



**El saber de mis hijos  
hará mi grandeza**

**Universidad de Sonora.**

**Lenguajes de programación.**

**Tarea:**

**Tarea-00**

**Leonardo Brambila Ayala.**

**Lic. Ciencia de la Computación.**

**23 de agosto del 2023.**

## Trabajos revisados de: Aldo Rafael Guzmán Villanueva.

### Programas de la página 81 del PDF:

Si la variable es declarada en el bloque actual, usamos esa declaración

```
#lang stacker/smol/fun
```

```
(deffun (f x)
```

```
  (defvar y 1)
```

```
  (- x y))
```

```
(f 0)
```

Como las variables  $x$  e  $y$ , son las únicas declaradas con esos nombres en el bloque actual se usarán en la operación de resta, en conclusión, esta bien, ya que esta declarada en el bloque actual y no esta declarada en otro bloque.

```
#lang stacker/smol/fun
```

```
(deffun (f x y)
```

```
  (defvar n 1)
```

```
  (* n x y))
```

```
(f 5 -9)
```

Como la función  $f\ x\ y$ , son las únicas declaradas con esos nombres en el bloque actual se usarán en la operación de multiplicación, también usando una variable local del bloque llamada  $n$ , que está declarada en el bloque actual, en conclusión, está bien, ya que está declarada en el bloque actual y no está declarada en otro bloque

```
#lang stacker/smol/fun
```

```
(deffun (f y)
```

```
  (defvar n 9)
```

```
  (+ y n))
```

```
(f 7)
```

Como las variables  $n$  e  $y$ , son las únicas declaradas con esos nombres en el bloque actual se usarán en la operación de suma, en conclusión, está bien, ya que está declarada en el bloque actual y no está declarada en otro bloque

### Problema de la página 82 del PDF:

De lo contrario, buscamos en el bloque donde aparece el bloque actual, continuando recursivamente.

```
#lang stacker/smol/fun
```

```
(defun (f x)
```

```
  (defvar y 1)
```

```
  (dex 2)
```

```
  (* y x))
```

```
(f -3)
```

Está bien, ya que busca en el bloque actual, si no busca en otro bloque más grande hasta llegar hasta el superior.

```
#lang stacker/smol/fun
```

```
(defun (f x)
```

```
  (defvar y 3)
```

```
  (defun (g k)
```

```
    (+ y k))
```

```
  (g 0)
```

```
  (- x y))
```

```
(f 13)
```

Está bien, ya que busca en el bloque actual, si no en otros, cuando se halle se hará las acciones del programa, por lo tanto, este programa cumple con el enunciado del problema.

```
#lang stacker/smol/fun
```

```
(defun (f x)
```

```
  (defvar g -3)
```

```
  (* 2 x g))
```

```
(f -4)
```

Está bien, ya que se busca en todos los bloques existentes hasta encontrar la vinculación y hacer las acciones correspondientes del programa, por lo tanto, este programa cumple con el enunciado del problema.

### Problemas de la pagina 83:

Si el bloque actual ya es el bloque primordial y aún no hemos encontrado una declaración correspondiente, la variable no está ligada.

```
#lang stacker/smol/fun
```

```
(defun (f x)
```

```
(defvar y 1)
```

```
(+ y z))
```

```
(x 1)
```

Si la variable no se encuentra hasta llegar hasta el bloque primordial eso quiere decir que la variable no esta declarada, por lo tanto, saldría error, o alguna advertencia que se cambie de/fun a / hof, por lo tanto, este programa es correcto.

```
#lang stacker/smol/fun
```

```
(defun (f x)
```

```
(defun (v w)
```

```
(defvar e 44)
```

```
(- w e))
```

```
(v 5)
```

```
(+ y z))
```

```
(x 1)
```

Ya que la variable o al valor que se intenta ligar no se encuentra entonces, la variable no está declarada en ningún bloque, por lo tanto, este programa es correcto.

```
#lang stacker/smol/fun
```

```
(defvar y 1)
```

```
(+ y z))
```

Ya que la variable a la que se intenta acceder no está ligada a ningún valor, y no esta declarada, entonces cumple con el enunciado del problema.