

# Fundamentos de programación

Estado de las variables

[carlos.andres.delgado@correounivalle.edu.co](mailto:carlos.andres.delgado@correounivalle.edu.co)

Carlos Andrés Delgado S.

Facultad de Ingeniería. Universidad del Valle

Noviembre de 2016

Fundamentos  
de  
programación

Carlos Andrés  
Delgado S.

Estado de las  
variables

Vectores

## 1 Estado de las variables

## 2 Vectores

Fundamentos  
de  
programación

Carlos Andrés  
Delgado S.

Estado de las  
variables

Vectores

## 1 Estado de las variables

## 2 Vectores

## Definición

Hasta el momento, una vez definimos un dato o una estructura este no puede cambiar en el tiempo. Sin embargo, en algunos casos es necesario, por ejemplo:

- Se requiere cambiar el valor de una variable, para llevar una cuenta por ejemplo
- Se requiere cambiar el valor de un campo en una estructura

Para esto, se va utilizar el operador **set**!

# Estado de las variables

Fundamentos  
de  
programación

Carlos Andrés  
Delgado S.

Estado de las  
variables

Vectores

## Definición

Ejemplo:

```
(define var 5)
(display var)
(set! var 8)
(display var)
(set! var 'hola)
(display var)
```

var 5

var 8

var 'hola

secuencialidad

Si observa nos imprime un void, esto se debe a que la operación `set!` no retorna ningún valor, es una operación de efecto, para esto introducimos `begin`

## Definición

begin, nos permite ejecutar más de una expresión y retornar un valor

```
( begin  exp-1 ←  
        exp-2 ←  
        exp-n ←  
        exp  
)
```

## Definición

El código anterior cambia a:

```
(define var 5)
(display var)
(begin
  (set! var 8)
  var
)
(begin
  (set! var 'hola)
  var
)
```

## Definición

Con esto, ya podemos hacer más de una operación en una misma función

```
(define c 3)
;funcion: numero -> numero
(define (funcion a)
  (begin
    (set! c (+ a a))
    (+ a 2)
  )
)
;;Esta función cambia el valor de c y retorna a + 2 (salida en drracket)
> c
3
> (funcion 4)
6
> c
8
>
```



## Definición

### Ejemplo con estructuras:

```
(define-struct empleado (nombre edad cargo))  
(define emp1 (make-empleado "carlos" 18 "gerente"))  
;; Debemos cambiar el nombre ya que nos quedó mal  
(begin  
  (set! emp1  
    (make-empleado  
      "Carlos"  
      (empleado-edad emp1)  
      (empleado-cargo emp1)  
    )  
  )  
  emp1  
)
```

Fundamentos  
de  
programación

Carlos Andrés  
Delgado S.

Estado de las  
variables

Vectores

1 Estado de las variables

2 Vectores

## Definición

Los vectores, son estructuras similares a las listas, pero presentan algunas diferencias:

- 1 Para cambiar un elemento, no es necesario reconstruir la lista (como lo sería usando set!)
- 2 Se pueden agregar o quitar elementos
- 3 Los vectores pueden almacenar cualquier tipo de dato (numero, texto, lista, estructura u otro vector)

## Definición

### Para operar vectores

```
;; Crear vectores
; Crear un vector con 5 ceros
(make-vector 5)
; Crear un vector a partir de una lista
(list->vector (list 1 2 3 4 5))
; Cambiar un elemento de un vector
(define vector1 (list->vector (list 1 2 3 4 5)))
; Cambiar la tercera posición por 45 (Están numeradas
  desde 0)
X (vector-set! vector1 2 45)
; Obtener la primera posición del vector
X (vector-ref vector1 0)
```



Universidad  
del Valle

# ¿Preguntas?

Fundamentos  
de  
programación

Carlos Andrés  
Delgado S.

Estado de las  
variables

Vectores

# VAMO A

