

Anterior:[Aplicando personalizações](#), Acima:[Customização](#) [[Conteúdo](#)][[Índice](#)]

## 15.6 Temas personalizados

*Temas personalizados* são coleções de configurações que podem ser habilitadas ou desabilitadas como uma unidade. Veja [Temas Personalizados](#) no Manual do GNU Emacs . Cada tema personalizado é definido por um arquivo fonte do Emacs Lisp, que deve seguir as convenções descritas nesta seção. (Em vez de escrever um tema personalizado à mão, você também pode criar um usando uma interface do tipo Personalizar; consulte [Criando temas personalizados](#) no Manual do GNU Emacs .)

Um arquivo de tema personalizado deve ser nomeado *foo* - theme . el, onde *foo* é o nome do tema. O primeiro formulário Lisp no arquivo deve ser uma chamada para `deftheme`, e o último formulário deve ser uma chamada para `provide-theme`.

### Macro: tema `deftheme` & documento opcional

Esta macro declara o *tema* (um símbolo) como o nome de um tema personalizado. O argumento opcional *doc* deve ser uma string descrevendo o tema; esta é a descrição mostrada quando o usuário invoca o `describe-themes` comando ou digita ?no '\*Temas Personalizados\*' amortecedor.

Dois nomes de temas especiais não são permitidos (usá-los causa um erro): `user` é um tema fictício que armazena as configurações de personalização diretas do usuário e `changed` é um tema fictício que armazena alterações feitas fora do sistema Personalizar.

### Macro: tema fornecer- *tema*

Essa macro declara que o tema chamado *theme* foi totalmente especificado.

Entre `deftheme` e `provide-theme` estão os formulários Lisp que especificam as configurações do tema: geralmente uma chamada para `custom-theme-set-variable` e/ou uma chamada para `custom-theme-set-faces`.

### Função: tema `custom-theme-set-variables` *theme* & rest args

Esta função especifica as configurações de variáveis do *tema de tema personalizado*. *tema* deve ser um símbolo. Cada argumento em *args* deve ser uma lista da forma

```
( expressão var [ agora [ solicitação [ comentário ] ] ] )
```

onde as entradas da lista têm os mesmos significados que em `custom-set-variables`. Consulte [Aplicação de personalizações](#) .

### Função: tema de rostos de conjunto de temas personalizados e argumentos de descanso

Esta função especifica as configurações de face do *tema de tema personalizado*. *tema* deve ser um símbolo. Cada argumento em *args* deve ser uma lista da forma

```
( especificação de rosto [ agora [ comentário ] ] )
```

onde as entradas da lista têm os mesmos significados que em `custom-set-faces`. Consulte [Aplicação de personalizações](#) .

Em teoria, um arquivo de tema também pode conter outros formulários Lisp, que seriam avaliados ao carregar o tema, mas isso não é bom. Para proteger contra o carregamento de temas contendo código malicioso, o Emacs exibe o arquivo de origem e solicita a confirmação do usuário antes de carregar qualquer tema não integrado pela primeira vez. Como tal, os temas normalmente não são compilados por byte, e os arquivos de origem sempre têm precedência quando o Emacs está procurando um tema para carregar.

As seguintes funções são úteis para habilitar e desabilitar temas programaticamente:

**Função: tema custom-theme-*p***

Esta função retorna um não nil valor se *theme* (um símbolo) for o nome de um tema personalizado (ou seja, um tema personalizado que foi carregado no Emacs, esteja o tema ativado ou não). Caso contrário, ele retorna nil.

**Variável: temas conhecidos personalizados**

O valor desta variável é uma lista de temas carregados no Emacs. Cada tema é representado por um símbolo Lisp (o nome do tema). O valor padrão desta variável é uma lista contendo dois temas fictícios: (user changed). O changedtema armazena as configurações feitas antes de qualquer tema personalizado ser aplicado (por exemplo, variáveis definidas fora de Personalizar). O usertema armazena as configurações que o usuário personalizou e salvou. Quaisquer temas adicionais declarados com a defthememacro são adicionados à frente desta lista.

**Comando: load-theme *theme* &opcional no-confirm no-enable**

Esta função carrega o tema personalizado chamado *tema* de seu arquivo de origem, procurando o arquivo de origem nos diretórios especificados pela variável custom-theme-load-path. Veja [Temas Personalizados](#) no Manual do GNU Emacs . Ele também *habilita* o tema (a menos que o argumento opcional *no-enable* seja non- nil), fazendo com que suas configurações de variável e face tenham efeito. Ele solicita a confirmação do usuário antes de carregar o tema, a menos que o argumento opcional *no-confirm* seja non- nil.

**Comando: enable-theme *theme***

Esta função habilita o tema personalizado chamado *theme* . Ele sinaliza um erro se nenhum tema foi carregado.

**Comando: desativar tema do *tema***

Esta função desativa o tema personalizado chamado *theme* . O tema permanece carregado, de modo que uma chamada subsequente para enable-themeativá-lo.

Anterior: [Aplicando personalizações](#), Acima: [Costumização](#) [[Conteúdo](#)][[Índice](#)]