

Veremos em §6.10.4 que você também pode expressar este objeto Copiar e-Substituir a operação usando o ... Spread Operator como este:

```
o = {... padrão, ... o};
```

Também poderíamos evitar a sobrecarga da criação extra de objetos e copiando escrevendo uma versão do `object.assign ()` que copia propriedades apenas se estiverem faltando:

```
// como object.assign (), mas não substitui
```

```
propriedades
```

```
// (e também não lida com propriedades de símbolo)
```

```
função mescla (alvo, ... fontes) {
```

```
  para (deixe a fonte de fontes) {
```

```
    para (deixe a chave do object.Keys (fonte)) {
```

```
      if (! (chave no alvo)) { // isso é diferente de
```

```
Object.assign ()
```

```
Target [key] = fonte [chave];
```

```
}
```

```
}
```

```
}
```

```
alvo de retorno;
```

```
}
```

```
Object.assign ({x: 1}, {x: 2, y: 2}, {y: 3, z: 4}) // => {x:  
2, y: 3, z: 4}
```

```
Merge ({x: 1}, {x: 2, y: 2}, {y: 3, z: 4}) // => {x:  
1, y: 2, z: 4}
```

É simples escrever outros utilitários de manipulação de propriedades, como esta função `mescla ()`. Uma função `RESTRITCI ()` pode excluir propriedades de um objeto se não aparecerem em outro objeto de modelo, por exemplo. Ou uma função `subtract ()` pode remover toda a propriedades de um objeto de outro objeto.