

Esto es una traducción de un post de Marin Abernethy en el blog dockyard.com. En modo de homenaje y agradecimiento. Con la intención de que la comunidad hispana pueda disfrutarlo en su idioma nativo.

En Elixir a menudo, encontramos que hay muchas formas de implementar una sola tarea. Por ejemplo, cuando queremos actualizar un valor clave en un mapa, tenemos varias opciones por delante...

Dado, `expenses = %{groceries: 200, rent: 1000, commute: 70}` podría emplear:

- Si deseamos actualizar varios pares de valores clave y / o varios nuevos.

```
# Sintaxis
Map.merge(map1, map2)
# Ejemplo
Map.merge(expenses, %{ rent: 1200, comcast: 100 })
```

- Si actualizamos o agregamos un solo valor de clave.

```
# Sintaxis
Map.put(map, key, val)
# Ejemplo
Map.put(expenses, :booze, 100)
```

- Si queremos incrementar un valor en cierto grado.

```
# Sintaxis
Map.update(map, key, initial, fun)
# Ejemplo
Map.update(expenses, :misc, 300, &(&1 * 2))
```

- Si queremos actualizar un valor en una estructura anidada.

```
# Sintaxis
Kernel.put_in(data, keys, value)
```



```
# Ejemplo
expenses = %{groceries: %{milk: 5}, apartment: %{rent: 1000, comcast: 100}}
put_in(expenses, [:rent, :comcast], "too much")
```

- Sin embargo, los mapas vienen con una sintaxis integrada para actualizar uno o más valores clave.

```
%{expenses | groceries: 150, commute: 75}
```