```
#lang racket

(define (vectores)
  (define vector (make-vector 10 0))
  ; make-vector CREA UN VECTOR de tamaño 10 con ceros
  (ver vector 0) ;esta función muestra el contenido del vector tras haberlo
creado
  (newline)
  (llenar vector 0) ; esta función ingresa datos al vector
  (newline)
  (ver vector 0) ;esta función muestra el contenido del vector tras ingresarle
datos
)

(define (ver vector i)
  (if (< i 10)
    (begin
      (display (vector-ref vector i))    ; vector-ref referencia el contenido de una
posición de un vector
      (ver vector (+ i 1))
    )
   (display "")
  )
)

(define (llenar vector i)
  (if (< i  10)
      (begin
        (vector-set! vector i (* (+ i 1) (+ i 1)))    ; vector-set! asigna un valor o
almacena un valor en una posición de un vector
        (llenar vector (+ i 1))
       )
      (display "")
   )
)

(vectores)
```

```
#lang racket

(define (vectores1)
  (define EDADES (vector 0 0 0 0 0 0))  ; crear vector
  (define NOMBRES (vector 0 0 0 0 0 0))
       ;  (vector-set! EDADES 0 55)   ;almacenaría un dato en un vector por
programa
  (display "NOMBRE?: ")
  (vector-set! NOMBRES 0 (read))   ;almacena en un vector por lectura
  (newline)
  (display "EDAD?: ")
  (vector-set! EDADES 0 (read))   ;almacena en un vector por lectura
  (newline)

  (display "SU NOMBRE ES: ")
  (display (vector-ref NOMBRES 0))          ;; muestra contenido de vector
  (display "\nSU EDAD ES: ")
  (display (vector-ref EDADES 0))          ;; muestra contenido de vector
  (newline)

  (if (> (vector-ref EDADES 0) 50) ;evalua contenido de vector
    (display "LLEVADO...")
    (display "TODAVIA AGUANTA...")
   )
)


(vectores1)
```

```
#lang racket

; función que eleva al cuadrado el parámetro recibido
(define (elevar y)
  (* y y)
)

;función que muestra mensajes y llama a otra función
(define (elevar2 y)
  (begin
    (display "El resultado de elevar ")
    (display y)
    (display " al cuadrado es = ")
    (display (elevar y))
    (display "\n\n")
  )
)

;La función mostrar Menu despliega en pantalla una lista de opciones
(define (mostrarMenu )
  (begin
    (display "MENU PRINCIPAL DE OPCIONES \n\n")
    (display "1. elevar al cuadrado \n")
    (display "2. En construcción… \n")
    (display "3. En construcción… \n")
    (display "0. salir \n\n")
    (display "escoja opcion: " )
  )
)

;La función leerOpcion x, recibe un parámetro que representa la opción escogida, la función evalúa la opción
;para ejecutar el código correspondiente.
(define (leerOpcion x)
  (begin
    (cond
      ((= x 1) (begin
                 (display "Digite un numero: ")
                   (elevar2 (read))  ;llamado a una función
               )
      )
      ((= x 2) (begin
                 ; llamado a funciones u otras operaciones
               )
      )
      ((= x 3) (begin
                 ; llamado a funciones u otras operaciones
               )
      )
      ((= x 0) (display "adios \n"))
    )
    (display "\n\n")
    (if (not(= x 0))
        (menu)     ;llamado a una función
    )
  )
)
;A través de la función menu se ejecuta el programa
(define (menu)
  (begin
    (mostrarMenu)  ;llamado a una función
    (leerOpcion (read))    ;llamado a una función
  )
)
(menu)
```