

Próximo: [Tópicos relacionados](#), Anterior: [Declarando Funções](#), Acima: [Funções](#) [[Conteúdo](#)][[Índice](#)]

## 13.16 Determinando se uma função é segura para chamar

Alguns modos principais, como SES, chamam funções que são armazenadas em arquivos de usuário. (Consulte [\(ses\)Topo](#) , para obter mais informações sobre SES.) Os arquivos de usuários às vezes têm pedigrees ruins - você pode obter uma planilha de alguém que acabou de conhecer ou pode obter uma por e-mail de alguém que nunca conheceu. Portanto, é arriscado chamar uma função cujo código-fonte está armazenado em um arquivo de usuário até que você determine que ela é segura.

### **Função: formulário *unsafep* e *unsafep-vars* opcionais**

Retorna `nil` se *form* for uma expressão Lisp *segura* ou retorna uma lista que descreve por que ela pode não ser segura. O argumento *unsafep-vars* é uma lista de símbolos conhecidos por terem ligações temporárias neste ponto; é usado principalmente para chamadas recursivas internas. O buffer atual é um argumento implícito, que fornece uma lista de ligações de buffer-local.

Sendo rápido e simples, *unsafep* faz uma análise muito leve e rejeita muitas expressões Lisp que são realmente seguras. Não há casos conhecidos em que *unsafep* retorna `nil` para uma expressão não segura. No entanto, uma expressão Lisp segura pode retornar uma string com uma *display* propriedade, contendo uma expressão Lisp associada a ser executada após a string ser inserida em um buffer. Esta expressão associada pode ser um vírus. Para estar seguro, você deve excluir as propriedades de todas as strings calculadas pelo código do usuário antes de inseri-las nos buffers.