notar que, para evitar que o exemplo fique muito grande, os objetos **editor-colors** e **text-styles** não contêm todas as chaves necessárias. Você pode ver o arquivo completo do [tema Breeze Light no repositório do KSyntaxHighlighting](#).

```
{
  "_comments": [
    "Isto é um comentário.",
    "Se o tema for uma adaptação de outro, coloque o link para o  ←
      repositório original."
  ],
  "metadata": {
    "name" : "Breeze Light",
    "revision" : 5,
    "copyright": [
      "SPDX-FileCopyrightText: 2016 Volker Krause <vkrause@kde.org>",
      "SPDX-FileCopyrightText: 2016 Dominik Haumann <dhaumann@kde.org ←
        >"
    ],
    "license": "SPDX-License-Identifier: MIT"
  },
  "editor-colors": {
    "BackgroundColor" : "#ffffff",
    "CodeFolding" : "#94caef",
```

Manual do KatePart

```
"BracketMatching" : "#ffff00",
"CurrentLine" : "#f8f7f6",
"IconBorder" : "#f0f0f0",
"IndentationLine" : "#d2d2d2",
"LineNumbers" : "#a0a0a0",
"CurrentLineNumber" : "#1e1e1e",
As outras chaves de cor do editor...

},
"text-styles": {
  "Normal" : {
    "text-color" : "#1f1c1b",
    "selected-text-color" : "#ffffff",
    "bold" : false,
    "italic" : false,
    "underline" : false,
    "strike-through" : false
  },
  "Keyword" : {
    "text-color" : "#1f1c1b",
    "selected-text-color" : "#ffffff",
    "bold" : true
  },
  "Function" : {
    "text-color" : "#644a9b",
    "selected-text-color" : "#452886"
  },
  "Variable" : {
    "text-color" : "#0057ae",
    "selected-text-color" : "#00316e"
  },
  As outras chaves de estilo de texto...
},
"custom-styles": {
  "ISO C++": {
    "Data Type": {
      "bold": true,
      "selected-text-color": "#009183",
      "text-color": "#00b5cf"
    },
    "Keyword": {
      "text-color": "#6431b3"
    }
  },
  "YAML": {
    "Attribute": {
      "selected-text-color": "#00b5cf",
      "text-color": "#00b5cf"
    }
  }
}
```


6.3.3.4 Metadados

O objeto JSON da chave **metadata** contém informações relevantes sobre o tema. Este objeto possui as seguintes chaves:

- **nome:** É uma *string* que define o nome da linguagem. Ela aparece nos menus e diálogos posteriormente. É obrigatória.
- **revision:** É um *número inteiro* que especifica a revisão atual do arquivo de tema. Sempre que você atualizar um arquivo de tema de cores, certifique-se de incrementar esse número. É obrigatória.
- **license:** É uma *string* que define a licença do tema, usando o identificador **SPDX-License-Identifier** do formato de comunicação de licença padrão **SPDX**. É opcional.
Você pode ver a lista completa de identificadores de licença SPDX [aqui](#).
- **copyright:** É uma *array* de *strings* que especifica os autores do tema, usando o identificador **SPDX-FileCopyrightText** do formato de comunicação de licença padrão **SPDX**. É opcional.

```
"metadata": {
  "name" : "Breeze Light",
  "revision" : 5,
  "copyright": [
    "SPDX-FileCopyrightText: 2016 Volker Krause <vkrause@kde.org>",
    "SPDX-FileCopyrightText: 2016 Dominik Haumann <dhaumann@kde.org>"
  ],
  "license": "SPDX-License-Identifier: MIT"
}
```

6.3.4 Cores em detalhes

Esta seção detalha todos os atributos de cor disponíveis e as configurações de cor disponíveis.

6.3.4.1 Cores do editor

Corresponde às cores da [área de edição de texto](#).

No [arquivo de tema JSON](#), a chave respectiva **editor-colors** tem como valor um *objeto* onde cada chave referencia um atributo de cor do editor de texto. Aqui, **todas as chaves disponíveis são obrigatórias**, seus valores são **strings** com códigos de cores hexadecimais, como `'#00B5CF'`.

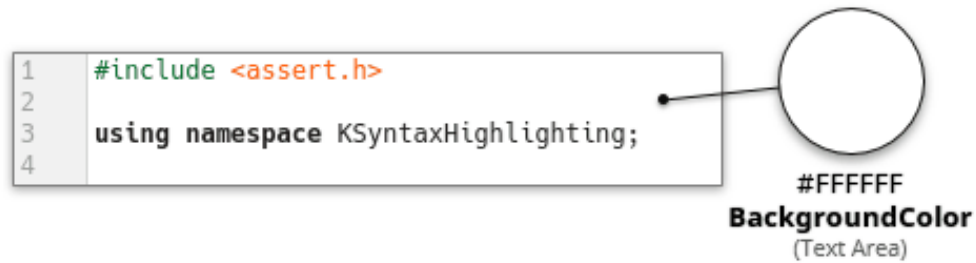
Na [GUI para gerenciar temas do KTextEditor](#), esses atributos podem ser modificados na aba **Cores**.

As chaves disponíveis são as seguintes; as chaves usadas no arquivo [JSON](#) estão listadas em *negrito*, os nomes usados na [GUI](#) são mostrados entre parênteses.

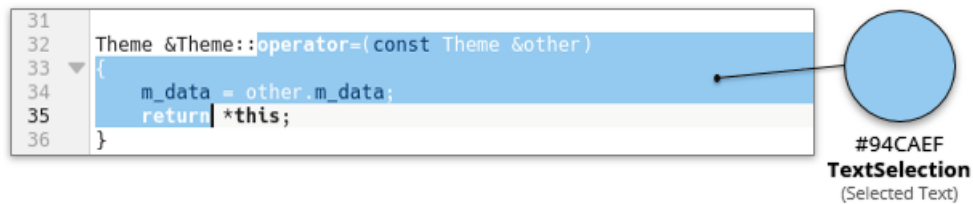
Cores de fundo do editor

BackgroundColor (Área do texto)

Este é o plano de fundo padrão para a área do editor; será a cor dominante.

**TextSelection (Texto selecionado)**

Este é o fundo para o texto selecionado.

**CurrentLine (Linha atual)**

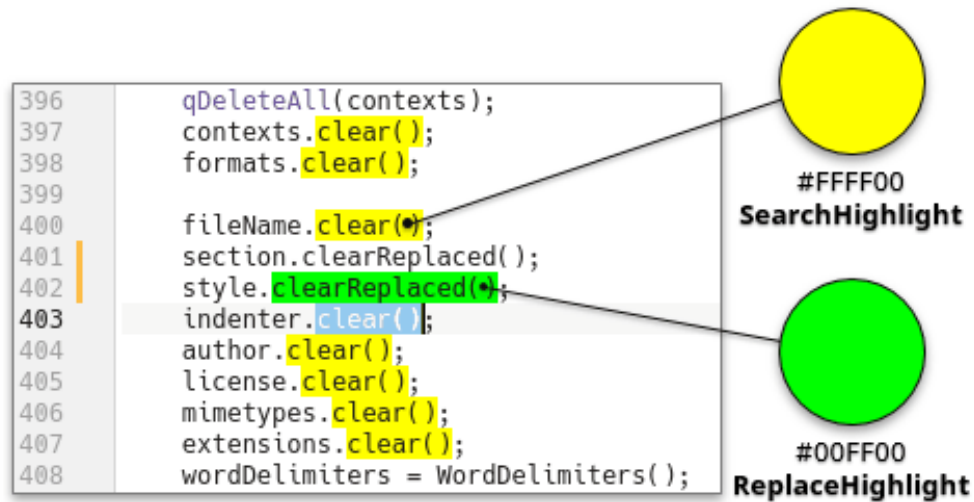
Define a cor da linha atual. Se configurar esta cor um pouco diferente do fundo do texto normal, isto o ajudará a concentrar-se na linha atual.

**SearchHighlight (Pesquisar realce)**

Define a cor do texto que corresponda à sua última pesquisa.

**ReplaceHighlight (Substituir realce)**

Define a cor do texto que corresponda à sua última operação de substituição.



Borda do ícone

IconBorder (Área de fundo)

Esta cor é usada para as marcas, números de linhas e bordas das marcações de desdobramento do lado esquerdo da janela de edição, quando forem apresentados.

LineNumbers (Números de linha)

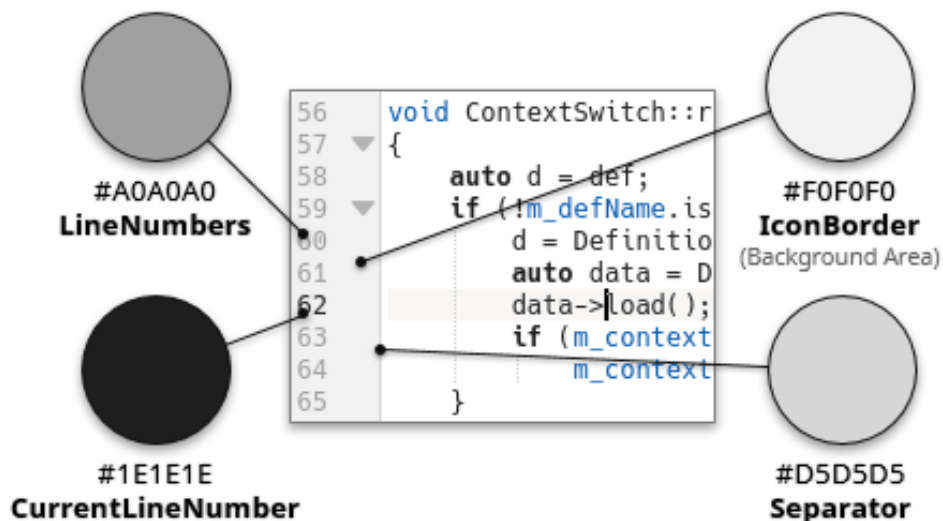
Esta cor é usada para desenhar os números de linha do lado esquerdo da janela, quando estiverem visíveis.

CurrentLineNumber (Número da linha atual)

Esta cor é usada para desenhar o número da linha atual, no lado esquerdo da visualização. Definir isso um pouco diferente de 'LineNumbers' ajuda a manter o foco na linha atual.

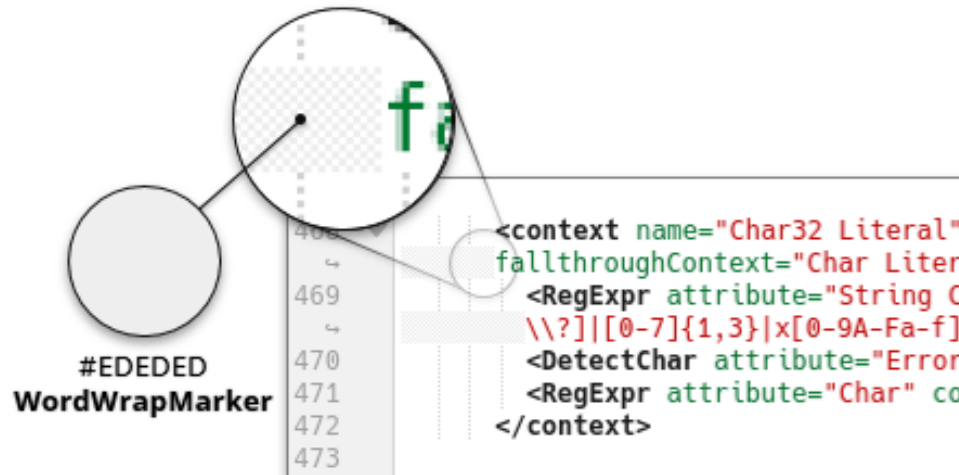
Separator (Separador)

Esta cor é usada para desenhar a linha vertical que separa a borda do ícone do fundo da área de texto.



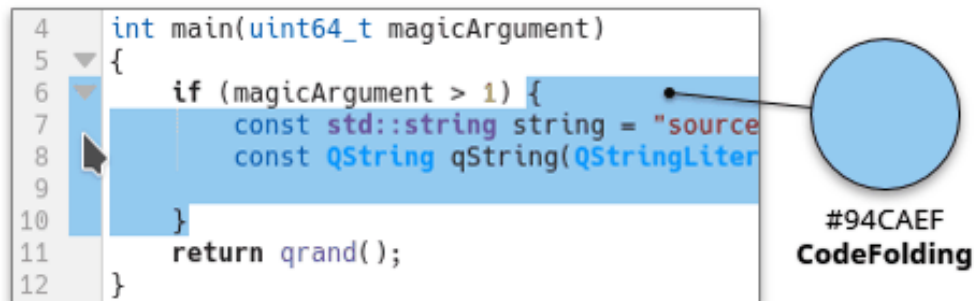
WordWrapMarker (Marcador de quebra de linha)

Esta cor é usada para desenhar um padrão à esquerda das linhas repartidas com quebra de linha, quando estiverem alinhadas na vertical, assim como para as marcações de quebras de linha estáticas.



CodeFolding (Dobradura de código)

Esta cor é usada para destacar a seção de código que será recolhida quando você clicar na seta de recolhimento de código à esquerda de um documento. Para mais informações, consulte a [documentação de recolhimento de código](#).

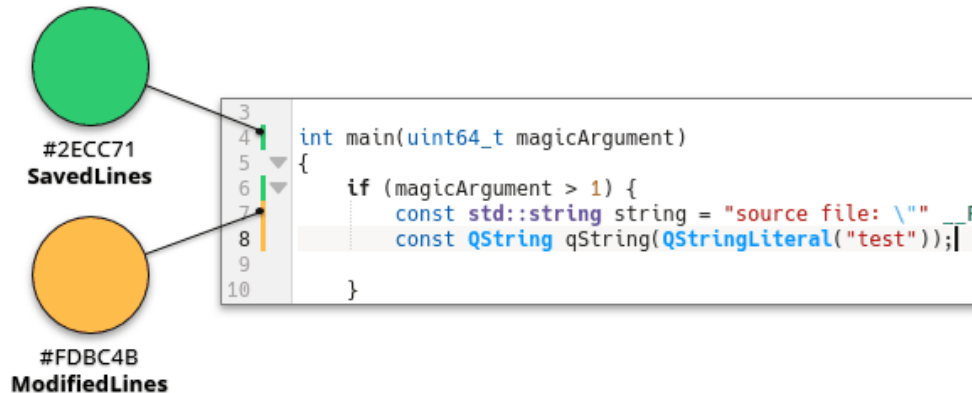


ModifiedLines (Linhas modificadas)

Esta cor é usada para destacar à esquerda de um documento as linhas que foram modificadas, mas ainda não salvas. Para mais informações, consulte Seção 3.9.

SavedLines (Linhas salvas)

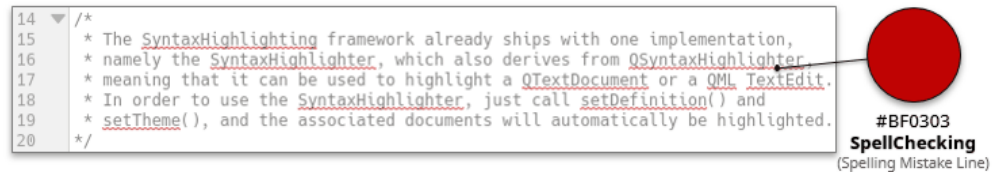
Esta cor é usada para destacar à esquerda de um documento as linhas que foram modificadas nesta sessão e salvas. Para mais informações, consulte Seção 3.9.



Decorações de texto

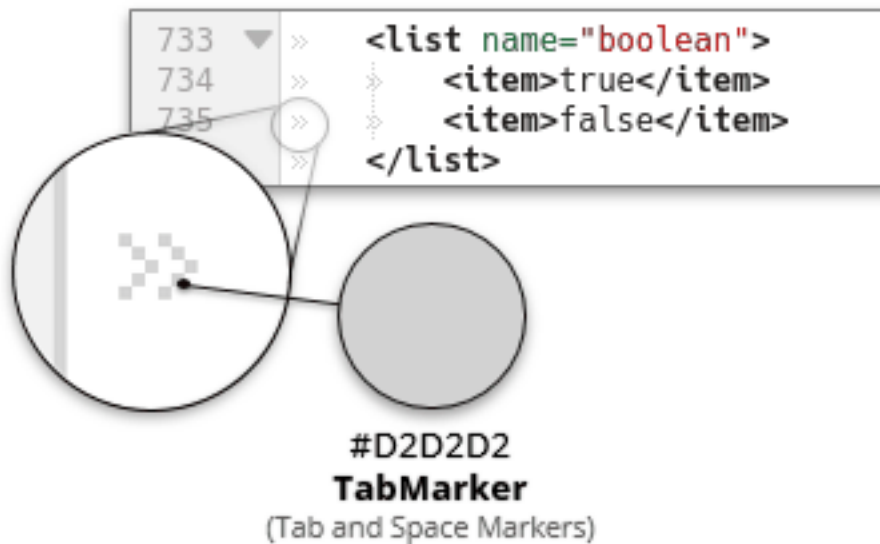
SpellChecking (Linha com erro de ortografia)

Esta cor é usada para indicar erros ortográficos.



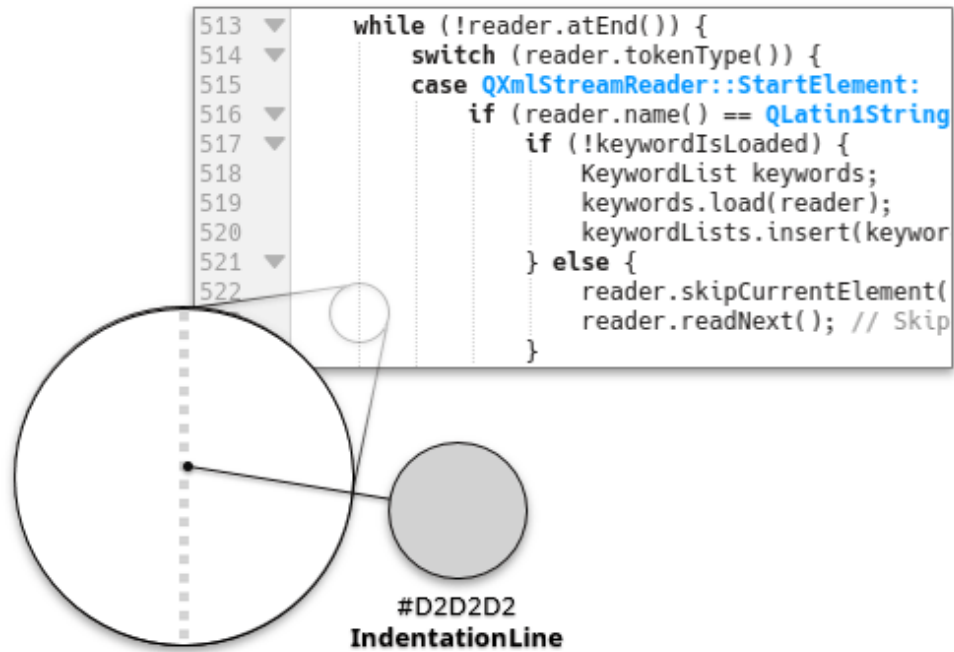
TabMarker (Marcadores de espaço e tabulação)

Esta cor é usada para desenhar os indicadores de espaços em branco, quando eles estiverem ativos.

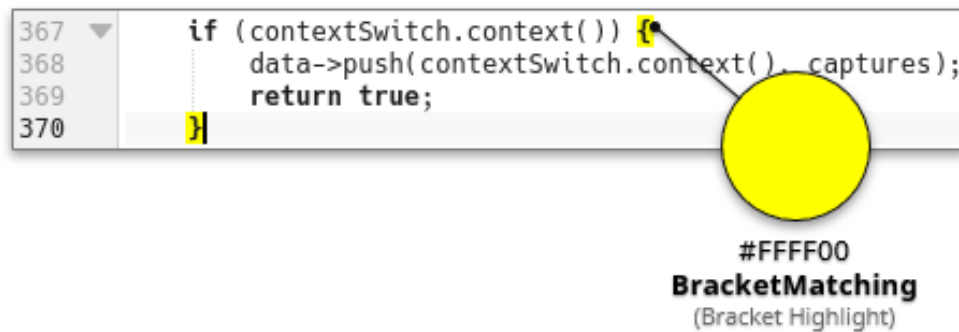


IndentationLine (Recuo da linha)

Esta cor é usada para desenhar uma linha à esquerda de blocos recuados, se [esse recurso](#) estiver ativado.

**BracketMatching (Realce de parênteses)**

Esta cor é usada para desenhar o fundo dos parênteses correspondentes.

**Cores dos marcadores****MarkBookmark (Favorito)**

Esta cor é usada para indicar marcadores. Observe que esta cor tem uma opacidade de 22% (e 33% para a linha atual) em relação ao fundo. Para mais informações, consulte Seção 3.6.

**MarkBreakpointActive (Ponto de parada ativo)**

Esta cor é usada pelo plugin GDB para indicar um ponto de parada ativo. Observe que esta cor tem opacidade em relação ao fundo. Para mais informações, consulte [a documentação do plugin GDB](#).

MarkBreakpointReached (Ponto de parada atingido)

Esta cor é usada pelo plugin GDB para indicar um ponto de parada que você atingiu durante a depuração. Observe que esta cor tem opacidade em relação ao fundo. Para obter mais informações, consulte [a documentação do plugin GDB](#).

MarkBreakpointDisabled (Ponto de parada desativado)

Esta cor é usada pelo plugin GDB para indicar um ponto de parada inativo. Observe que esta cor tem opacidade em relação ao fundo. Para mais informações, consulte [a documentação do plugin GDB](#).

MarkExecution (Execução)

Esta cor é usada pelo plugin GDB na linha que está sendo executada no momento. Observe que esta cor tem opacidade em relação ao fundo. Para mais informações, consulte [a documentação do plugin GDB](#).

MarkWarning (Aviso)

Esta cor é usada pelo plugin de compilação para indicar uma linha que causou um aviso do compilador. Observe que esta cor tem opacidade em relação ao fundo. Para obter mais informações, consulte [a documentação do plugin de compilação](#).

MarkError (Erro)

Esta cor é usada pelo plugin de compilação para indicar uma linha que causou um erro do compilador. Observe que esta cor tem opacidade em relação ao fundo. Para obter mais informações, consulte [a documentação do plugin de compilação](#).

Modelos de texto e trechos

TemplateBackground (Fundo)

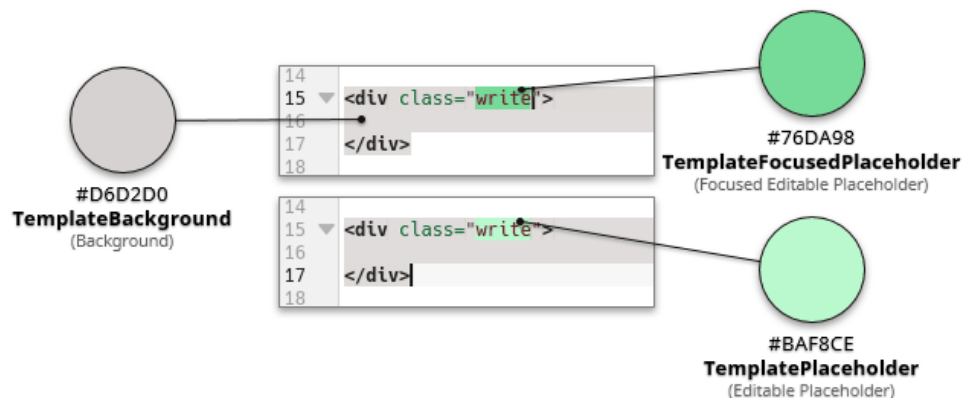
Esta cor é usada pelo plugin de trechos do Kate para marcar o fundo de um trecho de código. Para mais informações, consulte [a documentação de trechos do Kate](#).

TemplatePlaceholder (Espaço reservado editável)

Esta cor é usada pelo plugin de trechos do Kate para marcar um espaço reservado que você pode clicar para editar manualmente. Para mais informações, consulte [a documentação de trechos do Kate](#).

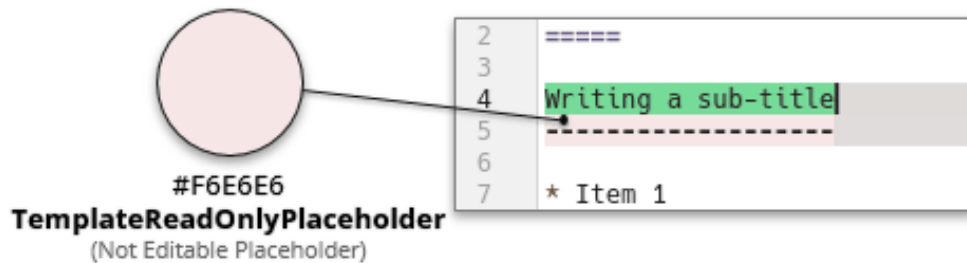
TemplateFocusedPlaceholder (Espaço reservado editável focado)

Esta cor é usada pelo plugin de trechos do Kate para marcar o espaço reservado que você está editando atualmente. Para mais informações, consulte [a documentação de trechos do Kate](#).



TemplateReadOnlyPlaceholder (Espaço reservado não editável)

Esta cor é usada pelo plugin de trechos do Kate para marcar um espaço reservado que não pode ser editado manualmente, como o que é preenchido automaticamente. Para mais informações, consulte [a documentação de trechos do Kate](#).



6.3.4.2 Estilos de texto padrão

Os estilos de texto padrão são herdados pelos estilos de texto realçados, permitindo ao editor apresentar o texto de forma bastante consistente. Por exemplo, o texto do comentário usa o mesmo estilo em quase todos os formatos de texto que o KSyntaxHighlighting consegue realçar.

NOTA

Esses estilos de texto podem ser referenciados nos **estilos padrão** usados nos arquivos de definição XML de [realce de sintaxe](#). Por exemplo, o atributo 'Normal' é equivalente a 'dsNormal' nos arquivos XML e 'DataType' é equivalente a 'dsDataType'. Consulte Seção [6.2.3.3](#) na documentação de realce de sintaxe.

DICA

Certifique-se de escolher cores legíveis com bom contraste, especialmente em combinação com as **Cores do editor**. Consulte Seção [6.3.6.1](#).

No [arquivo JSON](#), a chave **text-styles** respectiva tem como valor um *objeto* onde cada chave corresponde ao nome de um *estilo de texto padrão*, que são equivalentes aos usados nas definições de realce de sintaxe. Aqui, **todas as chaves de estilo de texto disponíveis são obrigatórias**, e estão listadas abaixo.

```
"text-styles": {
  "Normal" : {
    "text-color" : "#1f1c1b",
    "selected-text-color" : "#ffffff",
    "bold" : false,
    "italic" : false,
    "underline" : false,
    "strike-through" : false
  },
  "Keyword" : {
    "text-color" : "#1f1c1b",
    "selected-text-color" : "#ffffff",
    "bold" : true
  },
  "Function" : {
    "text-color" : "#644a9b",
    "selected-text-color" : "#452886"
  }
}
```



```
},
As outras chaves de estilo de texto...
}
```

Cada chave de *estilo de texto padrão* possui um objeto JSON como seu valor, onde valores como *color*, *bold*, *italic*, etc. são especificados. Essas chaves são as seguintes:

text-color: É uma *string* com a cor do texto em código hexadecimal. Este par chave/valor é obrigatório.

selected-text-color: A cor do texto quando ele é selecionado geralmente tem o mesmo valor que 'text-color'. Quando o texto é selecionado, o fundo é definido pelo valor de [TextSelection](#) em [Cores do editor](#), portanto, você deve garantir que o texto tenha bom contraste e seja legível com esse fundo. O valor é uma *string* com um código de cor hexadecimal. Este par chave/valor é obrigatório.

bold: É um *booleano* que determina se o texto está em negrito. Esta chave é opcional, o valor padrão é **false**.

italic: É um *booleano* que determina se o texto está em itálico. Esta chave é opcional, o valor padrão é **false**.

underline: É um *booleano* que determina se o texto está sublinhado. Esta chave é opcional, o valor padrão é **false**.

strike-through: É um *booleano* que determina se o texto está riscado. Esta chave é opcional, o valor padrão é **false**.

background-color: Determina a cor de fundo do texto, usada, por exemplo, em alertas e comentários. O valor é uma *string* com um código de cor hexadecimal. Esta chave é opcional; por padrão, não há cor de fundo.

selected-background-color: Determina a cor de fundo do texto quando ele é selecionado. O valor é uma *string* com um código de cor hexadecimal. Esta chave é opcional; por padrão, não há cor de fundo.

Na [GUI para gerenciar temas de cores do KTextEditor](#), esses atributos podem ser modificados na aba **Estilos de texto padrão**. O nome na lista de estilos usa o estilo configurado para o item, fornecendo uma pré-visualização imediata ao configurar um estilo. Cada estilo permite selecionar atributos comuns, bem como cores de primeiro plano e de fundo. Para remover uma cor de fundo, clique com o botão direito do mouse para usar o menu de contexto.

As chaves de estilo de texto disponíveis são as seguintes; as chaves usadas no [arquivo JSON](#) estão listadas em *negrito*, os nomes usados na [GUI](#) são mostrados entre parênteses se forem diferentes.

Texto normal e código-fonte

Normal: Estilo de texto padrão para texto normal e código-fonte sem realce especial.

Keyword: Estilo de texto para palavras-chave de linguagem integradas.

Function: Estilo de texto para definições de função e chamadas de função.

Variable: Estilo de texto para variáveis, se aplicável. Por exemplo, variáveis `​​` em PHP/Perl normalmente começam com um `$`, portanto, todos os identificadores que seguem o padrão `$foo` são destacados como variáveis.

ControlFlow (Fluxo de controle): Estilo de texto para palavras-chave de fluxo de controle, como *if*, *then*, *else*, *return*, *switch*, *break*, *yield*, *continue*, etc.

Operator: Estilo de texto para operadores, como `+`, `-`, `*`, `/`, `%`, etc.

BuiltIn (Integrado): Estilo de texto para classes de linguagem, funções e objetos integrados.

Extension: Estilo de texto para extensões conhecidas, como classes Qt™, funções/macros em C++ e Python ou boost.

Preprocessor: Estilo de texto para instruções do pré-processador ou definições de macro.

Attribute: Estilo de texto para anotações ou atributos de funções ou objetos, por exemplo, `@override` em Java, ou `__declspec(...)` e `__attribute__((...))` em C++.

Números, tipos e constantes

DataType (Tipo de dados): Estilo de texto para tipos de dados integrados como *int*, *char*, *float*, *void*, *u64*, etc.

DecVal (Decimal/Valor): Estilo de texto para valores decimais.

BaseN (Inteiro de base N): Estilo de texto para números com base diferente de 10.

Float (Ponto flutuante): Estilo de texto para números de ponto flutuante.

Constant: Estilo de texto para constantes de linguagem e constantes definidas pelo usuário, por exemplo, *True*, *False*, *None* em Python ou *nullptr* em C/C++; ou constantes matemáticas como *PI*.

Strings e caracteres

Char (Caractere): Estilo de texto para caracteres únicos como `'x'`.

SpecialChar (Caractere especial): Estilo de texto para caracteres escapados em strings, por exemplo, `'olá\n'`, e outros caracteres com significado especial em strings, como substituições ou operadores regex.

String: Estilo de texto para strings como `'olá mundo'`.

VerbatimString (String Verbatim): Estilo de texto para strings literais ou brutas como `'raw\backlash'` em Perl, CoffeeScript e shells, bem como `r'raw'` em Python, ou como na documentação AQUI.

SpecialString (String especial): Estilo de texto para strings especiais, como expressões regulares em ECMAScript, o modo matemático L^AT_EX, SQL, etc.

Import (Importações, Módulos, Inclusões): Estilo de texto para inclusões, importações, módulos ou pacotes L^AT_EX.

Comentários e documentação

Comment: Estilo de texto para comentários normais.

Documentation: Estilo de texto para comentários que refletem a documentação da API, como `/** comentários doxygen */` ou `"""docstrings"""`.

Annotation: Estilo de texto para anotações em comentários ou comandos de documentação, como `@param` em Doxygen ou JavaDoc.

CommentVar (Variável de comentário): Estilo de texto que se refere aos nomes das variáveis `​` usadas nos comandos acima em um comentário, como `foobar` em `@param foobar`, no Doxygen ou JavaDoc.

RegionMarker (Marcador de região): Estilo de texto para marcadores de região, normalmente definido por `/ BEGIN` e `/ END` em comentários.

Information: Estilo de texto para informações, notas e dicas, como a palavra-chave `@note` no Doxygen.

Warning: Estilo de texto para avisos, como a palavra-chave `@warning` no Doxygen.

Alert: Estilo de texto para palavras especiais em comentários, como `TODO`, `FIXME`, `XXXX` e `WARNING`.

Diversos

Error: Estilo de texto indicando realce de erro e sintaxe incorreta.

Others: Estilo de texto para atributos que não correspondem a nenhum dos outros estilos padrão.

6.3.4.3 Estilos de texto de realce personalizados

Aqui você pode definir estilos de texto para uma definição específica de realce de sintaxe, substituindo o **estilo de texto padrão** descrito na [seção anterior](#).

No [arquivo de tema JSON](#), isso corresponde à chave **custom-styles**, cujo valor é um *objeto* onde cada chave de subesquema corresponde ao **nome de uma definição de realce de sintaxe**. Seu valor é um *objeto* onde cada chave se refere ao **nome dos atributos de estilo** definidos nos [elementos itemData](#) do arquivo de realce de sintaxe XML, e o respectivo valor é um subobjeto com as chaves *text-color*, *selected-text-color*, *bold*, *italic*, *underline*, *strike-through*, *background-color* e *selected-background-color*, definidas [na seção anterior](#). Cada um desses valores é opcional, pois se eles não estiverem presentes, o estilo definido em **text-styles** é considerado.

Por exemplo, neste trecho de código, a definição de realce de sintaxe ‘ISO C++’ tem um estilo de texto especial para os atributos ‘Type Modifiers’ e ‘Standard Classes’. No arquivo XML correspondente ‘isocpp.xml’, o atributo definido ‘Standard Classes’ usa o estilo padrão **BuiltIn** (ou **dsBuiltIn**). Neste atributo, apenas o valor de **text-color** é sobrescrito pela nova cor ‘#6431b3’.

```
"custom-styles": {
  "ISO C++": {
    "Standard Classes": {
      "text-color": "#6431b3"
    },
    "Type Modifiers": {
      "bold": true,
      "selected-text-color": "#009183",
      "text-color": "#00b5cf"
    }
  }
}
```

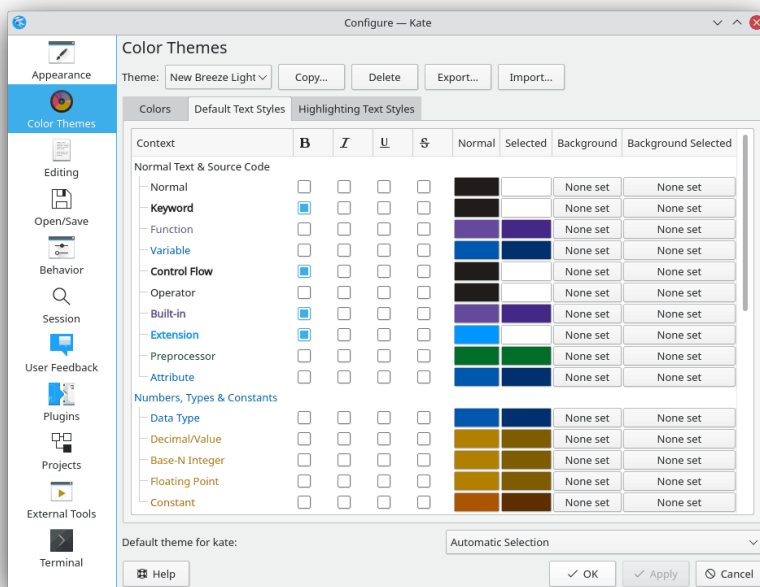
NOTA

- Você deve levar em consideração que esses estilos de texto estão associados aos nomes de atributos definidos nos arquivos de realce de sintaxe XML. Se um arquivo XML for atualizado e alguns atributos forem renomeados ou removidos, o estilo personalizado definido no tema deixará de ser aplicado.
- As definições de realce de sintaxe frequentemente incluem outras definições. Por exemplo, o realçador de sintaxe ‘QML’ inclui o realçador de sintaxe ‘JavaScript’, já que compartilham funcionalidades de realce.

Na [GUI para gerenciar temas do KTextEditor](#), esses atributos podem ser modificados na aba **Estilos de realce de texto**. Por padrão, o editor pré-seleciona o realce do documento atual. Você notará que muitos realces contêm outros realces representados por grupos na lista de estilos. Por exemplo, a maioria dos realces importa o realce ‘Alerta’ e muitos formatos de código-fonte importam o realce ‘Doxygen’.

6.3.5 GUI dos temas de cores

A maneira mais simples de criar e editar temas de cores é através da GUI dentro da [janela ‘Configurar’](#) fornecida pelo **KTextEditor**. Para acessá-la, selecione **Configurações** → **Configurar aplicativo...** na barra de menus do seu editor de texto. Isso abrirá a janela **Configurar**, onde você poderá selecionar **Temas de cores** no painel lateral.



Janela de configuração do Kate com o gerenciamento de temas de cores.

Nesta [janela](#) você pode configurar todas as cores de qualquer tema que você tenha, bem como criar/copiar novos temas, excluí-los, exportá-los para um arquivo `.theme` com [formato JSON](#) ou importá-los de arquivos `.theme` externos. Cada tema possui configurações para cores e estilos de texto.

Os temas integrados não podem ser modificados por padrão. Para fazer isso, você precisa copiá-los e dar-lhes um novo nome.

Para usar um tema permanentemente em seu editor de texto, você deve selecioná-lo na caixa de combinação rotulada como **Tema padrão para aplicativo** na parte inferior da janela e pressionar **Aplicar** ou **OK**. Por padrão, a opção **Seleção automática** está ativa, que escolhe um tema de cores mais apropriado de acordo com o *esquema de cores do KDE Plasma* usado no aplicativo de edição de texto; geralmente, escolhe entre 'Breeze Light' e 'Breeze Dark' se o esquema for claro ou escuro, respectivamente.

DICA

Você pode ajustar o esquema de cores global do KDE no [módulo Cores nas Configurações do sistema](#). Você também pode alterá-lo individualmente em alguns aplicativos, como o Kate ou o KDevelop, no menu **Configurações** → **Esquema de cores**.

6.3.5.1 Criar um novo tema

Para criar um novo tema, primeiro é necessário copiar um existente. Selecione um tema existente que você deseja usar como base, como 'Breeze Light' ou 'Breeze Dark', e clique em **Copiar**. Em seguida, digite um nome para o novo tema.

Se você quiser modificar um tema integrado ou somente leitura, primeiro você deve copiá-lo com um nome diferente.

6.3.5.2 Importar ou exportar arquivos de tema JSON

Você pode exportar um tema selecionado (incluindo os temas integrados) para um [arquivo JSON](#) com a extensão `.theme`, usando o botão **Exportar**. Isso abrirá uma janela para salvar o arquivo.

Para adicionar um tema de cores de um [arquivo JSON](#) externo, basta pressionar o botão **Importar** e selecionar o arquivo `.theme` na janela.

DICA

- Como [mencionado acima](#), os arquivos de tema personalizados pelo usuário são armazenados na pasta `org.kde.syntax-highlighting/themes/`. Ao copiar ou criar um tema, ele aparecerá automaticamente lá. Além disso, importar ou adicionar um tema é equivalente a copiar um arquivo `.theme` externo para esta pasta. O KSyntaxHighlighting detecta automaticamente os arquivos de tema de cores desta pasta.
- Se você deseja publicar um tema criado por você, é essencial verificar o objeto [metadata](#) do [arquivo JSON](#), adicionando a licença respectiva e verificando o número da revisão.

6.3.5.3 Editando temas de cores

6.3.5.3.1 Cores

Aqui, as cores da área de edição de texto são ajustadas. Elas são detalhadas em Seção [6.3.4.1](#).

6.3.5.3.2 Estilos de texto padrão

Os estilos de texto padrão são herdados pelos estilos de texto realçados, permitindo ao editor apresentar o texto de forma bastante consistente. Por exemplo, o texto do comentário usa o mesmo estilo em quase todos os formatos de texto que o KSyntaxHighlighting consegue realçar.

O nome na lista de estilos está usando o estilo configurado para o item, oferecendo-lhe uma visualização imediata ao configurar um estilo.

Cada estilo permite selecionar atributos comuns, bem como cores de primeiro plano e de fundo. Para remover uma cor de fundo, clique com o botão direito do mouse para usar o menu de contexto.

Os atributos desta área estão detalhados em Seção [6.3.4.2](#).

6.3.5.3.3 Estilos de texto de realce

Aqui você poderá editar os estilos de texto usados por uma definição de realce específica. O editor seleciona previamente o realce usado pelo seu documento atual. Para funcionar com um realce diferente, selecione um na lista **Realce**, acima da lista de estilos.

O nome na lista de estilos está usando o estilo configurado para o item, oferecendo-lhe uma visualização imediata ao configurar um estilo.

Cada estilo permite-lhe selecionar os atributos comuns, assim como as cores de fundo e do texto. Para retirar uma cor de fundo, clique com o botão direito para usar o menu de contexto. Além disso, você poderá verificar se um estilo é igual ao pré-definido que foi usado para o item, e configurá-lo como tal, se não estiver.

Você notará que vários realces contêm outros realces, representados por grupos na lista de estilos. Por exemplo, a maioria dos realces implementam o realce Alert (Alerta) e muitos formatos de código importam o realce do Doxygen. A edição das cores destes grupos só afetará os estilos se forem usados no formato de realce editado.

6.3.6 Dicas & Truques

6.3.6.1 Contraste das cores do texto

Um aspecto importante ao trabalhar com temas de cores é escolher um contraste de texto que facilite a leitura, especialmente em combinação com o fundo.

O aplicativo **Kontrast** é um verificador de contraste de cores. Ele informa se as combinações de cores de texto e de fundo são legíveis e acessíveis, sendo uma excelente ferramenta para ajudar na criação de temas de cores.

Você pode baixar o **Kontrast** do [site de aplicativos do KDE](#) ou do [pacote Flatpak no Flathub](#) (somente para GNU/Linux).

O aplicativo **Contrast** do GNOME é semelhante. Você pode baixar o [pacote Flatpak no Flathub](#) (somente para GNU/Linux).

Um script Python no [Repositório de realce de sintaxe](#) permite visualizar todas as cores em um tema, bem como o contraste com os vários planos de fundo configurados.

6.3.6.2 Sugestões de consistência com o realce de sintaxe

O KSyntaxHighlighting inclui [mais de 300 definições de realce de sintaxe](#), portanto, é ideal que você se certifique de que seu novo tema fique bem em todas as definições de realce de sintaxe. Os temas de cores integrados têm as seguintes semelhanças que é recomendável (mas não obrigatório) seguir para obter uma exibição correta de todas as definições de realce de sintaxe:

- Use negrito para os [estilos de texto](#) 'Keyword' e 'ControlFlow'.
- Não use cor de fundo em qualquer [estilo de texto](#), exceto 'Alert' e 'RegionMarker'.

A maioria dos realçadores de sintaxe são projetados para ter uma boa aparência nos temas padrão 'Breeze Light' e 'Breeze Dark', portanto, outra maneira de manter a consistência é usar cores semelhantes nos [estilos de texto](#), como *verde* para 'Preprocessor' e 'Others', *azul* para 'DataType' e 'Attribute', ou *roxo* para 'Function'.

Observe que essas recomendações não são obrigatórias ao criar e publicar um tema.

6.4 Scripts com JavaScript

O componente de edição do KatePart é facilmente extensível, através da criação de scripts. A linguagem de programação é o ECMAScript (popularmente conhecido como JavaScript). O KatePart suporta dois tipos de scripts: os de recuo e os de linha de comando. Uma funcionalidade semelhante aos scripts de linha de comando também é fornecida no [plugin de trechos](#).

6.4.1 Scripts de recuo

Scripts de recuo - também conhecidos por recuadores - formatam automaticamente o código-fonte enquanto você digita o texto. Por exemplo, depois de pressionar a tecla Return, o nível de recuo costuma aumentar.

As seções a seguir descrevem, passo-a-passo, como criar o esqueleto de um módulo de recuo simples. Como primeiro passo, crie um novo arquivo *.js, chamado de, por exemplo, javascript.js na pasta pessoal local \$XDG_DATA_HOME /katepart5/script/indentation. Nesse sentido, a variável de ambiente XDG_DATA_HOME normalmente se expande para qualquer ~/.local ou ~/.local/share.

No Windows® estes arquivos estão localizados em %USERPROFILE%\AppData\Local\katepart5\script\indentation. %USERPROFILE% que normalmente se expandem para C:\\Users\\usuário.

6.4.1.1 O cabeçalho do script de recuo

O cabeçalho do arquivo `javascript.js` está incorporado como JSON no início do documento no seguinte formato:

```
var katescript = {
  "name": "JavaScript",
  "author": "Nome de Exemplo <nome.exemplo@endereco.org>",
  "license": "BSD License",
  "revision": 1,
  "kate-version": "5.1",
  "required-syntax-style": "javascript",
  "indent-languages": ["javascript"],
  "priority": 0,
}; // kate-script-header, deve estar no início do arquivo, sem comentários
```

Cada item será agora explicado em detalhes:

- **name** [obrigatório]: Este é o nome do modo de recuo que aparece no menu **Ferramentas** → **Recuar** e na janela de configuração.
- **author** [opcional]: O nome e a informação de contato do autor.
- **license** [opcional]: Forma curta da licença, como por exemplo, BSD License ou LGPLv3.
- **revision** [obrigatório]: A versão do programa. Este número deverá ser aumentado sempre que o programa for modificado.
- **kate-version** [obrigatório]: A versão mínima do KatePart necessária.
- **required-syntax-style** [opcional]: O estilo do sintaxe necessário, que corresponde ao `style` indicado no sintaxe dos arquivos realçados. Isto é importante para recuadores que se baseiam em informações de realce no documento. Se uma determinada realce de sintaxe for especificada, o recuador estará disponível somente quando a realce de sintaxe apropriada estiver ativa. Isto evita um 'comportamento inesperado' causado pelo uso do recuador sem o esquema de realce esperado. Por exemplo, o recuado do Ruby faz uso dos arquivos `ruby.js` e `ruby.xml`.
- **indent-languages** [opcional]: Uma lista JSON de estilos de sintaxe que este módulo consegue recuar corretamente, por exemplo: `["c++", "java"]`.
- **priority** [opcional]: Se existirem vários módulos de recuo adequados para um determinado arquivo de realce, este campo decide qual deverá ser usado por padrão.

6.4.1.2 O código-fonte do recuador

Tendo definido o cabeçalho, esta seção explica como funciona a programação de recuadores em si. O esqueleto básico do código parece-se com o seguinte:

```
// Necessário para as bibliotecas js do katepart, por exemplo range.js se ←
você usar Range
require ("range.js");

caracteresAtivacao = "{}/:;";
function indent(linha, larguraRecuo, caractere)
{
  // invocado para cada linha nova (caractere == '\n') e todos os ←
  caracteres indicados na
  // variável global 'caracteresAtivacao'. Ao invocar o ←
  FerramentasFormatar recuo
```

```
// a variável 'caractere' está em branco, isto é, caractere == ''.
//
// ver também: API de Programação
return -2;
}
```

A função `indent()` tem três parâmetros:

- `linha`: a linha que tem de ser recuada
- `larguraRecuo`: o número de espaços correspondentes a cada recuo
- `caractere`: ou um caractere de mudança de linha (`ch == '\n'`), algum dos caracteres de ativação indicados em `caracteresAtivacao` ou vazio, caso o usuário tenha invocado a ação **Feramentas** → **Alinhar**.

O valor devolvido pela função `indent()` define como a linha será recuada. Se o valor devolvido for um número inteiro, é interpretado da seguinte forma:

- valor devolvido `-2`: não fazer nada
- valor devolvido `-1`: mantém o recuo (procura pela linha não-vazia anterior)
- valor devolvido `0`: números `>= 0` definem a largura do recuo em espaços

Em alternativa, poderá ser devolvida uma lista com dois elementos:

- `return [ recuo, alinhamento ];`

Nesse caso, o primeiro elemento é a profundidade de recuo, tendo o mesmo significado que os valores especiais. Contudo, o segundo elemento é um valor absoluto que representa uma coluna para o 'alinhamento'. Se este valor for maior que o valor do recuo, a diferença representa um número de espaços a adicionar após o recuo do primeiro parâmetro. Caso contrário, o segundo número será ignorado. A utilização de tabulações e espaços nos recuos é normalmente referida como sendo um 'modo misto'.

Considere o seguinte exemplo: Considerando que são usadas as tabulações para recuar, e que o tamanho da tabulação é de 4. Aqui, o `<tab>` representa uma tabulação e o `' '` representa um espaço:

```
1: <tab><tab>xpto("olá",
2: <tab><tab>....."mundo");
```

Ao recuar a linha 2, a função `indent()` devolve `[8, 15]`. Em função disso, são introduzidas duas tabulações para recuar até à coluna 8, e são introduzidos 7 espaços para alinhar o segundo parâmetro com o primeiro, de modo a permanecer alinhado, caso o arquivo seja visto com outro tamanho de tabulação.

Uma instalação padrão do KDE fornece o KatePart com vários módulos de recuo. O código-fonte correspondente em JavaScript poderá ser encontrado em `$XDG_DATA_DIRS /katepart5/script/indentation`.

No Windows® estes arquivos estão localizados em `%USERPROFILE%\AppData\Local\katepart5\script\indentation`. `%USERPROFILE%` que normalmente se expandem para `C:\\Users\\usuário`.

A criação de um módulo de recuo necessita do recarregamento dos scripts, de modo a ver se as alterações se comportam de forma adequada. Em vez de reiniciar o aplicativo, basta ir para a linha de comando e executar o comando **reload-scripts**.

Se você criar scripts úteis, considere a hipótese de contribuir com eles para o Projeto KatePart enviando uma solicitação de mesclagem para [nosso projeto no GitLab](#).

6.4.2 Scripts de Linha de Comando

NOTA

Uma funcionalidade semelhante a scripts de linha de comando também é fornecida no [plugin de trechos](#). Este plugin pode fornecer um ponto de partida mais fácil, especialmente para pequenos scripts personalizados.

Como é difícil satisfazer as necessidades de todos, o KatePart tem suporte a algumas ferramentas auxiliares para manipular rapidamente o texto, usando para isso a [linha de comando incorporada](#). Por exemplo, o comando **sort** está implementado como um script. Esta seção explica como criar arquivos *.js que estendem o KatePart com algumas funções auxiliares arbitrárias.

Os scripts de linha de comando estão localizados na mesma pasta que os scripts de recuo. Assim, como primeiro passo, crie um arquivo *.js novo, chamado `utilitários.js` na pasta pessoal local, em `$XDG_DATA_HOME /katepart5/script/commands`. Nesse sentido, a variável de ambiente `XDG_DATA_HOME` normalmente se expande para `~/local` ou `~/local/share`.

No Windows, esses arquivos estão localizados em `%USERPROFILE%\AppData\Local\katepart5\script\commands`. `%USERPROFILE%` geralmente se expande para `C:\\Users\\usuário`.

6.4.2.1 O Cabeçalho do Script de Linha de Comando

O cabeçalho de cada script de linha de comando é está incorporado como JSON no início do script no seguinte formato:

```
var katescript = {
  "author": "Nome de exemplo <nome.exemplo@endereco.org>",
  "license": "LGPLv2+",
  "revision": 1,
  "kate-version": "5.1",
  "functions": ["ordenar", "descerLinhas"],
  "actions": [
    {
      "function": "ordenar",
      "name": "Ordena o texto selecionado",
      "category": "Edição",
      "interactive": "false"
    },
    {
      "function": "descerLinhas",
      "name": "Desce as linhas",
      "category": "Edição",
      "shortcut": "Ctrl+Shift+Down",
      "interactive": "false"
    }
  ]
}; // kate-script-header, deve estar no início do arquivo, sem comentários
```

Cada item será agora explicado em detalhes:

- **author** [opcional]: O nome e a informação de contato do autor.
- **license** [opcional]: Forma curta da licença, como por exemplo, BSD License ou LGPLv2.
- **revision** [obrigatório]: A versão do programa. Este número deverá ser aumentado sempre que o programa for modificado.
- **kate-version** [obrigatório]: A versão mínima do KatePart necessária.

- **functions** [obrigatório]: Uma lista JSON dos comandos do script.
- **actions** [opcional]: Array em JSON de objetos JSON que definem as ações que aparecem no menu do aplicativo. São oferecidas informações detalhadas na seção [Associar combinações de teclas](#).

Uma vez que o valor de **functions** é uma lista em JSON, um único script é capaz de conter um número arbitrário de comandos da linha de comando. Cada função está disponível através da [linha de comando incorporada](#) do KatePart.

6.4.2.2 O Código-Fonte do Programa

Todas as funções indicadas no cabeçalho terão de estar implementadas no programa. Por exemplo, o arquivo de programa do exemplo acima tem que implementar as duas funções **ordenar** e **descerLinhas**. Todas as funções têm a seguinte sintaxe:

```
// bibliotecas necessárias do katepart js, por exemplo range.js se você ←
  usar Range
function <nome>(arg1, arg2, ...)
{
    // ... implementação, ver também: API de Programação
}
```

Os argumentos na linha de comando são passados à função como *arg1*, *arg2*, etc. Para poder indicar a documentação de cada comando, basta implementar a função 'help', como demonstrado a seguir:

```
function help(cmd)
{
    if (cmd == "sort") {
        return il8n("Ordena o texto selecionado.");
    } else if (cmd == "...") {
        // ...
    }
}
```

Executar **help sort** na linha de comando, irá invocar esta função de ajuda com o argumento *cmd* igual ao comando indicado, como isto é, *cmd == "sort"*. O KatePart irá assim apresentar o texto devolvido como documentação ao usuário. Certifique-se de [traduzir os textos](#).

Criar um script para a linha de comando requer o recarregamento dos scripts, de modo a ver se as alterações funcionam apropriadamente. Em vez de reiniciar o aplicativo, basta mudar para a linha de comando e executar o comando **reload-scripts**.

6.4.2.2.1 Associar combinações de teclas

Para tornar os scripts acessíveis no menu do aplicativo e atribuir teclas de atalho, o script precisa fornecer um cabeçalho do script apropriado. No exemplo acima, ambas as funções **ordenar** e **descerLinhas** aparecem no menu, graças ao seguinte componente no cabeçalho do script:

```
var katescript = {
    ...
    "actions": [
        {
            "function": "ordenar",
            "name": "Ordena o texto selecionado",
            "icon": "",
            "category": "Edição",
            "interactive": "false"
        }
    ]
}
```

```

    },
    {
      "function": "descerLinhas",
      "name": "Desce as linhas",
      "icon": "",
      "category": "Edição",
      "shortcut": "Ctrl+Shift+Down",
      "interactive": "false"
    }
  ]
};

```

Os campos de uma ação são os seguintes:

- `function` [obrigatório]: A função que deverá aparecer no menu **Ferramentas** → **Scripts**.
- `name` [obrigatório]: O texto que aparece no menu do script.
- `icon` [opcional]: O ícone aparece após ao texto no menu. Todos os nomes de ícones do KDE poderão ser usados aqui.
- `category` [opcional]: Se for indicada uma categoria, o programa aparece num submenu.
- `shortcut` [opcional]: A combinação de teclas indicada aqui é o atalho de teclado padrão. Por exemplo: Ctrl+Alt+T. Veja a [documentação do Qt](#) para mais detalhes.
- `interactive` [opcional]: Se o script precisa de interação por parte do usuário na linha de comando, configure este valor como `true` (verdadeiro).

Se você criar scripts úteis, considere a hipótese de contribuir com eles para o Projeto KatePart enviando uma solicitação de mesclagem para [nosso projeto no GitLab](#).

6.4.3 API de Programação

A API de criação de scripts aqui apresentada está disponível em todos os scripts, isto é, os scripts de recuo e os comandos do terminal. As classes `Cursor` e o `Range` são fornecidas por arquivos de bibliotecas em `$XDG_DATA_DIRS /katepart5/libraries`. Elas serão necessárias, caso queira usar algumas das funções `Document` ou `View` no seu script. Inclua a biblioteca necessária usando:

```

// bibliotecas necessárias do katepart js, por exemplo range.js se você ↔
  usar Range
require ("range.js");

```

Para ampliar a API padrão de scripts com suas funções e protótipos próprios, basta criar um arquivo novo na pasta de configuração local do KDE, em `$XDG_DATA_HOME /katepart5/libraries` e incluí-lo no seu script usando:

```

require ("myscriptnamehere.js");

```

No Windows, esses arquivos estão localizados em `%USERPROFILE%\AppData\Local\katepart5\libraries`. `%USERPROFILE%` geralmente se expande para `C:\\Users\\usuário`.

Para estender os protótipos existentes, como o `Cursor` ou o `Range`, a forma recomendada é *não* modificar os arquivos `*.js` globais. Em vez disso, altere o protótipo do `cursor.js` em JavaScript, após o `cursor.js` que é incluído no seu script através do `require`.

6.4.3.1 Cursores e Intervalos

Como o KatePart é um editor de texto, sempre que possível, toda a API de criação de scripts é baseada em cursores e intervalos. Um cursor é uma simples dupla (*linha*, *coluna*) que representa uma posição de texto no documento. Um Range (Intervalo) corresponde a uma área de texto coberta desde uma posição inicial do cursor até outra posição de fim. A API é explicada em detalhes nas seções a seguir.

6.4.3.1.1 O Protótipo do Cursor

Cursor();

Construtor. Devolve um Cursor na posição (0, 0).

Exemplo: `var cursor = new Cursor();`

Cursor(int linha, int coluna);

Construtor. Devolve um Cursor na posição (*linha*, *coluna*).

Exemplo: `var cursor = new Cursor(3, 42);`

Cursor(Cursor outro);

Construtor de cópia. Devolve uma cópia do cursor *outro*.

Exemplo: `var copia = new Cursor(outro);`

Cursor Cursor.clone();

Devolve uma cópia do cursor.

Exemplo: `var clone = cursor.clone();`

Cursor.setPosition(int linha, int coluna);

Configura a posição do cursor em *linha* e *coluna*.

Desde: KDE 4.11

bool Cursor.isValid();

Verifica se o cursor é válido. O cursor é inválido no caso em que a linha e/ou coluna sejam iguais a -1.

Exemplo: `var valido = cursor.isValid();`

Cursor Cursor.invalid();

Devolve um novo cursor inválido, localizado em (-1, -1).

Exemplo: `var cursorInvalido = cursor.invalid();`

int Cursor.compareTo(Cursor outro);

Compara este cursor com o cursor *outro*. Devolve

- -1, se este cursor for localizado antes do cursor *outro*,
- 0, se ambos os cursores forem iguais e
- +1, se este cursor se localizar após o cursor *outro*.

bool Cursor.equals(Cursor outro);

Devolve `true` (verdadeiro), se este o cursor e o *outro* forem iguais, caso contrário devolve `false` (falso).

String Cursor.toString();

Devolve o cursor como um texto no formato 'Cursor(*linha*, *coluna*)'.

6.4.3.1.2 O Protótipo do Intervalo

Range () ;

Construtor. A invocação de `new Range()` devolve um intervalo Range de (0, 0) - (0, 0).

Range (Cursor inicio, Cursor fim) ;

Construtor. A invocação de `new Range(inicio, fim)` devolve o intervalo (*inicio*, *fim*).

Range (int linhaInicial, int colunaInicial, int linhaFinal, int colunaFinal) ;

Construtor. A invocação de `new Range(linhaInicial, colunaInicial, linhaFinal, colunaFinal)` devolve um intervalo Range de (*linhaInicial*, *colunaInicial*) até (*linhaFinal*, *colunaFinal*).

Range (Range outro) ;

Construtor de cópia. Devolve uma cópia do intervalo Range *outro*.

Range Range.clone () ;

Devolve uma cópia do intervalo.

Exemplo: `var clone = intervalo.clone();`

bool Range.isEmpty () ;

Devolve `true` (verdadeiro), se o cursor de início e de fim forem iguais.

Exemplo: `var vazio = intervalo.isEmpty();`

Desde: KDE 4.11

bool Range.isValid () ;

Devolve `true` (verdadeiro), se tanto o cursor de início como o de fim forem válidos, caso contrário devolve `false` (falso).

Exemplo: `var valido = intervalo.isValid();`

Range Range.invalid () ;

Devolve o intervalo Range de (-1, -1) até (-1, -1).

bool Range.contains (Cursor cursor) ;

Devolve `true` (verdadeiro) se este intervalo contiver a posição do cursor, caso contrário, devolve `false` (falso).

bool Range.contains (Range outro) ;

Devolve `true` (verdadeiro) se este intervalo contiver o intervalo Range *outro*, caso contrário, devolve `false` (falso).

bool Range.containsColumn (int coluna) ;

Devolve `true` (verdadeiro) se a *coluna* estiver no intervalo semiaberto [*inicio.coluna*, *fim.coluna*), caso contrário, devolve `false` (falso).

bool Range.containsLine (int linha) ;

Devolve `true` (verdadeiro) se a *linha* estiver no intervalo semiaberto [*inicio.linha*, *fim.linha*), caso contrário, devolve `false` (falso).

bool Range.overlaps (Range outro) ;

Devolve `true` (verdadeiro) se este intervalo e o intervalo *outro* compartilharem uma região em comum, caso contrário devolve `false` (falso).

bool Range.overlapsLine (int linha) ;

Devolve `true` (verdadeiro) se a *linha* estiver no intervalo [*inicio.linha*, *fim.linha*), caso contrário devolve `false` (falso).

bool Range.overlapsColumn(int coluna);

Devolve `true` (verdadeiro) se a `coluna` estiver no intervalo `[início.coluna, fim.coluna]`, caso contrário devolve `false` (falso).

bool Range.onSingleLine();

Devolve `true` (verdadeiro) se o intervalo inicia e termina na mesma linha, isto é, se `Range.start.line == Range.end.line`.

Desde: KDE 4.9

bool Range.equals(Range outro);

Devolve `true` (verdadeiro) se este intervalo e o intervalo `outro` forem iguais, caso contrário devolve `false` (falso).

String Range.toString();

Devolve o intervalo como uma string no formato `'Range(Cursor(linha, coluna), Cursor(linha, coluna))'`.

6.4.3.2 Funções globais

Esta seção apresenta todas as funções globais.

6.4.3.2.1 Arquivos de leitura e inclusão

String read(String arquivo);

Irá procurar pelo `arquivo` indicado em relação à pasta `katepart5/script/files` e irá devolver o seu conteúdo como texto.

void require(String Arquivo);

Irá procurar pelo `arquivo` indicado em relação à pasta `katepart5/script/libraries` e avaliá-lo. O `require` está protegido internamente contra inclusões múltiplas do mesmo `arquivos`.

Desde: KDE 4.10

6.4.3.2.2 Depuração

void debug(String texto);

Imprime o `texto` no `stdout`, mais precisamente no console que lança a aplicação.

6.4.3.2.3 Tradução

Para um suporte regional completo, existem diversas funções para traduzir os textos nos programas, nomeadamente o `i18n`, o `i18nc`, o `i18np` e o `i18ncp`. Estas funções comportam-se exatamente como as [funções de tradução do KDE](#).

As funções de tradução convertem os textos indicados ao sistema de traduções do KDE para o idioma usado no aplicativo. Os textos nos scripts que são desenvolvidos no âmbito do código oficial do KatePart são automaticamente extraídos e preparados para tradução. Em outras palavras, como desenvolvedor do KatePart, você não precisa que se preocupar com a extração e tradução. Contudo, a tradução só funciona dentro da infraestrutura do KDE, isto é, novos textos em scripts de terceiros, desenvolvidos fora do KDE, não são traduzidos. Dessa forma, considere contribuir com seus scripts para o Kate, para que seja possível obter uma tradução adequada.

```
void i18n(String texto, arg1, ...);
```

Traduz o *texto* na língua usada pela aplicação. Os argumentos *arg1*, ..., são opcionais e usados para substituir os itens %1, %2, etc.

```
void i18nc(String contexto, String texto, arg1, ...);
```

Traduz o *texto* na língua usada pela aplicação. Além disso, o texto do *contexto* fica visível aos tradutores para que possam dar uma tradução mais adequada. Os argumentos *arg1*, ..., são opcionais e são usados para substituir os itens de substituição %1, %2, etc.

```
void i18np(String singular, String plural, int número, arg1, ...);
```

Traduz tanto o texto *singular* como o *plural* para a língua usada pela aplicação, dependendo do *número* indicado. Os argumentos *arg1*, ..., são opcionais e usados para substituir os itens de substituição %1, %2, etc.

```
void i18ncp(String contexto, String singular, String plural, int número, arg1, ...);
```

Traduz tanto o texto *singular* como o *plural* para a língua usada pela aplicação, dependendo do *número* indicado. Adicionalmente, o texto do *contexto* é visível para os tradutores, para que possam fornecer uma tradução mais adequada. Os argumentos *arg1*, ..., são opcionais e são usados para substituir os itens %1, %2, etc.

6.4.3.3 A API do View

Sempre que um programa está sendo executado, existe uma variável global 'view' que representa a área de edição ativa no momento. Segue-se uma lista com todas as funções disponíveis para o View.

```
void view.copy();
```

Copia a seleção se houver uma, caso contrário, a linha atual se a opção [] **Copiar/Recortar a linha atual se nenhuma seleção** estiver definida.

Desde: KDE Frameworks 5.79

```
void view.cut();
```

Recorta a seleção se houver uma, caso contrário, a linha atual se a opção [] **Copiar/Recortar a linha atual se nenhuma seleção** estiver definida.

Desde: KDE Frameworks 5.79

```
void view.paste();
```

Cola o conteúdo da área de transferência.

Desde: KDE Frameworks 5.79

```
Cursor view.cursorPosition();
```

Devolve a posição atual do cursor na janela.

```
void view.setCursorPosition(int linha, int coluna); void
```

```
view.setCursorPosition(Cursor cursor);
```

Altera a posição atual do cursor para uma (linha, coluna) qualquer ou para o cursor indicado.

```
Cursor view.virtualCursorPosition();
```

Devolve a posição virtual do cursor com cada tabulação a corresponder ao número indicado de espaços, dependente da largura de tabulação atual.

```
void view.setVirtualCursorPosition(int linha, int coluna); void
```

```
view.setVirtualCursorPosition(Cursor cursor);
```

Altera a posição atual do cursor virtual para uma (linha, coluna) qualquer ou para o cursor indicado.

String view.selectedText();

Devolve o texto selecionado. Se não tiver nenhum texto selecionado, o texto devolvido vem vazio.

bool view.hasSelection();

Devolve `true` (verdadeiro) se a janela contiver algum texto selecionado, caso contrário, devolve `false` (falso).

Range view.selection();

Devolve o intervalo de texto selecionado. O intervalo devolvido é inválido, caso não esteja selecionado nenhum texto.

void view.setSelection(Range intervalo);

Altera o texto selecionado para o intervalo indicado.

void view.removeSelectedText();

Remove o texto selecionado. Se a janela não tiver nenhuma seleção feita, isto não faz nada.

void view.selectAll();

Seleciona todo o texto no documento.

void view.clearSelection();

Limpa a seleção de texto atual, sem remover o texto.

void view.setBlockSelection(bool on);

Liga/desliga o modo de seleção de blocos.

bool view.blockSelection();

Devolve `true` (verdadeiro) se o modo de seleção de bloco estiver ligado, caso contrário, devolve `false` (falso).

void view.align(Range intervalo);

Ajusta corretamente o recuo das linhas dentro do *intervalo* de acordo com as configurações de recuo atuais.

void view.alignOn(Range intervalo, String padrão = "");

Alinha as linhas no *intervalo* na coluna especificada pela expressão regula *padrão*. Com um *padrão* vazio especificado, o alinhamento será feito no primeiro caractere não vazio. Se o padrão contiver uma captura, o recuo será feito na correspondência capturada.

Exemplos:

`view.alignOn(document.documentRange(), '-');` irá inserir espaços antes do primeiro - de cada linha para alinhá-las todas na mesma coluna.

`view.alignOn(document.documentRange(), ':\s+(.)');` irá inserir espaços antes do primeiro caractere não em branco que ocorrer após dois pontos para alinhá-los todos na mesma coluna.

object view.executeCommand(String comando, String args, Range intervalo);

Executa o [comando de linha de comando](#) *comando* com os argumentos opcionais *args* e o intervalo opcional *intervalo*. O objeto retornado *object* possui uma propriedade booleana *object.ok* que indica se a execução do comando *comando* foi bem-sucedida. Em caso de erro, a string *object.status* contém uma mensagem de erro.

Desde: KDE Frameworks 5.50

Range view.searchText(Range intervalo, String padrão, bool backwards = false);

Busca a primeira ocorrência do *padrão* em *intervalo* e retorna o intervalo correspondente. A busca é realizada de trás para frente se o parâmetro booleano opcional *backwards* estiver definido como `true`.

O intervalo retornado é inválido (consulte `Range.isValid()`) se *padrão* não for encontrado no *intervalo*.

Desde: KDE Frameworks 5.97

6.4.3.4 API do Document

Sempre que um script estiver em execução, existe um objeto global (variável) 'document' representando o documento ativo atual. A seguir é apresentada uma lista de todas as funções disponíveis para o Document.

String document.fileName();

Devolve o nome do arquivo do documento, ou então um texto vazio para as janelas de texto ainda por salvar.

String document.url();

Devolve o URL completo do documento, ou então um texto vazio para as janelas de texto ainda por salvar.

String document.mimeType();

Devolve o tipo MIME do documento, ou então o tipo MIME `application/octet-stream` se não for encontrado qualquer tipo MIME apropriado.

String document.encoding();

Devolve a codificação usada atualmente para salvar o arquivo.

String document.highlightingMode();

Devolve o modo de realce global usado para todo o documento.

String document.highlightingModeAt(Cursor pos);

Devolve o modo de realce usado na posição do cursor indicada do documento.

Array document.embeddedHighlightingModes();

Devolve uma lista com os modos de realce incorporados neste documento.

bool document.isModified();

Devolve `true` (verdadeiro) se o documento tiver alguma alteração por salvar, caso contrário devolve `false`.

String document.text();

Devolve o conteúdo inteiro do documento numa única sequência de texto. As mudanças de linha estão marcadas com o caractere de mudança de linha '\n'.

String document.text(int daLinha, int daColuna, int paraLinha, int paraColuna); String document.text(Cursor de, Cursor para); String document.text(Range intervalo);

Devolve o texto no intervalo indicado. Recomenda-se que use a versão baseada nos cursores e nos intervalos, de modo a ter uma melhor visibilidade do código-fonte.

String document.line(int linha);

Devolve a linha de texto indicada como uma sequência de texto. O texto devolvido fica em branco, caso a linha pedida esteja fora do intervalo.

String document.wordAt(int linha, int coluna); String document.wordAt(Cursor cursor);

Devolve a palavra na posição do cursor indicada.

Range document.wordRangeAt(int linha, int coluna); Range document.wordRangeAt(Cursor cursor);

Devolve o intervalo da palavra na posição indicada do cursor. O intervalo devolvido é inválido (ver `Range.isValid()`), caso a posição do texto esteja após o fim de uma linha. Se não existir nenhuma palavra no cursor indicado, é devolvido um intervalo vazio.

Desde: KDE 4.9

String document.charAt(int linha, int coluna); String document.charAt(Cursor cursor);

Devolve o caractere na posição do cursor indicada.

String document.firstChar(int linha);

Devolve o primeiro caractere na *linha* que não seja um espaço em branco. O primeiro caractere encontra-se na coluna 0. Se a linha estiver em branco ou conter apenas espaços, a string devolvida será vazia.

String document.lastChar(int linha);

Devolve o último caractere da *linha* indicada que não seja um espaço em branco. O primeiro caractere encontra-se na coluna 0. Se a linha estiver em branco ou só tiver espaços em branco, o texto devolvido vem vazio.

bool document.isSpace(int linha, int coluna); bool document.isSpace(Cursor cursor);

Devolve *true* (verdadeiro), se o caractere na posição indicada do cursor for um espaço em branco, caso contrário, devolve *false* (falso).

bool document.matchesAt(int linha, int coluna, String texto); bool document.matchesAt(Cursor cursor, String texto);

Devolve *true* (verdadeiro) se o *texto* indicado corresponder à posição indicada do cursor, caso contrário, devolve *false* (falso).

bool document.startsWith(int linha, String texto, bool ignorarEspacos);

Devolve *true* (verdadeiro) se a linha começar por *texto*, caso contrário, devolve *false* (falso). O argumento *ignorarEspacos* controla se os espaços envolventes são ignorados ou não.

bool document.endsWith(int linha, String texto, bool ignorarEspacos);

Devolve *true* (verdadeiro), caso a linha termine em *texto*, caso contrário devolve *false* (falso). O argumento *ignorarEspacos* controla se os espaços envolventes são ignorados.

bool document.setText(String texto);

Altera o texto do documento por inteiro.

bool document.clear();

Limpa o texto no documento por inteiro.

bool document.truncate(int linha, int coluna); bool document.truncate(Cursor cursor);

Trunca a linha indicada, na coluna ou posição do cursor indicadas. Devolve *true* (verdadeiro) em caso de sucesso ou *false* (falso) se a linha não estiver dentro do intervalo do documento.

bool document.insertText(int linha, int coluna, String texto); bool document.insertText(Cursor cursor, String texto);

Insere o *texto* na posição do cursor indicada. Devolve *true*, em caso de sucesso, ou *false* (falso), se o documento estiver apenas para leitura.

bool document.removeText(int daLinha, int daColuna, int paraLinha, int paraColuna); bool document.removeText(Cursor de, Cursor para); bool document.removeText(Range intervalo);

Remove o texto no intervalo indicado. Devolve *true* (verdadeiro), em caso de sucesso, ou *false* (falso), se o documento estiver no modo apenas para leitura.

bool document.insertLine(int linha, String texto);

Insere o texto na linha indicada. Devolve *true* (verdadeiro), em caso de sucesso, ou *false* (falso), caso o documento esteja apenas para leitura ou se a linha não estiver no intervalo do documento.

bool document.removeLine(int linha);

Remove a linha de texto indicada. Devolve `true` (verdadeiro), em caso de sucesso, ou `false` (falso), caso o documento esteja no modo apenas para leitura ou se a linha não estiver no intervalo do documento.

bool document.wrapLine(int linha, int coluna); bool document.wrapLine(Cursor cursor);

Reparte a linha na posição indicada pelo cursor. Devolve `true` (verdadeiro) em caso de sucesso ou `false` (falso) se, por exemplo, a linha < 0 .

Desde: KDE 4.9

void document.joinLines(int linhaInicial, int linhaFinal);

Junta as linhas de *linhaInicial* até *linhaFinal*. Duas linhas de texto sucessivas estão sempre separadas por um espaço em branco.

int document.lines();

Devolve o número de linhas do documento.

bool document.isLineModified(int linha);

Devolve `true` (verdadeiro), se a *linha* contém dados que não foram salvos.

Desde: KDE 5.0

bool document.isLineSaved(int linha);

Devolve `true` (verdadeiro), se a *linha* foi alterada, mas o documento foi salvo. Assim, a linha não contém dados não salvos.

Desde: KDE 5.0

bool document.isLineTouched(int linha);

Devolve `true` (verdadeiro), se a *linha* contém dados que não foram salvos ou foi anteriormente alterado.

Desde: KDE 5.0

bool document.findTouchedLine(int linhaInicial, bool abaixo);

Procura pela próxima linha tocada que comece na *linha*. A pesquisa é efetuada tanto para cima como para baixo, dependendo da direção de pesquisa indicada em *abaixo*.

Desde: KDE 5.0

int document.length();

Devolve o número de caracteres do documento.

int document.lineLength(int linha);

Devolve o comprimento da *linha*.

void document.editBegin();

Inicia um grupo de edição para agrupar operações a desfazer/refazer. Certifique-se de invocar sempre o `editEnd()` tantas vezes quanto invoca o `editBegin()`. A invocação do `editBegin()` usa um contador de referências interno, isto é, esta chamada pode ser encadeada.

void document.editEnd();

Termina um grupo de edição. A última invocação do `editEnd()` (isto é, a correspondente à primeira chamada do `editBegin()`) termina o passo de edição.

int document.firstColumn(int linha);

Devolve a primeira coluna não em branco para a *linha* indicada. Se só existirem espaços em branco na linha, o valor devolvido é -1.

int document.lastColumn(int linha);

Devolve a última coluna que não esteja em branco para a *linha* indicada. Se só existirem espaços em branco na linha, o valor devolvido é -1.

```
int document.prevNonSpaceColumn(int linha, int coluna); int  
document.prevNonSpaceColumn(Cursor cursor);
```

Devolve a coluna com caracteres não-brancos que começa na posição de cursor indicada e pesquisa para trás.

```
int document.nextNonSpaceColumn(int linha, int coluna); int  
document.nextNonSpaceColumn(Cursor cursor);
```

Devolve a coluna com caracteres não-brancos que começa na posição de cursor indicada e pesquisa para a frente.

```
int document.prevNonEmptyLine(int linha);
```

Devolve a primeira linha não-vazia que contém caracteres não-nulos, pesquisando depois para trás.

```
int document.nextNonEmptyLine(int linha);
```

Devolve a primeira linha não-vazia que contém caracteres não-nulos, pesquisando depois para a frente.

```
bool document.isInWord(String caractere, int atributo);
```

Devolve *true* (verdadeiro), caso o *caractere* e *atributo* indicados possam fazer parte de uma palavra, caso contrário devolve *false* (falso).

```
bool document.canBreakAt(String caractere, int atributo);
```

Devolve *true* (verdadeiro) se o *caractere* indicado com o *atributo* indicado for adequado para mudar de linha, caso contrário devolve *false*.

```
bool document.canComment(int atributoInicial, int atributoFinal);
```

Devolve *true* (falso), caso um intervalo que começa e termina com os atributos indicados possa ser comentado; caso contrário, devolve *false* (falso).

```
String document.commentMarker(int atributo);
```

Devolve o marcador de comentários, usado em linhas únicas, para um determinado *atributo*.

```
String document.commentStart(int atributo);
```

Devolve o marcador de início de comentários, usado em linhas múltiplas, para um determinado *atributo*.

```
String document.commentEnd(int atributo);
```

Devolve o marcador de fim de comentários, usado em linhas múltiplas, para um determinado *atributo*.

```
Range document.documentRange();
```

Devolve um intervalo que engloba todo o documento.

```
Cursor documentEnd();
```

Devolve um cursor posicionado na última coluna da última linha do documento.

```
bool isValidTextPosition(int linha, int coluna); bool  
isValidTextPosition(Cursor cursor);
```

Devolve *true* (verdadeiro), se a posição atual do cursor estiver em uma posição de texto válida. Uma posição de texto é válida apenas se estiver localizada no início, no meio ou no fim de uma linha válida. Além disso, uma posição de texto é inválida se estiver localizada em um substituto Unicode.

Desde: KDE 5.0

```
int document.attribute(int linha, int coluna); int document.attribute(C-  
ursor cursor);
```

Devolve o atributo na posição do cursor indicada.

```
bool document.isAttribute(int linha, int coluna, int atributo); bool  
document.isAttribute(Cursor cursor, int atributo);
```

Devolve *true* (verdadeiro), se o atributo na posição do cursor indicada for igual a *atributo*, caso contrário devolve *false* (falso).

```
String document.attributeName(int linha, int coluna); String  
document.attributeName(Cursor cursor);
```

Devolve o nome do atributo como um texto legível. Isto é igual ao nome *itemData* dos arquivos de realce de sintaxe.

```
bool document.isAttributeName(int linha, int coluna, String nome); bool  
document.isAttributeName(Cursor cursor, String nome);
```

Devolve *true* (verdadeiro), se o nome do atributo, numa dada posição do cursor, corresponder ao *nome* indicado, caso contrário devolve *false* (falso).

```
String document.variable(String chave);
```

Devolve o valor da variável do documento identificada pela *chave*. Se a variável do documento não existir, o valor devolvido é um texto em branco.

```
void document.setVariable(String chave, String valor);
```

Define o valor da variável do documento solicitada pela *chave*.

Veja também: [Variáveis de documento do Kate](#)

Desde: KDE 4.8

```
int document.firstVirtualColumn(int linha);
```

Devolve a coluna virtual do primeiro caractere não-nulo na linha indicada, ou então -1 se a linha estiver em branco ou só tiver caracteres de espaços em branco.

```
int document.lastVirtualColumn(int linha);
```

Devolve a coluna virtual do último caractere não-nulo na linha indicada, ou então -1 se a linha estiver em branco ou só tiver caracteres de espaços em branco.

```
int document.toVirtualColumn(int linha, int coluna);
```

```
int document.toVirtualColumn(Cursor cursor); Cursor
```

```
document.toVirtualCursor(Cursor cursor);
```

Converte a posição do cursor 'real' para uma posição virtual, retornando um *int* ou um objeto *Cursor*.

```
int document.fromVirtualColumn(int linha, int colunaVirtual);
```

```
int document.fromVirtualColumn(Cursor cursorVirtual); Cursor
```

```
document.fromVirtualCursor(Cursor cursorVirtual);
```

Converte a posição virtual do cursor para uma posição do cursor 'real', retornando um *int* ou um objeto *Cursor*.

```
Cursor document.anchor(int linha, int coluna, Char caractere); Cursor
```

```
document.anchor(Cursor cursor, Char caractere);
```

Pesquisa para trás pelo caractere indicado, começando na posição do cursor indicada. Por exemplo, se for passado o '(', como caractere, esta função irá devolver a posição do '(' de abertura. Isto implica uma contagem das referências, isto é, os outros '(...)' são ignorados.

```
Cursor document.rfind(int linha, int coluna, String texto, int atributo  
= -1); Cursor document.rfind(Cursor cursor, String texto, int atributo =  
-1);
```

Pesquisa para trás pelo texto indicado, com o *atributo* apropriado. O argumento *atributo* é ignorado se for igual a -1. O cursor devolvido é inválido, caso o texto não tenha sido encontrado.

```
int document.defStyleNum(int linha, int coluna); int  
document.defStyleNum(Cursor cursor);
```

Devolve o estilo padrão que é usado na posição do cursor indicada.

```
bool document.isCode(int linha, int coluna); bool document.isCode(Cursor  
cursor);
```

Devolve `true` (verdadeiro) se o atributo na posição do cursor indicada não for igual a todos os seguintes estilos: `dsComment`, `dsString`, `dsRegionMarker`, `dsChar`, `dsOthers`.

```
bool document.isComment(int linha, int coluna); bool  
document.isComment(Cursor cursor);
```

Devolve `true` se o atributo do caractere na posição do cursor for `dsComment`, caso contrário, devolve `false`.

```
bool document.isString(int linha, int coluna); bool document.isString(C-  
ursor cursor);
```

Devolve `true` (verdadeiro) se o atributo do caractere na posição do cursor for `dsString`, caso contrário, devolve `false` (falso).

```
bool document.isRegionMarker(int linha, int coluna); bool  
document.isRegionMarker(Cursor cursor);
```

Devolve `true` (verdadeiro) se o atributo do caractere na posição do cursor for `dsRegionMarker`, caso contrário, devolve `false` (falso).

```
bool document.isChar(int linha, int coluna); bool document.isChar(Cursor  
cursor);
```

Devolve `true` (verdadeiro) se o atributo do caractere na posição do cursor for `dsChar`, caso contrário, devolve `false` (falso).

```
bool document.isOthers(int linha, int coluna); bool document.isOthers(C-  
ursor cursor);
```

Devolve `true` (verdadeiro) se o atributo do caractere na posição do cursor for `dsOthers`, caso contrário, devolve `false` (falso).

```
void document.indent(Range intervalo, int número);
```

Recua todas as linhas em *intervalo* por *número* de tabulações ou *número* vezes `tabSize` espaços, dependendo das preferências do usuário. O parâmetro *número* pode ser negativo.

6.4.3.5 A API do editor

Além da API de documento e visualização, existe uma API de editor geral que fornece funções para funcionalidades gerais de script do editor.

```
String editor.clipboardText();
```

Retorna o texto que está atualmente na área de transferência global.

Desde: KDE Frameworks 5.50

```
String editor.clipboardHistory();
```

O editor mantém um histórico da área de transferência que contém até 10 entradas. Esta função retorna todas as entradas que estão atualmente no histórico da área de transferência.

Desde: KDE Frameworks 5.50

```
void editor.setClipboardText(String texto);
```

Define o conteúdo da área de transferência para *texto*. O *texto* será adicionado ao histórico da área de transferência.

Desde: KDE Frameworks 5.50

Capítulo 7

Configurar o KatePart

A seleção de **Configurações** → **Configurar o Aplicativo...**, a partir do menu, provoca o aparecimento da caixa de diálogo **Configurar**. Esta caixa de diálogo pode ser usada para alterar algumas das várias opções existentes. As opções disponíveis para alteração podem variar de acordo com a categoria que o usuário seleciona, numa lista vertical do lado esquerdo da janela. Recorrendo aos três botões na parte inferior da janela, o usuário poderá controlar o processo.

Você pode invocar o sistema de **Ajuda**, aceitar as configurações atuais e fechar o diálogo através do botão **OK** ou **Cancelar**. As categorias **Aparência**, **Fontes & Cores**, **Edição**, **Abrir/Salvar** e **Extensões** são detalhadas abaixo.

7.1 Configuração do Componente de Edição

Este grupo contém todas as páginas relacionadas com o componente de edição do KatePart. A maioria das configurações aqui são os valores pré-definidos, que podem ser substituídos ao [definir um tipo de arquivo](#), nas [Variáveis do Documento](#) ou através da sua alteração, durante uma sessão de edição do documento.

7.1.1 Aparência

7.1.1.1 Geral

Fonte do editor

Aqui, o usuário poderá escolher a fonte do editor de texto. Você pode escolher qualquer fonte disponível no seu sistema, assim como definir um tamanho padrão. Um texto de exemplo é exibido na parte inferior da janela, para que você possa ver o resultado das suas opções.

Para mais informações sobre como selecionar uma fonte, leia a seção [Escolhendo as fontes na documentação dos Fundamentos do KDE](#).

Mostrar indicadores de espaço em branco

Nunca

O editor nunca mostrará pontos para indicar a presença de espaços em branco.

No final da linha

O editor mostrará pontos para indicar a presença de espaços em branco extra no fim das linhas.

Sempre

O editor sempre mostrará pontos para indicar a presença de espaços em branco.

Tamanho do indicador de espaço em branco

Use o controle deslizante para alterar o tamanho do marcador indicador visível.

Mostrar indicadores de tabulação

Quando selecionado, o editor mostrará um símbolo » para indicar a presença de uma tabulação no texto.

Correspondência de parênteses

Realçar o intervalo entre os parênteses selecionados

Se esta opção estiver ativa, o intervalo entre os parênteses selecionados será realçado.

Mostrar uma visualização do parênteses aberto correspondente

Quando ativado, o editor exibirá uma dica de ferramenta com o parênteses de abertura correspondente.

Piscar parênteses correspondentes quando o cursor mover sobre o outro parêntesis do par

Se estiver assinalado, a passagem sobre os sinais gráficos ({, [,], }, (ou)) irá piscar rapidamente o sinal correspondente.

Mostrar linhas de recuo

Se esta opção estiver assinalada, o editor irá mostrar linhas verticais para ajudar a identificar as linhas de recuo.

Contagens

Mostrar quantidade de palavras

Exibe o número de palavras e caracteres no documento e na seleção atual na barra de status. Esta opção também está disponível no menu de contexto da barra de status.

Mostrar número de linhas

Exibe o número total de linhas no documento na barra de status. Esta opção também está disponível no menu de contexto da barra de status.

Dobrar a primeira linha

Se estiver assinalado, a primeira linha é dobrada, se possível. Isso é útil se o arquivo começar com um comentário, como por exemplo, uma informação sobre direitos autorais.

Quebra de linhas dinâmica

Se esta opção estiver marcada, as linhas de texto serão quebradas na borda da visão da tela.

Quebra dinâmica no marcador estático de quebra de linhas

Quando selecionado, o editor quebra as linhas dinamicamente na [posição de quebra de linha estática](#).

Desconsiderar os limites das palavras para quebra dinâmica

Quando selecionada, a opção do editor não leva em consideração os limites das palavras ao quebrar as linhas de texto.

Indicadores de quebra de linhas dinâmico

Escolhe quando deverão ser mostrados os indicadores de quebra de linha, sendo as opções **Desligado**, **Seguir número de linhas** ou **Sempre ligado**.

Recuar linhas quebradas

Além disso, isto permite-lhe definir uma largura máxima da tela, como porcentagem, após a qual as linhas repartidas de forma dinâmica não serão mais alinhadas na vertical. Por exemplo, a 50%, as linhas cujos níveis de recuo sejam mais profundas que 50% da largura da tela não terão qualquer alinhamento vertical aplicado às linhas repartidas subsequentes.

Multiplicador de altura da linha

Esse valor será multiplicado pela altura de linha padrão da fonte. Um valor de 1,0 significa que a altura padrão será usada.

7.1.1.2 Bordas

Dobradura de bloco de código

Mostrar setas para recolher blocos de código

Se esta opção estiver assinalada, a janela atual irá mostrar marcações para dobrar ou desdobrar o código, se este estiver disponível.

Mostrar uma visualização do bloco de código dobrado ao passar o mouse

Se estiver marcada, a passagem do ponteiro do mouse sobre uma região dobrada mostra uma visualização do texto dobrado em uma janela popup.

Visibilidade das setas de dobradura

Alterna as setas de recolhimento entre **Mostrar ao passar o mouse** e **Mostrar sempre**.

Lado esquerdo

Mostrar marcas

Se esta opção estiver assinalada, você verá uma borda para os ícones do lado esquerdo. A borda dos ícones mostra os sinais de favoritos, por exemplo.

Mostrar número das linhas

Se esta opção estiver assinalada, você verá os números das linhas do lado esquerdo.

Realçar linhas alteradas e não salvas

Se estiver assinalada, os marcadores de modificação das linhas ficarão visíveis. Para mais informações, veja em Seção 3.9.

Barras de rolagem

Mostrar marcas

Se esta opção estiver assinalada, a janela atual irá mostrar algumas marcas na barra de rolagem vertical. Essas marcas poderão mostrar, por exemplo, os favoritos.

Mostrar a visualização ao passar o mouse na barra de rolagem

Se esta opção estiver marcada e você posicionar o cursor do mouse sobre a barra de rolagem, uma pequena pré-visualização de texto com várias linhas do documento atual ao redor da posição do cursor será exibida. Isso permite que você alterne rapidamente para outra parte do documento.

Minimapa

Mostrar minimapa

Se esta opção estiver assinalada, cada nova visão exibirá um minimapa do documento na barra de rolagem vertical.

Para obter mais informações sobre o minimapa na barra de rolagem, veja Seção 3.10.

Largura do minimapa

Ajusta a largura em pixels do minimapa na barra de rolagem.

Visibilidade das barras de rolagem

Ativa, desativa ou exibe a barra de rolagem somente quando necessário. Clique com o botão esquerdo do mouse no retângulo azul para exibir o intervalo de números de linha do documento exibido na tela. Mantenha o botão esquerdo do mouse pressionado fora do retângulo azul para rolar automaticamente pelo documento.

Ordenar menu de favoritos

Por data de criação

Cada novo favorito será adicionado no fim, independentemente de onde se encontra colocado no documento.

Por número de linha

Os favoritos serão ordenados pelos números de linhas onde estão colocados.

7.1.2 Temas de cores

Esta seção da janela permite configurar todas as cores em qualquer tema de cores que você tenha, além de criar novos temas, excluir os existentes ou simplesmente **Seguir o esquema de cores do sistema**. Cada esquema possui configurações para cores e estilos de texto normal e destacado.

O KatePart selecionará previamente o tema atualmente ativo para você; se quiser trabalhar sobre um tema diferente, comece por selecioná-lo na lista **Selecionar tema**. Com os botões **Copiar** e **Excluir** você pode criar um novo tema (copiando um existente) ou remover um existente.

Isto é descrito em detalhes em Seção [6.3.5](#).

7.1.3 Edição

7.1.3.1 Geral

Quebra de linha

A quebra de linha é um recurso que faz com que o editor inicie automaticamente uma nova linha de texto e mova (quebre) o cursor para o início da nova linha. O KatePart automaticamente iniciará uma nova linha de texto quando a linha atual atingir o comprimento especificado pela opção [Quebrar linhas em:](#).

Quebrar linhas em uma coluna fixa

Ativa ou desativa a quebra de linha estática.

Desenhar uma linha vertical na coluna da quebra de linha

Se esta opção estiver assinalada, será desenhada uma linha vertical na coluna da quebra de linha, como está definido na opção **Configurações** → **Configurar Editor...** na página de Edição. Lembre-se que o marcador de quebra de linha só é desenhado se você usar uma fonte monoespaçada.

Quebrar linhas em:

Se a opção [Ativar quebra de linha em uma coluna fixa](#) estiver selecionada esta entrada determina o comprimento (em caracteres) no qual o editor automaticamente iniciará uma nova linha.

Modo de entrada padrão

O modo de inserção de dados será ativado ao abrir um novo modo de exibição. Você ainda poderá ativar/desativar o modo de entrada VI, para um determinado modo de exibição através do menu **Editar**.

Parênteses

Se a opção **Fechar parênteses automaticamente ao digitar um parênteses de abertura** estiver selecionada, quando o usuário digitar um sinal gráfico de abertura ([, (ou {) o KatePart irá inserir automaticamente o sinal gráfico de fechamento (],) ou }) à direita do cursor.

Caracteres de fechamento

É possível selecionar os caracteres delimitadores usando a lista suspensa correspondente.

Quando o texto for selecionado, a digitação de um desses caracteres envolve o texto selecionado.

Copiar e colar

Mover o texto selecionado quando arrastado

Esta opção permite arrastar e soltar o texto selecionado dentro da janela do editor.

Copiar/recortar a linha atual se invocado sem nenhum texto selecionado

Se esta opção estiver ativa e se a seleção de texto estiver vazia, as ações de cópia e corte são efetuadas para a linha de texto na posição atual do cursor.

Não mover o cursor do texto ao colar com o mouse

Se esta opção estiver ativada e você colar algum texto na janela do editor com o botão do meio do mouse, o KatePart não moverá o cursor de texto para a posição clicada.

7.1.3.2 Navegação de texto

Movimento do cursor de texto

Início e fim inteligentes

Quando estiver selecionado, pressionar na tecla Home, fará com que o cursor salte os espaços em branco e vá para o início do texto de uma linha.

PageUp/PageDown move o cursor

Esta opção muda o comportamento do cursor quando o usuário pressiona as teclas **Page Up** e **Page Down**. Se não estiver selecionada, o cursor de texto manterá sua posição relativa dentro do texto visível no KatePart conforme o novo texto se torna visível como resultada da operação. Assim, se o cursor estiver no meio do texto visível quando a operação ocorre, ele permanecerá lá (exceto quando o início ou fim são atingidos). Com esta opção selecionada, o primeiro pressionamento de tecla fará com que o cursor se mova para cima ou para baixo do texto visível conforme uma nova página de texto é exibida.

Ativar movimento de cursor em Camel Case

Esta opção altera o comportamento do cursor quando o usuário pressiona o atalho **Ctrl-Seta para a esquerda** ou **Ctrl-Seta para a direita**. Se desmarcada, o cursor de texto pula sobre as palavras inteiras. Com esta opção marcada, o cursor pula nas letras camel case.

Centralizar cursor automaticamente:

Indica o número de linhas a manter visível acima e abaixo do cursor, sempre que possível.

Modo de seleção de texto

Normal

As seleções serão sobrepostas pelo texto escrito e serão perdidas com o movimento do cursor.

Persistente

As seleções serão mantidas mesmo após o movimento do cursor ou a digitação.

Permitir rolagem após o fim do documento

Esta opção permite-lhe deslocar além do fim do documento. Isto poderá ser usado para centrar na vertical o fundo do documento, ou para colocá-lo no topo da janela atual.

A tecla Backspace remove a base do caractere e a sua terminação diacrítica

Quando estiver selecionada, os caracteres compostos são removidos com as suas terminações em vez de remover apenas o caractere de base. Isso é útil para os localidades do Índico.

Modificador multicursor

Esta opção permite definir o modificador que será usado para criar vários cursores com um clique do botão esquerdo do mouse. Você precisa pressionar os modificadores e clicar com o botão esquerdo do mouse para criar um cursor no local desejado. Consulte [Criando vários cursores](#) para descobrir outras maneiras de criar vários cursores.

7.1.3.3 Recuo

Modo de recuo padrão:

Selecione o modo de recuo automático que desejar usar por padrão. Recomenda-se que use o **Nenhum** ou o **Normal** aqui e use as configurações do tipo de arquivo para definir outros modos de recuo para os formatos de texto, como o código em C/C++ ou o XML.

Recuar usando

Tabulações

Quando estiver ativo, o editor irá inserir caracteres de tabulação quando você clicar na tecla **Tab** ou usar o [recuo automático](#).

Espaços

Quando isto estiver ativo, o editor irá inserir um número calculado de espaços, de acordo com a posição no texto e a configuração `tab-width`, quando pressionar a tecla **Tab** ou usar o [recuo automático](#).

Tabulações e espaços

Quando a opção estiver ativa, o editor irá inserir espaços, como descrito acima, quando recuar ou pressionar a tecla **Tab** no início de uma linha, mas irá inserir tabulações quando pressionar na tecla **Tab** no meio ou no fim de uma linha.

Largura da tabulação:

Isto configura o número de espaços que são apresentados no lugar de um caractere de tabulação.

Largura do recuo:

A largura do recuo é o número de espaços usado para recuar uma linha. Se estiver configurado para recuar com tabulações, será inserido um caractere de tabulação se o recuo for divisível pela largura da tabulação.

Propriedades do recuo

Manter espaços extras

Se esta opção estiver desativada, a alteração do nível de recuo alinha uma linha a um múltiplo da largura definida em **Largura do recuo**.

Ajustar o recuo do texto colado da área de transferência

Se esta opção for selecionada, o texto colado da área de transferência será recuado. A execução da ação **Desfazer** irá remover o recuo.

Ações de recuo

A tecla Backspace nos espaços iniciais retira o recuo

Se esta opção for selecionada, a tecla **Backspace** diminui o nível de recuo se o cursor estiver localizado no espaço em branco inicial de uma linha.

Ação da tecla Tab (se não existir seleção)

Se você quiser que o **Tab** alinhe a linha atual no bloco de código atual, como acontece no Emacs, torne o **Tab** um atalho para a ação **Formatar recuo**.

Sempre avançar para a tabulação seguinte

Se esta opção for selecionada, a tecla **Tab** insere sempre espaços em branco até atingir a posição da tabulação seguinte. Se a opção **Inserir espaços em vez de tabulações**, na página **Geral** em **Edição** estiver ativa, são inseridos espaços; caso contrário, é inserida uma única tabulação.

Sempre aumentar o nível de recuo

Se esta opção for selecionada, a tecla **Tab** recua sempre a linha atual de acordo com o número de posições de caracteres indicada na **Largura do recuo**.

Aumentar o nível de recuo se estiver num espaço em branco inicial

Se esta opção estiver assinalada, a tecla **Tab** recua a linha atual ou avança para a próxima posição de tabulação. Se o ponto de inserção estiver sobre ou antes do primeiro caractere que não seja espaço na linha, ou se existir algo selecionado, a linha atual é recuada pelo número de posições de caracteres indicada em **Largura do recuo**. Se o ponto de inserção estiver após o primeiro caractere não-espaço e nada estiver selecionado, são inseridos espaços em branco até atingir a próxima posição de tabulação: se a opção **Inserir espaços em vez de tabulações** da página **Geral** em **Edição** estiver ativa, são inseridos espaços; caso contrário, é inserida uma única tabulação.

7.1.3.4 Complementação automática

Geral

Ativar completamento automático

Se habilitado, aparece um campo de completação automática, mostrando uma lista com os itens de texto para completar o texto atual sob o cursor.

Selecionar automaticamente a primeira entrada da complementação

Se ativada, a primeira opção de autocompletar é sempre pré-selecionada para que você possa inseri-la com a tecla **Enter**. Se você não deseja esse comportamento, por exemplo se quiser que pressionar **Enter** insira apenas uma nova linha, desative esta opção.

Tamanho mínimo da palavra a completar

Ao digitar o texto, o completamento de palavras procura por palavras no documento iniciando com o texto já digitado. Esta opção configura a quantidade mínima de caracteres que são necessários para tornar o completamento de palavras ativo e exibir uma caixa de completamento.

Remover o fim com a completção

Remover o fim de uma palavra anterior quando o item de completção for escolhido a partir de uma lista.

Complementação de palavras-chave

Se estiver assinalado, a complementação automática incorporada usa as palavras-chave definidas pelo realce de sintaxe.

7.1.3.5 Verificação ortográfica

Estas opções de configuração estão descritas na documentação do módulo do Configurações do sistema sobre [Verificação Ortográfica](#).

7.1.3.6 Modo de entrada Vi

Geral

Permitir que os comandos do Vi substituam os atalhos do Kate

Quando selecionado, os comandos do 'Vi' irão substituir os comandos incorporados do KatePart. Por exemplo: o **Ctrl+R** irá refazer, substituindo a ação padrão (mostrar a janela de pesquisa e substituição).

Mostrar números de linha relativos

Se isto estiver assinalado, a linha atual sempre se irá se referir à linha 0. As linhas acima e abaixo aumentam o número da linha de forma relativa.

Mapeamento de teclas

A associação de teclas é usada para alterar o significado das teclas pressionadas. Isto permite-lhe mover os comandos para outras teclas ou criar sequências de teclas especiais para fazer uma determinada série de comandos.

Exemplo:

F2 -> I-- Esc

Isto irá anteceder o **I--** com uma linha, ao pressionar **F2**.

7.1.4 Abrir/Salvar

7.1.4.1 Geral

Formato do arquivo

Codificação

Isto define a codificação padrão a ser usada para abrir/salvar os arquivos, se não for alterado no diálogo de abertura/gravação ou na opção da linha de comandos.

Deteção da codificação

Selecione um item na lista, quer para desativar a deteção automática como para usar a **Universal**, de modo a ativar a deteção automática para todas as codificações. Contudo, como isto provavelmente só irá detectar o UTF-8/UTF-16, selecionar uma região irá usar heurísticas personalizadas para melhores resultados. Se a codificação selecionada como padrão acima, a codificação especificada na janela de abertura/salvamento de arquivos ou a codificação especificada na linha de comando corresponder ao conteúdo do arquivo, esta deteção será realizada.

Codificação secundária

Isto define a codificação secundária a ser tentada quando abrir os arquivos, caso não funcione a codificação padrão, a indicada através da janela para abrir/salvar ou pela linha de comando, de modo a corresponder ao conteúdo do arquivo. Antes desta ser usada, a componente do Kate irá tentar deduzir a codificação, lendo um possível marcador de ordem dos 'bytes' no início do arquivo; caso se detecte algum conhecido, será escolhida a codificação correta, caso contrário será testada esta como último recurso.

Fim da linha

Escolha o seu fim de linha preferido para o seu documento ativo. Você poderá optar pelo padrão UNIX[®], DOS/Windows[®] ou o Macintosh.

Deteção automática de fim de linha

Assinale isto se quiser que o editor detecte automaticamente o tipo de fim da linha. O primeiro tipo encontrado será o usado para todo o arquivo.

Ativar o marcador da ordem de bytes (BOM)

O marcador da ordem dos bytes é uma sequência especial no início dos documentos codificados em Unicode. Ajuda os editores a abrir os documentos de texto com a codificação Unicode correta. Para mais informações, veja o [Marcador da Ordem de Bytes](#).

Limite do tamanho da linha

Infelizmente, devido a deficiências no Qt[™], o KatePart tem um desempenho ruim ao lidar com linhas extremamente longas. Por este motivo, o KatePart tentará quebrar as linhas automaticamente, quando o tamanho for maior que o número de caracteres aqui indicado. Para desativar isto, defina como 0.

Limpezas automáticas ao salvar

Remover espaços finais

O editor irá eliminar automaticamente os espaços extras no fim das linhas de texto enquanto salva o arquivo. Você pode selecionar **Nunca** para desativar essa funcionalidade, **Nas linhas modificadas** para fazer apenas nas linhas que você modificou desde a última gravação do documento ou **Em todo o documento** para removê-las de forma incondicional do documento inteiro.

Adicionar nova linha no fim do arquivo ao salvar

O editor irá adicionar automaticamente uma nova linha ao fim do arquivo, caso não exista, ao salvar o arquivo.

Habilitar salvamento automático (apenas para arquivos locais)

Marque esta opção se quiser que o editor salve automaticamente os documentos enquanto você estiver trabalhando neles.

Salvar automaticamente o documento quando o foco sai do editor

O editor salvará automaticamente os documentos quando você alternar para algo fora do editor, por exemplo o painel do terminal no Kate.

Intervalo de salvamento automático

Você pode definir o intervalo de salvamento automático em segundos aqui. Se o intervalo for 0, o documento não será salvo automaticamente após o intervalo definido.

7.1.4.2 Avançado

Gravar um arquivo de backup ao salvar para

Fazer backup ao salvar fará com que o KatePart copie o arquivo de disco (a versão salva anteriormente do arquivo) para <prefixo><nomedoarquivo><sufixo> antes de salvar as novas alterações. Um arquivo de backup pode ajudar você a recuperar o trabalho caso algo dê

errado durante o salvamento ou se você quiser recuperar posteriormente a versão anterior do arquivo. O sufixo é definido como ~ por padrão e o prefixo está vazio por padrão.

Arquivos locais

Assinale isto se quiser cópias de segurança dos arquivos locais ao salvar.

Arquivos remotos

Assinale isto se quiser cópias de segurança dos arquivos remotos ao salvar.

Prefixo para os arquivos de backup

Indique o prefixo com que antecede os nomes das cópias de segurança.

Sufixo para os arquivos de backup

Indique o sufixo a ser adicionado aos nomes das cópias de segurança.

Modo arquivo de troca

O KatePart é capaz de recuperar (a maior parte) do trabalho não salvo no caso de um travamento ou falta de energia. Um arquivo temporário (<nomedoarquivo>.kate-swp) é criado após o documento ser editado. Se o usuário não salvar as mudanças e o KatePart travar, o arquivo temporário permanecerá no disco. Ao abrir um arquivo, o KatePart verifica se existe um arquivo temporário para o documento e, caso exista, ele pergunta se o usuário deseja recuperar os dados perdidos ou não. O usuário também pode ver as diferenças entre o arquivo original e o recuperado. O arquivo temporário é excluído após cada salvamento ou saída normal do programa.

O KatePart sincroniza os arquivos temporários a cada 15 segundos, mas somente se eles tiverem mudado desde a última sincronização. O usuário poderá desativar a sincronização dos arquivos temporários se desejar, selecionando a opção **Desativar**, mas isto pode levar a mais perda de dados.

Quando o arquivo de troca está habilitado, é possível alternar entre dois modos, nomeadamente **Habilitado, armazenar no diretório padrão** e **Habilitado, armazenar em diretório personalizado**.

Armazenar arquivos de troca em

Por padrão, os arquivos temporários são salvos na mesma pasta do arquivo principal. Quando a **Habilitado, armazenar em diretório personalizado** for selecionado, os arquivos temporários serão criados na pasta indicada. Isso é útil para sistemas de arquivos de rede, de forma a evitar o tráfego de rede desnecessário.

Salvar arquivos de troca a cada

O KatePart sincroniza os arquivos temporários a cada 15 segundos, mas somente se eles tiverem mudado desde a última sincronização. O usuário poderá alterar o intervalo de sincronização como desejar.

7.1.4.3 Modos & Tipos de Arquivos

Esta página permite-lhe substituir a configuração padrão dos documentos dos tipos MIME indicados. Quando o editor carregar um documento, ele verificará se o arquivo corresponde às máscaras de arquivos ou tipos MIME para cada um dos tipos definidos e, se corresponder, aplica as variáveis definidas. Se de um tipo de arquivo corresponder, será usado o que tiver maior prioridade.

Tipo de arquivo:

O tipo de arquivo com a maior prioridade é o que será apresentado na primeira lista. Se forem encontrados mais tipos de arquivo, eles serão também listados.

Novo

Isto é usado para criar um novo tipo de arquivo. Após clicar este botão, os campos abaixo ficam em branco, para que você possa preencher as propriedades que deseja para o novo tipo de arquivo.

Excluir

Para remover um tipo de arquivo existente, selecione-o na lista e clique no botão Excluir.

Propriedades do tipo de arquivo atual

O tipo de arquivo com a maior prioridade é o que será apresentado na primeira lista. Se forem encontrados mais tipos de arquivo, eles serão também listados.

Nome:

O nome do tipo de arquivo será o texto do item de menu correspondente. Este nome é mostrado no menu **Ferramentas** → **Tipos de arquivo**.

Seção:

O nome da seção é usado para organizar os tipos de arquivos em menus. Ele também é usado no menu **Ferramentas** → **Tipos de arquivo**.

Variáveis:

Este texto permite-lhe configurar as opções do KatePart para os arquivos selecionados por este tipo MIME, usando as variáveis do KatePart. Você poderá definir quase todas as opções de configuração, como o 'highlight', 'indent-mode', etc.

Pressione **Editar** para ver uma lista de todas as variáveis disponíveis e suas descrições. Selecione a caixa à esquerda para ativar uma determinada variável e então defina o valor da variável à direita. Algumas variáveis fornecem uma caixa combinada para selecionar os valores possíveis enquanto outras necessitam que você insira um valor válido manualmente.

Para informações completas sobre estas variáveis, veja [Configurando com variáveis de documento](#).

Realce:

Se você criar um novo tipo de arquivo, esta lista permite-lhe selecionar um tipo de arquivo para o realce.

Modo de recuo:

A lista indica o modo de recuo para os documentos novos.

Extensões dos arquivos:

As máscaras com caracteres especiais permitem-lhe selecionar arquivos pelo nome. Uma máscara típica usa um asterisco e a extensão do arquivo, como por exemplo *.txt; *.text. O texto é uma lista de máscaras separada por ponto e vírgula.

Tipos MIME:

Exibe um assistente que o ajuda a escolher facilmente tipos MIME.

Prioridade:

Escolhe a prioridade deste tipo de arquivo. Se mais de um tipo de arquivo corresponder a um arquivo, será utilizado o com a prioridade mais elevada.

7.2 Configurando com as Variáveis do Documento

As variáveis do KatePart são a implementação do KatePart para as variáveis do documento, de forma semelhante à que acontece no Emacs e no VI. Na 'katepart', as linhas têm o seguinte formato: **kate: NOMEVARIÁVEL VALOR; [NOMEVARIÁVEL VALOR; ...]**. As linhas poderão estar, claro, num comentário, se o arquivo estiver num formato que suporte comentários. Os nomes das variáveis são palavras únicas (sem espaços) e tudo o que estiver até o próximo ponto e vírgula é o valor. Esse ponto e vírgula é obrigatório.

Aqui está uma linha de variável de exemplo, obrigando a configuração do recuo para um arquivo em C++, Java™ ou JavaScript:

```
// kate: replace-tabs on; indent-width 4; indent-mode cstyle;
```

NOTA

Só as primeiras e últimas 10 linhas serão pesquisadas pelas linhas de variáveis.

Além disso, as variáveis do documento podem ser colocadas em um arquivo chamado `.kateconfig` em qualquer diretório, e as opções configuradas serão aplicadas se as modelines forem inseridas em qualquer arquivo no diretório e subdiretórios. As variáveis de documento no `.kateconfig` usam a mesma sintaxe das modelines, mas com [opções estendidas](#).

Existem variáveis para suportar quase todas as configurações no KatePart e os plugins adicionais podem usar as variáveis; nesse caso, isto deverá estar indicado na documentação do plugin.

O KatePart oferece suporte à leitura de configurações de arquivos `.editorconfig`, quando a biblioteca `editorconfig` está instalada. O KatePart busca automaticamente por um arquivo `.editorconfig` sempre que você abre um arquivo. No entanto, ele dá prioridade a arquivos `.kateconfig`.

7.2.1 Como o KatePart usa Variáveis

Ao ler a configuração, o 'katepart' procura nos seguintes locais (nesta ordem):

- A configuração global.
- Os dados opcionais da sessão.
- A configuração do "Tipo de arquivo".
- As variáveis do documento no `.kateconfig`.
- As variáveis do próprio documento.
- As configurações feitas durante a edição do menu ou da linha de comando.

Como pode ver, as variáveis do documento somente podem ser sobrepostas por alterações feitas no momento da execução. Sempre que um documento for salvo, as variáveis do documento são lidas novamente e irão sobrepor as alterações feitas, usando os itens do menu ou nas linhas de comando.

Todas as variáveis que não estejam indicadas abaixo são guardadas no documento e poderão ser pesquisadas por outros objetos, como os plugins, usando-as para os seus próprios fins. Por exemplo, o modo de recuo variável usa as variáveis do documento para a sua própria configuração.

As variáveis aqui listadas dizem respeito ao KatePart versão 5.38. Poderão ser adicionadas mais variáveis no futuro. Existem 3 tipos de valores definidos para as variáveis, com as seguintes expressões válidas:

- **BOOL** - on | off | true | false | 1 | 0
- **INTEIRO** - qualquer número inteiro
- **TEXTO** - todo o resto

7.2.2 Variáveis Disponíveis

auto-brackets [BOOL]

Ativar a inserção automática de parênteses.

auto-center-lines [INT]

Define o número de linhas de centralização automática.

background-color [TEXTO]

Define a cor de fundo do documento. O valor deverá ser algo que possa ser avaliado como uma cor válida, como por exemplo, **#ff0000**.

backspace-indent [BOOLEANO]

Ativa ou desativa a remoção do recuo quando o **Backspace** é pressionado.

block-selection [BOOL]

Ativa ou desativa a [seleção em bloco](#).

bom | **byte-order-mark** | **byte-order-marker** [BOOL]

Ativar/desativar o marcador da ordem dos bytes (BOM) ao salvar os arquivos no formato Unicode (utf8, utf16, utf32).

Desde: Kate 3.4 (KDE 4.4)

bracket-highlight-color [TEXTO]

Define a cor de realce dos parênteses. O valor deverá ser algo que possa ser avaliado como uma cor válida, como por exemplo, **#ff0000**.

current-line-color [TEXTO]

Define a cor da linha atual. O valor deverá ser algo que possa ser avaliado como uma cor válida, como por exemplo **#ff0000**.

default-dictionary [TEXTO]

Configurar o dicionário padrão usado na verificação ortográfica.

Desde: Kate 3.4 (KDE 4.4)

dynamic-word-wrap [BOOL]

Ativa ou desativa a [quebra de linha dinâmica](#).

eol | **end-of-line** [TEXTO]

Define o modo de fim-de-linha. Os valores válidos são o **unix**, o **mac** e o **dos**.

folding-markers [BOOL]

Ativa ou desativa a visualização dos [marcadores de expansão/contração do código](#).

folding-preview [BOOL]

Ativar a visualização de dobragem no contorno do editor.

font-size [INT]

Define o tamanho em pontos da fonte do documento.

font [TEXTO]

Define o tipo de fonte do documento. O valor deverá ser um nome de fonte válido, como por exemplo, **courier**.

hl | **syntax** [TEXTO]

Define o realce de sintaxe. Os textos usados são todos os nomes disponíveis nos menus. Por exemplo, para o C++, basta digitar **C++**.

icon-bar-color [TEXTO]

Define a cor da barra de ícones. O valor deverá ser algo que possa ser avaliado como uma cor válida, como por exemplo, **#ff0000**.

icon-border [BOOL]

Ativa ou desativa a visibilidade da borda dos ícones.

indent-mode [TEXTO]

Define o modo de recuo automático. As opções **none**, **normal**, **cstyle**, **haskell**, **lilypond**, **lisp**, **python**, **ruby** e **xml** são reconhecidas. Veja a seção Seção 3.8 para mais detalhes.

indent-pasted-text [BOOLEANO]

Ativa/desativa o ajuste de recuo do texto colado da área de transferência.

Desde: Kate 3.11 (KDE 4.11)

indent-width [INTEIRO]

Define a largura do recuo.

keep-extra-spaces [BOOL]

Define se deseja manter os espaços extras ao calcular a largura do recuo.

line-numbers [BOOL]

Ativa ou desativa a visibilidade dos números de linha.

newline-at-eof [BOOLEANO]

Adiciona uma linha vazia no fim do arquivo (EOF) ao salvar o documento.

Desde: Kate 3.9 (KDE 4.9)

overwrite-mode [BOOL]

Ativa ou desativa o modo de sobreposição.

persistent-selection [BOOL]

Ativa ou desativa as [seleções persistentes](#).

replace-tabs-save [BOOL]

Ativa ou desativa a conversão de tabulações para espaços no salvamento.

replace-tabs [BOOL]

Ativa ou desativa a conversão dinâmica de tabulações para espaços no salvamento.

remove-trailing-spaces [TEXTO]

Remove os espaços finais ao salvar o documento. As opções válidas são:

- **none**, **-** ou **0**: nunca remove os espaços finais.
- **modified**, **mod**, **+** ou **1**: remove os espaços finais apenas nas linhas modificadas. Estas linhas são marcadas pelo sistema de modificação de linhas.
- **all**, ***** ou **2**: remove os espaços finais no documento inteiro.

scrollbar-minimap [BOOL]

Mostrar minimapa na barra de rolagem.

scrollbar-preview [BOOL]

Mostrar a visualização na barra de rolagem.

scheme [TEXTO]

Define o esquema de cores. O texto deverá ser o nome de um esquema de cores que exista na sua configuração, para que faça qualquer efeito.

selection-color [TEXTO]

Define a cor da seleção. O valor deverá ser algo que possa ser avaliado como uma cor válida, como por exemplo **#ff0000**.

show-tabs [BOOL]

Ativa ou desativa o caractere de tabulação visível.

smart-home [BOOL]

Ativa ou desativa a [navegação inteligente do Home](#).

tab-indent [BOOLEANO]

Ativa ou desativa o recuo com **Tab**.

tab-width [INT]

Define a largura de exibição do caractere de tabulação.

undo-steps [INT]

Define o número de passos a recordar no Desfazer/Refazer.

Nota: Obsoleto desde o Kate 3 no KDE4. Esta variável é ignorada. O número máximo de passos de anulação é ilimitado.

word-wrap-column [INT]

Define a largura da [quebra de linha estática](#).

word-wrap-marker-color [TEXTO]

Define a cor do marcador de quebra de linha. O valor deverá ser algo que possa ser avaliado como uma cor válida, como por exemplo, **#ff0000**.

word-wrap [BOOL]

Ativa ou desativa a quebra de linha estática.

7.2.3 Opções estendidas nos arquivos **.kateconfig**

O KatePart sempre procura um arquivo **.kateconfig** para arquivos locais (não arquivos remotos). Além disso, é possível definir opções com base em curingas (extensões de arquivo) da seguinte forma:

```
kate: tab-width 4; indent-width 4; replace-tabs on;  
kate-wildcard (*.xml): indent-width 2;  
kate-wildcard (Makefile): replace-tabs off;
```

Neste exemplo, todos os arquivos usam uma largura de tabulação de 4 espaços, uma largura de recuo de 4 espaços e as tabulações são substituídas por espaços quando expandidas. No entanto, para todos os arquivos ***.xml**, a largura do recuo é definida como 2 espaços. E os Makefiles usam tabulações, isto é as tabulações não são substituídas por espaços.

Os caracteres curinga são separados por ponto e vírgula, isto é você também pode especificar várias extensões de arquivo da seguinte forma:

```
kate-wildcard (*.json;*.xml): indent-width 2;
```

Além disso, você também pode usar o tipo MIME para corresponder a determinados arquivos, por exemplo para indentar todos os arquivos de origem C++ com 4 espaços, você pode escrever:

```
kate-mimetype (text/x-c++src): indent-width 4;
```

NOTA

Além do suporte em arquivos **.kateconfig**, curingas e variáveis `​​` de documento dependentes do tipo MIME também são suportados nos próprios arquivos como comentários.

Capítulo 8

Créditos e licença

Direitos autorais do KatePart e KWrite 2001-2022 da equipe do Kate.

Baseado no KWrite original, que possui Direitos Autorais 2000 para Jochen Wilhelmy digisnap@cs.tu-berlin.de

Contribuições:

- Christoph Cullmann cullmann@kde.org
- Michael Bartl michael.bartl1@chello.at
- Philip phlip_cpp@my-deja.com
- Anders Lund anders@alweb.dk
- Matt Newell newellm@proaxis.com
- Joseph Wenninger kde@jowenn.at
- Jochen Wilhelmy digisnap@cs.tu-berlin.de
- Michael Koch koch@kde.org
- Christian Gebauer gebauer@kde.org
- Simon Hausmann hausmann@kde.org
- Glen Parker glenebob@nwlink.com
- Scott Manson sdmanson@altel.net
- John Firebaugh jfirebaugh@kde.org
- Nibaldo González nibgonz@gmail.com

A documentação do KatePart é baseada na documentação original do KWrite, modificada para ser relevante a todos os usuários do KatePart.

A documentação original do KWrite foi escrita por Thad McGinnis ctmcginnis@compuserve.com, com diversas modificações de Christian Tibirna tibirna@kde.org. Foi revista e convertida para DocBook por Lauri Watts lauri@kde.org e atualizada por Anne-Marie Mahfouf annma@kde.org e Anders Lund anders@alweb.dk

A documentação atual do KatePart é mantida por T.C. Hollingsworth tchollingsworth@gmail.com. Para solicitar alterações, registre um bug no [Sistema de Rastreamento de bugs do KDE](#) ou forneça um patch no [nosso projeto no GitLab](#).

Tradução de Marcus Gama marcus.gama@gmail.com e André Marcelo Alvarenga alvarenga@kde.org

Esta documentação é licenciada sob os termos da [Licença de Documentação Livre GNU](#).

Este programa é licenciado sob os termos da [Licença Pública Geral GNU](#).

Capítulo 9

O modo de entrada do VI

Erlend Hamberg

Tradução: Marcus Gama

9.1 Modo de entrada do VI

O objetivo do modo VI não é ser um substituto completo para o Vim e suportar todas as funcionalidades do Vim. O objetivo é tornar o ‘jeito Vim’ de edição de texto - e os hábitos Vim aprendidos - disponíveis para os programas que usam o editor de texto do KatePart como seu editor interno.

O modo VI busca integrar-se de maneira transparente aos programas e se afastar do comportamento do Vim onde fizer sentido. Por exemplo, **:w** abrirá uma janela para salvar no modo VI do KatePart.

Para ativar o Modo de entrada VI para todas as novas visualizações, vá para **Configurações** → **Configurar o KatePart...+Edição** → **Modo de entrada VI**. Nesta página você pode definir opções para o Modo de entrada VI e definir e editar o mapeamento de teclado neste modo. O Modo de entrada VI pode também ser alternado com a configuração **Modo de entrada VI** no menu **Editar**. (O atalho de teclado padrão é **Meta+Ctrl+V** - onde o **Meta** é normalmente a tecla **Windows**).

NOTA

Muitos comandos de teclado do modo Vi são sensíveis à caixa, diferentemente da maioria dos atalhos de teclado do KDE. Isto significa que **y** e **Y** são comando diferentes. Para inserir o comando **y** (yank - empurrar), certifique-se de que o **Caps Lock** está desativado e pressione **Y**. Para inserir o comando **Y** (empurrar para o fim da linha), **Shift+Y**.

Isto não se aplica aos comandos que usam a tecla **Ctrl**, que podem ser inseridos independentemente do modo do **Caps Lock** e sem pressionar o **Shift**. No entanto, alguns comandos necessitam do uso da combinação da tecla **Ctrl** seguida por outra tecla que diferencie maiúsculas e minúsculas. Por exemplo, para inserir **Ctrl+W, h** (alternar para a visão dividida à direita) certifique-se de que o **Caps Lock** está desativado, pressione **Ctrl+W**, solte e então pressione **H**.

9.1.1 Incompatibilidades com o Vim

Existem algumas poucas funcionalidades do modo VI do KatePart que são incompatíveis com o Vim (sem contar as que estão faltando). Elas estão listadas abaixo juntamente com os motivos para esta incompatibilidade.

- KatePart: o **U** e o **Ctrl+R** são o 'Refazer'.
Vim: **Ctrl+R** é o 'refazer' normal, enquanto o **U** é usado para anular todas as últimas alterações numa linha.
O motivo para ter a ação **U** como refazer no modo VI do KatePart é que o atalho **Ctrl+R** por padrão é usado pela função substituir do KatePart (localizar e substituir). Por padrão, o modo VI não se sobrepõe aos atalhos do KatePart (isto pode ser configurado em **Configurações** → **Configurar o KatePart...+Edição** → **Modo de entrada VI**), assim a ação de refazer precisa estar disponível como uma tecla 'normal'. Além disso, o comportamento do comando **U** no Vim não corresponde exatamente ao sistema de refazer interno do KatePart, assim isto seria um suporte não tão trivial de qualquer maneira.
- KatePart: **print** mostra a janela **Imprimir**.
Vim: **print** imprime as linhas de um determinado intervalo como seu avô, o 'ed'.
Comandos como **:print** estão disponíveis não somente no modo VI para usuários que usam o KatePart 'normal' também - assim, o comando **:print** abre a janela de impressão - seguindo o princípio da menor surpresa ao invés de simplesmente imitar o comportamento do Vim.
- KatePart: **Y** copia até o fim da linha.
Vim: **Y** copia toda a linha, como no **yy**.
O comportamento do VI para o comando **Y** é na prática um erro. Para os comandos 'change' e 'delete', **cc/ dd** fará esta ação na linha atual e **C/D** funcionará da coluna do cursor até o final da linha. No entanto, **yy** e **Y** copia a linha atual. No Modo VI do KatePart, **Y** copia até o fim da linha. Isto é considerado 'mais lógico' na [documentação](#) do Vim.
- KatePart: **O** e **o** abre [*tantas*] novas linhas e entra no modo de inserção.
Vim: **O** e **o** abre uma nova linha e insere o texto [*tantas*] vezes ao deixar o modo de inserção.
Isto é normalmente feito como uma consequência de se observar muitas pessoas confusas com este comportamento no canal IRC do vim (#vim no Libera Chat).

9.1.2 Alternando modos

- *Modo Normal* permite que você insira comandos para navegar ou editar um documento, sendo o modo padrão. Você pode retornar à ele a partir de outro modo pressionando **Esc**.
- *Modo Visual* permite que você selecione texto em um documento. A maioria dos comandos do Modo Normal são também válidos neste modo. Você pode entrar nele pressionando **v** para selecionar caracteres ou **V** para selecionar linhas.
- *Modo Inserção* permite que você edite o documento diretamente. Você pode entrar nele pressionando **i** ou um dos vários comandos listados abaixo.
- O *Modo Comando* invoca a linha de comando do KatePart, permitindo que você execute muitos dos comandos disponíveis nas implementações do Vi bem como alguns específicos do KatePart. Para mais informações sobre estes comandos, veja Seção 5.2. Para usá-lo, pressione **:**, insira o comando e pressione **Enter**.

9.1.3 Integração com as funcionalidades do Kate

- O Modo Visual é ativado automaticamente quando o texto é selecionado com o mouse. Isto é também ativado ao usar funções do Kate que selecionam texto, como a Selecionar tudo (por meio do menu ou usando o **Ctrl+A**).
- As marcas do Vi e os [favoritos do Kate](#) são integrados. Quando uma marca é criada no Modo Vi, um favorito correspondente no Kate é criado e aparece no menu **Favoritos**. Do mesmo modo, quando um favorito do Kate é criado, uma marca correspondente do Vi na coluna 0 é também criada.

9.1.4 Comandos suportados nos modos normal/visual

a	Entra no Modo Inserção; adiciona após o cursor
A	Entra no Modo Inserção; adiciona após a linha
i	Entra no Modo Inserção; insere antes do cursor
Ins	Entra no Modo Inserção; insere antes do cursor
I	Entra no Modo Inserção; insere antes do primeiro caractere não-vazio da linha
gi	Entra no Modo Inserção; insere antes do cursor, deixando o último modo de inserção
v	Entra no Modo Visual; seleciona caracteres
V	Entra no Modo Visual; seleciona linhas
Ctrl+v	Entra no Modo Visual; seleciona blocos
gb	Entra no Modo Visual; seleciona novamente a última seleção
o	Abre uma nova linha abaixo da linha atual
O	Abre uma nova linha acima da linha atual
J	Junta linhas
c	Muda: seguido de um deslocamento para excluir e entrar no Modo Inserção
C	Muda até o final da linha: Exclui até o final da linha e entra no Modo Inserção
cc	Muda a linha: Exclui a linha e entra no Modo Inserção
s	Substitui caractere
S	Substitui linha
dd	Excluir linha
d	Seguido de deslocamento para excluir
D	Excluir até o fim da linha
x	Excluir caractere à direita do cursor
Del	Excluir caractere à direita do cursor
X	Excluir caractere à esquerda do cursor
gu	Seguido com um deslocamento para transformar em minúsculas
guu	Transforma a linha atual em minúsculas
gU	Seguido de um deslocamento para tornar maiúsculas
gUU	Transforma a linha atual em maiúsculas
y	Seguido de um deslocamento para copiar (do termo em inglês 'yank')
yy	Copia (do termo em inglês) a linha
Y	Copia (do termo em inglês) a linha
p	Colar após o cursor
P	Colar antes do cursor
lp	Colar após o cursor com recuo
[p	Colar antes do cursor com recuo
r	Seguido de um caractere para substituir o caractere após o cursor
R	Entra no modo Substituição

:	Entra no Modo Comando
/	Procurar
u	Desfazer
Ctrl+R	Refazer
U	Refazer
m	Definir marcador (pode ser usado por deslocamentos mais tarde)
n	Localizar próxima
N	Localizar anterior
>>	Recuar linha
<<	Remover recuo da linha
>	Recuar linhas
<	Remover recuos da linhas
Ctrl+F	Página abaixo
Ctrl+B	Página acima
ga	Imprimir o valor ASCII do caractere
.	Repetir última mudança
==	comandoAlinharLinha
=	comandoAlinharLinhas
~	Muda a caixa (maiúscula ou minúscula) do caractere atual
Ctrl+S	Dividir visão horizontalmente
Ctrl+V	Dividir visão verticalmente
Ctrl+W, w	Próxima divisão da janela
Ctrl+W, h CtrlW Esquerda	Ir para a divisão da janela à esquerda
Ctrl+W, l CtrlW Direita	Ir para a divisão da janela à direita
Ctrl+W, k CtrlW Acima	Ir para a divisão da janela acima
Ctrl+W, j CtrlW Abaixo	Ir para a divisão da janela abaixo

9.1.5 Movimentos suportados

As teclas a seguir podem ser usadas para mover-se em um documento no modo Normal ou Visual, ou em conjunção com um dos comandos acima. Elas podem ser precedidas por um número, que indica quantas vezes o referido movimento será feito.

h	Esquerda
Esquerda	Esquerda
Backspace	Esquerda
j	Abaixo
Abaixo	Abaixo
k	Acima
Acima	Acima
l	Direita
Direita	Direita
Espaço	Direita
\$	Fim da linha

End	Fim da linha
0	Primeiro caractere da linha (Coluna 0)
Home	Primeiro caractere da linha
^	Primeiro caractere não vazio da linha
f	Seguido pelo caractere para mover para direita do cursor
F	Seguido pelo caractere para mover para esquerda do cursor
t	Seguido pelo caractere para mover para direita do cursor, colocando o cursor no caractere antes dele
T	Seguido pelo caractere para mover para esquerda do cursor, colocando o cursor no caractere antes dele
gg	Primeira linha
G	Última linha
w	Próxima palavra
W	Próxima palavra separada por espaço em branco
b	Palavra anterior
B	Palavra anterior separada por espaço em branco
e	Fim da palavra
E	Fim da palavra separada por espaço em branco
ge	Fim da palavra anterior
gE	Fim da palavra anterior separada por espaço em branco
 	Seguido de um número de coluna para mover para esta coluna
%	Seguido por um item para mover para este item
`	Marca
``	Marca o primeiro caractere não em branco da linha
[[	Colchete de abertura anterior
]]	Próximo colchete de abertura
[]	Colchete de fechamento anterior
] [	Próximo colchete de fechamento
Ctrl+I	Saltar para a próxima localização
Ctrl+O	Saltar para a localização anterior
H	Ir para a primeira linha da tela
M	Ir para a linha do meio da tela
L	Ir para a última linha da tela
%porcentagem	Ir para a porcentagem especificada do documento
gk	Ir para a linha visualmente acima (quando usar a quebra de linha dinâmica)
gj	Ir para a linha visualmente abaixo (quando usar a quebra de linha dinâmica)
Ctrl+Esquerda	Move para uma palavra à esquerda
Ctrl+Direita	Move para uma palavra à direita

9.1.6 Objetos de texto suportados

Estes objetos podem ser usados para selecionar determinadas porções de um documento.

iw	Palavra interna: palavra incluindo espaços em branco
aw	Uma palavra: palavra excluindo espaços em branco
i"	Aspas duplas (") anterior até as próximas aspas duplas, incluindo marcas de citação
a"	Aspas duplas (") anterior até as próximas aspas duplas, excluindo marcas de citação
i'	Aspas simples (') anterior até as próximas aspas simples, incluindo marcas de citação
a'	Aspas simples (') anterior até as próximas aspas simples, excluindo marcas de citação
i (	Parênteses de abertura [(] anterior até o próximo parênteses de fechamento [)], incluindo os parênteses
a (	Parênteses de abertura [(] anterior até o próximo parênteses de fechamento [)], excluindo os parênteses
i [	Colchete de abertura [[] anterior até o próximo colchete de fechamento []], incluindo os colchetes
a [	Colchete de abertura [[] anterior até o próximo colchete de fechamento []], excluindo os colchetes
i {	Colchete de abertura [{ } anterior até o próximo colchete de fechamento [}], incluindo os colchetes
a {	Parênteses de abertura [{ } anterior até o próximo parênteses de fechamento [}], excluindo os parênteses
i <	Divisa de abertura (<) anterior até a próximo divisa de fechamento (>), incluindo as divisas
a <	Divisa de abertura (<) anterior até a próxima divisa de fechamento (>), excluindo as divisas
i `	Apóstrofo anterior (`) até o próximo apóstrofo, incluindo os apóstrofes
a `	Apóstrofo anterior (`) até o próximo apóstrofo, excluindo os apóstrofes

9.1.7 Comandos de modo de inserção suportados

Ctrl+D	Retirar recuo
Ctrl+T	Recuar

Ctrl+E	Inserir abaixo
Ctrl+Y	Apagar palavra
Ctrl+W	Apagar palavra
Ctrl+U	Excluir linha
Ctrl+J	Nova linha
Ctrl+H	Apaga o caractere atrás
Ctrl+Home	Move para o primeiro caractere no documento
Ctrl+R n	Inserir o conteúdo do registro n
Ctrl+O , <i>comando</i>	Entrar no modo normal somente para um comando
Ctrl+A	Incrementar o número atualmente selecionado
Ctrl+X	Decrementar o número atualmente selecionado

9.1.8 O Objeto de Texto Vírgula

Este objeto está faltando no Vim. O objeto de texto vírgula torna mais fácil modificar listas de parâmetros em linguagens no estilo C e outras listas separadas por vírgula. Isto é basicamente a área entre duas vírgulas ou entre uma vírgula e um colchete. Na linha mostrada na ilustração, os três intervalos que este objeto de texto pode ter são realçados.

```
int f(int arg1, double arg2, char arg3);
```

*Intervalos de objeto de texto de vírgula. Se o cursor está sobre o por exemplo `arg2`, pressionar **ci**, ('mudar vírgula interna') removeria `double arg2` e posicionaria o cursor entre as duas vírgulas no modo de inserção. Uma maneira muito conveniente de mudar os parâmetros da função.*

9.1.9 Funcionalidades que faltam

Como dito anteriormente, o objetivo do Modo VI do KatePart não é suportar 100% das funcionalidades do Vim.

Apêndice A

Expressões Regulares

Este Apêndice contém uma breve mas suficiente introdução sobre o mundo das *expressões regulares*. Ele documenta expressões regulares na forma disponível dentro do KatePart, que não é compatível com as expressões regulares do Perl, ou de outros comandos como o **grep**.

A.1 Introdução

As *Expressões Regulares* nos fornecem uma maneira de descrever alguns possíveis conteúdos de uma string de texto, de modo a ser entendido por um pedaço de software, e assim, ele pode investigar se um texto combina, e também no caso de aplicativos avançados, com o objetivo de salvar pedaços ou o texto que combina com a busca.

Um exemplo. Digamos que você deseja buscar, em um texto, por parágrafos que iniciam com um desses nomes: 'Henrik' ou 'Pernille', seguido por alguma forma do verbo 'dizer'.

Com uma pesquisa normal, você iria começar a procurar pelo primeiro nome, 'Henrique', sendo talvez seguido de 'di', como o seguinte: **Henrique di** e, enquanto procura por ocorrências, iria ignorar os que não se encontram no início de um parágrafo, assim como os que têm uma palavra que começa por 'di' que não é 'diz', 'disse' ou algo do gênero. Claro, e ainda ter que repetir isto com o próximo nome...

Com as Expressões Regulares, aquela tarefa poderia ser cumprida com uma única busca, e com um grau maior de precisão.

Para obter isto, as Expressões Regulares definem regras para expressar em detalhes uma generalização de uma string que será procurada. Podemos literalmente expressar nosso exemplo assim: 'Uma linha iniciada com 'Henrik' ou 'Pernille' (possivelmente seguido por 4 caracteres em branco ou tabs) seguido por um espaço em branco, seguido por um 'di' e então, um 'iz' ou 'sse', que poderia ser colocado na seguinte expressão regular:

```
^[ \t]{0,4}(Henrik|Pernille) di(sse|z)
```

O exemplo anterior demonstra todos os quatro conceitos principais das Expressões Regulares modernas, chamados:

- Padrões
- Afirmações
- Quantificadores
- Referências para trás

O acento circunflexo (^) no início da expressão é uma afirmação, sendo verdade somente se a string relacionada estiver no início de uma linha.

As strings `[ \t]` e `(Henrik|Pernille) di(sse|z)` são padrões. A primeira é uma *classe de caractere* que combina ou com um espaço em branco ou com um caractere de tabulação (horizontal); o outro padrão contém primeiramente um subpadrão que combina ou com `Henrik` ou com `Per nille`. Em seguida, vem um pedaço que corresponde exatamente com a string `di` e, finalmente segue um subpadrão que combina com `sse` ou com `z`.

A string `{0, 4}` é um quantificador dizendo ‘qualquer número de 0 até 4 da string anterior’.

Por causa do suporte ao conceito de *referência para trás* das expressões regulares, é economizado uma parte inteira combinada da string, bem como os subpadrões incluídos em parênteses; fornecidos alguns tipos de acesso para aquelas referências, podemos obter nossos meios de encontrar a string inteira (ao se procurar um documento de texto em um editor com uma expressão regular, o que é marcado frequentemente) ou o nome encontrado, ou a última parte do verbo.

Toda junta, a expressão procurará o que você deseja que ela procure, e somente ali.

As seções seguintes descreverão em detalhes como construir e usar padrões, classes de caracteres, afirmações, quantificadores e referências para trás, e a seção final fornecerá alguns exemplos úteis.

A.2 Padrões

Padrões consistem de strings literais e classes de caracteres, e podem conter subpadrões, que são padrões cercados por parênteses.

A.2.1 Escapando caracteres

Em padrões, bem como em classes de caracteres, alguns caracteres possuem um significado especial. Para combinar literalmente quaisquer um destes caracteres, eles devem ser marcados ou *escapados*, para deixar com que o software saiba que deve interpretar tais caracteres de maneira literal.

Isto é feito anexando no início do caractere uma barra invertida (\).

O software de expressão regular silenciosamente ignorará, escapando um caractere que não possui nenhum significado especial no contexto; assim, escapar, por exemplo, um ‘j’ (\j) é seguro. Se você tiver dúvidas se um caractere possui significado especial, você pode, portanto, escapá-lo seguramente.

A marcação inclui o próprio caractere de barra; por isso, para escapar uma barra, você deveria escrever \\.

A.2.2 Classes de Caracteres e abreviações

Uma *classe de caractere* é uma expressão que combina um dos caracteres de um conjunto definido. Em Expressões Regulares, as classes de caracteres são definidas colocando-se os caracteres legais em uma classe com colchetes ([]), ou usando uma das classes abreviadas descritas abaixo.

Classes de caracteres simples contém um ou mais caracteres literais, como por exemplo `[abc]` (combinando com uma das letras: ‘a’, ‘b’ ou ‘c’) ou `[0123456789]` (combinando com qualquer dígito).

Uma vez que as letras e os números possuem uma ordem lógica, você poderá abreviá-los, definindo para isso intervalos: `[a-c]` é igual a `[abc]` e `[0-9]` é igual a `[0123456789]`. Combinar estas opções, por exemplo `[a-fy not 1-38]` é perfeitamente possível (a última classe irá corresponder a um dos seguintes caracteres: ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘y’, ‘n’, ‘o’, ‘t’, ‘1’, ‘2’, ‘3’ ou ‘8’).

Como as letras maiúsculas são diferentes de minúsculas em expressões, para criar uma classe de caracteres que não diferenciase maiúsculas de minúsculas, combinando por exemplo com 'a' ou 'b', em qualquer caso, você precisaria escrever **[aAbB]**.

Obviamente, é possível criar uma classe 'negativa' que corresponda a 'tudo exceto'. Para fazer isso, coloque um acento circunflexo (^) no início da classe:

[^abc] combinará com qualquer caractere *menos com* 'a', 'b' ou 'c'.

Além disso, para caracteres literais, algumas abreviações são definidas, tornando a vida um pouco mais fácil:

\a

Isto combina com o caractere ASCII da campainha (BEL, 0x07).

\f

Isto combina com o caractere ASCII de nova página (FF, 0x0C).

\n

Isto combina com o caractere ASCII de nova linha (LF, 0x0A, nova linha do Unix).

\r

Isto combina com o caractere ASCII de retorno de carro (CR, 0x0D).

\t

Isto combina com o caractere ASCII de tabulação horizontal (HT, 0x09).

\v

Isto combina com o caractere ASCII de tabulação vertical (VT, 0x0B).

\xhhh

Isto combina com o caractere Unicode correspondente ao número hexadecimal hhhh (entre 0x0000 e 0xFFFF). \0ooo (isto é, \zero ooo) combina com o caractere ASCII/Latin-1 correspondente ao número octal ooo (entre 0 e 0377).

. (ponto)

Isto combina com qualquer caractere (incluindo nova linha).

\d

Isto combina com um dígito. É igual a **[0-9]**

\D

Isto combina com um caractere que não seja dígito. Igual a **[^0-9]** ou **[^\d]**

\s

Isto combina com um caractere em branco. Praticamente igual a **[\t\n\r]**

\S

Combina com o que não seja espaço em branco. Praticamente igual a **[\t\r\n]**, e igual a **[^\s]**

\w

Combina com quaisquer 'caractere da palavra', neste caso quaisquer letras, dígitos ou sublinhado. Igual a **[a-zA-Z0-9]**

\W

Combina com qualquer caractere diferente de palavra - qualquer coisa menos letras, números ou sublinhado. Igual a **[^a-zA-Z0-9]** ou **[^\w]**

A notação POSIX de classes, **[<class name>:]** também é suportada. Por exemplo, **[:digit:]** é equivalente a **\d**, e **[:space:]** a **\s**. Veja a lista completa de classes de caracteres POSIX [aqui](#).

As classes abreviadas pode ser colocadas dentro de classes personalizadas, como por exemplo: para combinar um caractere de palavra, um em branco ou um ponto, você poderia escrever **[\w\s.]**

A.2.2.1 Caracteres com significado especial dentro das classes

Os seguintes caracteres possuem um significado especial dentro da construção ‘[]’ de classes de caractere, e deve ser escapado para ser literalmente incluído em uma classe:

]

Finaliza a classe. Deve ser escapado, a menos que este caractere seja o primeiro da classe (pode ser seguido por um circunflexo não-escapado).

^ (circunflexo)

Denota uma classe negativa, se for o primeiro caractere. Deve ser escapado para combinar literalmente, se for o primeiro caractere da classe.

– (hífen)

Denota um intervalo lógico. Deve sempre ser escapado dentro de uma classe.

\ (barra invertida)

O caractere de escape. Deve sempre ser escapado.

A.2.3 Alternativas: combinando ‘um deles’

Se você deseja combinar um de um conjunto de padrões alternativos, pode separá-los com o caractere de barra vertical: |

Por exemplo: para encontrar a palavra ‘John’ ou a palavra ‘Harry’, você usaria uma expressão do tipo **John|Harry**.

A.2.4 Subpadrões

Subpadrões são padrões fechados entre parênteses, e possuem vários usos no mundo das expressões regulares.

A.2.4.1 Especificando alternativas

Você pode usar subpadrões para agrupar um conjunto de alternativas dentro de um padrão maior. As alternativas são separadas pelo caractere ‘|’ (barra vertical).

Por exemplo: Para procurar uma das palavras ‘int’, ‘float’ ou ‘double’, você poderia usar o padrão **int|float|double**. Se você deseja somente encontrar delas, se for seguida por algum espaço em branco e então algumas letras, coloque as alternativas dentro de um subpadrão: **(int|float|double)\s+\w+**.

A.2.4.2 Capturando texto relacionado (referências anteriores)

Se você quiser usar uma referência anterior, use um subpadrão **(PADRÃO)** para que a parte desejada do padrão seja lembrada. Para impedir que o subpadrão seja lembrado, use um grupo não capturador **(?:PADRÃO)**.

Por exemplo: Se você deseja encontrar duas ocorrências da mesma palavra, separadas por vírgula, e possivelmente algum espaço em branco, poderia escrever: **(\w+),\s*\1**. O subpadrão **\w+** procuraria pelo pedaço dos caracteres da palavra, e a expressão inteira combinaria se aquelas forem separadas por uma vírgula, 0 ou mais espaços em branco, e então, um pedaço igual de caracteres da palavra (a string **\1** referencia o primeiro subpadrão entre os parênteses).

NOTA

Para evitar ambiguidades com o uso de `\1` com alguns dígitos depois dele (por exemplo, `\12` pode ser o 12º subpadrão ou apenas o primeiro subpadrão com 2), usamos `\{12}` como sintaxe para subpadrões com vários dígitos.

Exemplos:

- `\{12}1` é 'usar subpadrão 12'
- `\123` é 'usar captura 1 e então 23 como texto normal'

A.2.4.3 Afirmações Adiante

Uma afirmação do tipo adiante é um subpadrão, iniciando com `?=` ou `?!`.

Por exemplo: para combinar a string literal 'Bill', mas somente seguida por 'Gates', você poderia usar esta expressão: **Bill(?! Gates)** (isto procuraria por 'Bill Clinton', bem como por 'Billy the kid', mas ele silenciosamente ignoraria as outras combinações).

Os subpadrões usados para as afirmações não são capturados.

Veja também em [Afirmações](#).

A.2.4.4 Afirmações atrás

Uma afirmação atrás é um subpadrão, iniciando com `?<=` ou `?<!`.

Atrás tem o mesmo efeito que a adiante, mas funciona de trás para frente. Por exemplo, para corresponder à string literal 'fruit' mas somente se não for precedida por 'grape', você poderia usar esta expressão: **(?<!grape) fruit**.

Os subpadrões usados para as afirmações não são capturados.

Veja também em [Afirmações](#).

A.2.5 Caracteres com significado especial dentro de padrões

Os seguintes caracteres possuem um significado especial dentro de um padrão, e devem ser escapados, caso você deseje procurá-los literalmente:

`\` (barra invertida)

O caractere de escape.

`^` (circunflexo)

Combina com o início da string.

`$`

Combina com o fim da string.

`()` (parênteses esquerdo e direito)

Denota subpadrões.

`{ }` (abre e fecha chaves)

Denota quantificadores numéricos.

`[ ]` (abre e fecha colchetes)

Denota classes de caracteres.

| (barra vertical)

OR lógico. Alternativas separadas.

+ (sinal de mais)

Quantificador, 1 ou mais.

*** (asterisco)**

Quantificador, 0 ou mais.

? (ponto de interrogação)

Um caractere opcional. Pode ser interpretado como um quantificador, 0 ou 1.

A.3 Quantificadores

Quantificadores permitem que uma expressão regular combine dentro de um número especificado ou de um intervalo de números, com um caractere, uma classe de caracteres ou um subpadrão.

Os quantificadores estão dentro de chaves (`{ e }`), e possuem como forma geral: `{[mínimo-ocorrências][, [máximo-ocorrências]]}`

O uso é melhor explicado através do exemplo:

{1}

Exatamente 1 ocorrência

{0,1}

Zero ou 1 ocorrência

{,1}

O mesmo, mas com menos trabalho ;)

{5,10}

No mínimo 5 e no máximo 10 ocorrências.

{5, }

No mínimo 5 ocorrências, sem máximo.

Adicionalmente, existe algumas abreviações:

*** (asterisco)**

igual a `{0, }`, procura por quaisquer número de ocorrências.

+ (sinal de mais)

similar a `{1, }`, no mínimo 1 ocorrência.

? (ponto de interrogação)

similar a `{0,1}`, zero ou 1 ocorrência.

A.3.1 Guloso

Ao usar os quantificadores sem máximo, as expressões regulares padronizam a combinação para o mais próximo da string procurada possível, comumente conhecido como comportamento *guloso*.

O software moderno de expressões regulares fornece o chamado ‘desligamento’ do modo ‘guloso’, embora em um ambiente gráfico a interface fornece a você o acesso a isto. Por exemplo, um diálogo de busca, fornecendo a busca para uma expressão regular, poderia ter uma caixa chamada ‘Combinação Mínima’, bem como indicar se o ‘modo guloso’ é o comportamento padrão.

A.3.2 Exemplos do Contexto

Aqui estão alguns exemplos do uso de quantificadores:

`^\d{4,5}\s`

Combina com os dígitos em '1234 ir' e '12345 agora,' mas não combina com '567 sete' nem com '223459 algum local'.

`\s+`

Combina uma ou mais vezes com caracteres de espaço em branco.

`(bla){1,}`

Combina com todos os 'blablabla', e com 'bla' em 'blackbird' ou 'tabla'.

`/?>`

Combina com '/>' em '<fecharitem/>' bem como com '>' em '<abriritem>'.

A.4 Afirmações

Afirmações permitem que uma expressão regular combine com certas condições controladas.

Uma afirmação não precisa de um caractere para combinar, e em vez disso ela investiga as adjacências de uma possível combinação, antes de conhecê-la. Por exemplo: a afirmação *limite da palavra* não tenta encontrar um caractere que não seja da palavra, oposto de uma palavra em sua posição, e ao invés disso, ele certifica-se de que não existe um caractere da palavra. Isto significa que a afirmação pode combinar onde não existe caractere, isto é, no final de uma string procurada.

Algumas verificações, de fato, possuem um padrão de correspondência, mas a parte do texto que correspondeu não será uma parte do resultado da correspondência, em relação à expressão completa.

As Expressões Regulares, como documentadas aqui, suportam as seguintes afirmações:

`^` (circunflexo: início de string)

Combina com o início da string procurada.

A expressão **`^Peter`** combinará com 'Peter' na string 'Peter, ei!', mas não com 'Ei, Peter!'

`$` (final da string)

Combina com o fim da string procurada.

A expressão **`você\?$`** combinará com o 'você' no final da string 'É isto que deseja para você?', mas não combinará com 'É isto que deseja para você, certo?'

`\b` (limite da palavra)

Combina se existir um caractere da palavra, mas não combina com o caractere de outra.

Isto é útil para encontrar finais de palavras, como por exemplo, finais para encontrar uma expressão inteira. A expressão **`\bem\b`** procurará separado por 'em' na string 'Ele estava em prantos no embarque', mas não combinará, por exemplo, com o 'em' de 'embarque'.

`\B` (sem limite de palavra)

Combina com tudo que '\b' não combina.

Isto significa que ele combinará com string dentro de palavras. A expressão **`\Em\B`** combinará com 'embarque' mas não com 'Ele estava em prantos'.

(?=PADRÃO) (adiante positivo)

Uma afirmação adiante olha na parte da string seguinte, como uma possível combinação. O 'adiante positivo' prevenirá a string de combinação, se o texto seguinte não combina com o *PADRÃO* da afirmação, mas o texto combinado não será incluído no resultado.

A expressão **carta(=?\w)** combinará com 'carta' em 'cartada', mas não em 'O jogador deu uma cartada!'

(?!PADRÃO) (adiante negativo)

O 'adiante negativo' previne que uma possível combinação seja de conhecimento, se a seguinte parte da string buscada não combina com seu *PADRÃO*.

A expressão **const \w+\b(?!s*&)** combinará com 'const char' na string 'const char* foo' enquanto que não combinará 'const QString' em 'const QString& bar', pois o '&' combina com o padrão de afirmação negativo.

(?<=PADRÃO) (atrás positivo)

Retrospectiva tem o mesmo efeito que o adiante, mas funciona de trás para frente. Retrospectiva examina a parte da string anterior a uma possível correspondência. Retrospectiva positiva corresponderá a uma string somente se ela for precedida pelo *PADRÃO* da asserção, mas o texto correspondido por esse *PADRÃO* não será incluído no resultado.

A expressão **(?<=cup) cake** corresponderá a 'cake' se for seguida por 'cup' (em 'cupcake' mas não em 'cheesecake' ou em 'cake' sozinho).

(?<!PADRÃO) (retrospectiva negativa)

A 'atrás negativo' previne que uma possível combinação seja de conhecimento, se a seguinte parte da string buscada não combina com seu *PADRÃO*.

A expressão **(?<![\w\.]) [0-9]+** corresponderá em '123' nas strings '=123' e '-123' enquanto não pode corresponder a '123' em '. 123' ou 'word123'.

(PADRÃO) (grupo de captura)

O subpadrão dentro dos parênteses é capturado e memorizado, para que possa ser usado em referências anteriores. Por exemplo, a expressão **("+)[^"]*\1** corresponde a **""texto""** e **"texto"**.

Consulte a seção [Captura de texto correspondente \(referências anteriores\)](#) para obter mais informações.

(?:PADRÃO) (grupo não-captura)

O subpadrão dentro dos parênteses não é capturado e não é lembrado. É preferível usar sempre grupos não capturantes se as capturas não forem usadas.

Apêndice B

Índice

C
comentar, [33](#)

D
descomentar, [33](#)

S
substituir no estilo sed
procurar no estilo sed, [37](#)