

bar_history é uma lista

```
class Class
  def attr_accessor_with_history(attr_name)
    attr_name = attr_name.to_s # Garantir que o nome do atributo seja uma string
    attr_reader attr_name # Cria o getter para o atributo
    attr_reader "#{attr_name}_history" # Cria o getter para o histórico do atributo

    class_eval %Q{
      def #{attr_name} = (value)
        # Inicializa o histórico com [nil] se ainda não foi definido
        @#{attr_name}_history = [nil] unless @#{attr_name}_history
        # Adiciona o valor ao histórico
        @#{attr_name}_history << value
        # Define o novo valor para o atributo
        @#{attr_name} = value
      end
    }
  end
end
```

```
class Foo
  attr_accessor_with_history :bar
end

# Testando
f = Foo.new
f.bar = 1
f.bar = 2
puts f.bar_history.inspect # Deve retornar [nil, 1, 2]
```

sempre haverá nil, pois Foo não tem inicializador.

- primeiramente é necessário criar um método adicional na classe de classe (`attr_accessor_with_history`)
- o getter e o setter de `:bar` é diferente, pois ele é dinâmico, ou seja, ele poderia ter um nome diferente ou poderia haver outros atributos também