

## TRABALHO 15/10/2025

```
(ns notas.core (:require [clojure.string :as str]))
```

```
(defn ler-int [prompt]
  (loop []
    (print prompt) (flush)
    (let [s (read-line)]
      (try
        (Integer/parseInt (str/trim s))
        (catch Exception _
          (println "Valor inválido. Tente novamente.")
          (recur)))))))
```

```
(defn ler-nota []
  (loop []
    (print "Nota (0–100): ") (flush)
    (let [s (read-line)]
      (try
        (let [v (Double/parseDouble (str/trim s))]
          (if (and (<= 0 v) (<= v 100))
            v
            (do (println "A nota deve estar entre 0 e 100.") (recur))))
        (catch Exception _
          (println "Valor inválido. Digite um número, ex: 75 ou 92.5.")
          (recur)))))))
```

```
(defn conceito [nota]
```

```
(cond
```

```
(>= nota 90) "A"
```

```
(>= nota 80) "B"
```

```
(>= nota 70) "C"
```

```
(>= nota 60) "D"
```

```
:else      "F"))
```

```
(defn -main [& _]
```

```
(let [qtd (max 1 (ler-int "Quantos alunos na turma? "))]
```

```
(println)
```

```
(loop [i 1, soma 0.0, aprov 0]
```

```
(if (> i qtd)
```

```
(let [media (/ soma qtd)
```

```
desempenho (if (>= media 80) "Turma excelente!"
```

```
(if (>= media 60) "Bom desempenho!"
```

```
"É necessário melhorar!"))]
```

```
(println (format "\nMédia da turma: %.1f" media))
```

```
(println (str "Aprovados: " aprov))
```

```
(println (str "Desempenho geral: " desempenho)))
```

```
(do
```

```
(print (format "Nome do aluno %d: " i)) (flush)
```

```
(let [nome (or (str/trim (read-line)) (str "Aluno-" i))
```

```
nota (ler-nota)
```

```
conc (conceito nota)
```

```
aprov? (>= nota 60)]
```

```
(println (format "%s - Conceito: %s" (if (empty? nome) (str "Aluno-" i) nome) conc))
```

```
(println)
```

```
(recur (inc i)
```

```
(+ soma nota)
```

```
(if aprov? (inc aprov) aprov))))))
```