

03

Trabalhando com as variáveis

Inicialmente criamos uma variável `precep`, contendo os 2 primeiros caracteres do CEP, e depois criamos a variável `precep2` que convertia o `precep` em número, com o código abaixo:

```
DATA teste1;
  set alura.cadastro_cliente;

  precep = substr(cep,1,2);
  precep2 = input(precep,best.);

RUN;
```

Mas não queríamos manter a variável de texto na base pois era apenas uma variável intermediária. Na aula resolvemos isso reescrevendo o código, "aninhando" o comando que criava a variável `precep` dentro do comando `input`. Mas não precisaríamos resolver o problema desta forma.

Abaixo temos um outro código baseado no *data step* acima. Como é a base resultante de sua execução?

```
DATA teste1;
  set alura.cadastro_cliente (obs=max);

  precep = substr(cep,1,2);
  precep2 = input(precep,best.);

  drop precep;
  rename precep2 = precep;

RUN;
```

Selezione uma alternativa

A É uma base com menos observações (linhas) que a base `cadastro_cliente`, mas que contém todas as suas variáveis mais uma variável que criamos chamada `precep`, que contém o início do CEP em formato numérico.

B É uma base com o mesmo número de linhas e todas as variáveis da base `cadastro_cliente`, mais uma variável que criamos chamada `precep`, que contém o início do CEP em formato numérico.

C É uma base com o mesmo número de linhas que a base `cadastro_cliente`, mas que contém apenas uma variável, chamada `precep`, que contém o início do CEP em formato numérico.

D É uma base com o mesmo número de linhas e todas as variáveis da base `cadastro_cliente`, mais uma variável que criamos chamada `precep`, que contém o início do CEP em formato de texto (caractere).

