

# Diário de estudos CLI

## Resumo

Agora que os conceitos mais básicos de programação estão mais consolidados e você já praticou a lógica de programação, vamos avançar um pouco nos conteúdos e começar a ver como podemos usar Ruby para criar aplicações. Inicialmente vamos criar uma ferramenta CLI ([1](#), [2](#)) (Interface de linha de comandos) para armazenar uma lista de objetivos de estudo.

Se você ainda não leu as apostilas ou assistiu os vídeos de conteúdos disponibilizados, mais uma vez recomendamos estudar esse material antes de começar.

```
~/study-diary-terminal 15:29 10/11
```



Para ver a animação no seu navegador, clique [aqui](#).

## Como você vai trabalhar

Crie um diretório no qual você vai trabalhar e dentro dele crie um arquivo com o nome que desejar, por exemplo `study_diary.rb`. No Terminal, dentro do diretório de trabalho que você criou anteriormente, execute o comando `ruby study_diary.rb` para rodar o código do arquivo.

Dentro do arquivo `study_diary.rb` você vai implementar o código da sua aplicação. Diferentemente dos desafios no Code Saga, aqui você vai fazer uso dos comandos `puts` e `gets` para imprimir informações em tela e adquirir dados de *input* de usuário.

## Escopo da Aplicação e Funcionalidades

Assim que a aplicação começar a rodar, devem aparecer em tela algumas opções de uso:

```
[1] Cadastrar um item para estudar
[2] Ver todos os itens cadastrados
[3] Buscar um item de estudo
[4] Sair
Escolha uma opção:
```

Depois de imprimir essa lista, o Terminal deve aguardar o *input* de usuário com um dos números indicados. A opção “Cadastrar um item para estudar” cria um item de estudo, “Ver todos os itens cadastrados”, lista os itens que já foram cadastrados em memória, e “Sair”, fecha a execução do programa.

Quando a opção de cadastro de item for selecionada, o programa deve solicitar o título do item e em seguida uma categoria deve ser atribuída a ela. Para essa funcionalidade é obrigatório utilizar **Orientação a Objetos**, assim, cada item deve ser representado por um novo objeto da Classe `StudyItem`, e cada categoria é representada por um objeto da Classe `Category`. Todo item de estudo deve conter uma categoria. As categorias podem estar previamente criadas dentro da aplicação, ou seja, não é necessário criar uma opção de criação de categorias.

Após a criação de uma categoria, uma mensagem de sucesso pode ser impressa em tela com o título do item e sua categoria. Em seguida, o usuário deve poder escolher novamente uma das opções do menu inicial, ou seja, o programa continua funcionando até que o usuário escolha a opção “Sair”.

Na opção “Buscar um item de estudo”, o usuário pode pesquisar por uma palavra e a aplicação imprime em tela todos os itens de estudo cadastrados que contenham o termo em seu título.

Esse é o escopo mais básico da aplicação. A seguir vamos listar algumas funcionalidades adicionais que podem ser realizadas. É recomendado que pelo menos duas das funcionalidades abaixo sejam implementadas, mas acreditamos que você seja capaz de fazer todas elas.

## Funcionalidades adicionais

### Listar por categoria

Um usuário pode listar todos os itens cadastrados pertencentes a uma determinada categoria.

### Apagar um item

Um usuário pode apagar um item cadastrado na aplicação.

### Descrição de um item e pesquisa

Um item pode conter uma descrição, além do título e de uma categoria. A pesquisa por item deve avaliar a presença do termo da busca no título e na descrição.

### Marcar um item como concluído

Um item pode ser marcado como concluído, quando você terminar de estudá-lo. Assim ele não aparece nas listas e buscas. Deve existir uma opção para listar os itens concluídos.

### Guardar os itens ao finalizar o programa

Ao finalizar o programa, as informações que foram inseridas devem ser guardadas e carregadas na próxima execução. Você pode começar salvando as informações em um arquivo, o que seria a forma mais simples. Depois, se quiser se desafiar, pode tentar implementar uma estrutura de banco de dados.

## Links úteis

- <https://ruby-doc.org/>
- <https://campuscode.com.br/conteudos/ruby-puro-com-banco-de-dados-sqlite>
- <https://ruby-doc.org/core/File.html>