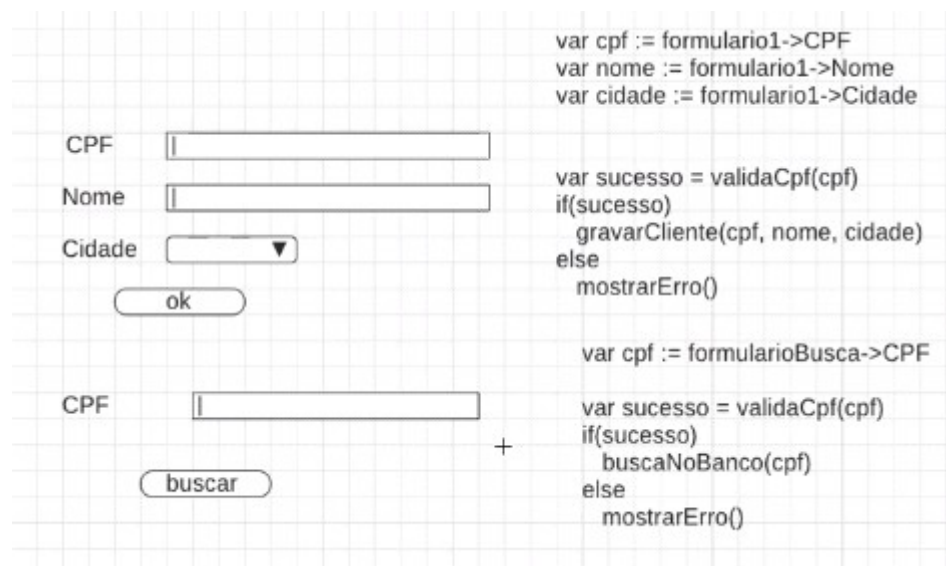


## 1 - problema do procedural

Aula explicando sobre a importância de criar um programa que possibilite a fácil manutenção. De maneira mais prática, a orientação a objetos possibilita que qualquer programador consiga realizar alterações no programa sem ter participado no início de seu desenvolvimento.

Dado e funcionalidade sobre ele andam juntos.

"Para que várias equipes consigam trabalhar em um mesmo projeto, é necessário que as responsabilidades de cada código estejam bem definidas e claras, evitando conflitos na hora de realizar mudanças e evoluções. Código com responsabilidades coesas é sinal do paradigma OO."



Nessa aula conhecemos o paradigma procedural, que era utilizado como prática de programação antes da introdução das linguagens orientadas a objetos.

Foi usada como exemplo a necessidade de validar o CPF em um formulário para discutirmos os principais problemas que podem aparecer neste paradigma.

Em especial, à medida que outros formulários e desenvolvedores precisam da mesma validação de CPF, não era fácil de perceber que já havia procedimentos e funções que faziam esse trabalho, pois dados e funções não possuíam um vínculo tão forte. Isso poderia resultar em mais uma nova função ou trecho de código com responsabilidade parecida.

Além disso, conhecemos a ideia central do paradigma da orientação a objetos, que é criar unidades de código que combinem os dados associados a determinada informação com as funcionalidades aplicadas

aqueles dados (por exemplo, CPF + validação). São os atributos e métodos.