

Departamento de Ingeniería de Sistemas e Industrial
Universidad Nacional de Colombia
Facultad de Ingeniería
Lenguajes de Programación
Solución al taller de clojure

1. a (220)
2. b (6)
3. a (verdadero)
4. a (verdadero)
5. d (joe)
- 6.

```
1 (defn mayor
2   [x y z]
3     (def mayor x)
4     (if (> y mayor)
5       (def mayor y))
6     (if (> z mayor)
7       (def mayor z))
8     mayor
9   )
10 )
11
12 (println (mayor 1 2 3))
```

```
1 (defn mayor
2   [x y z]
3     (max x y z))
4
5 (println (mayor 4 3 2))
```

- 7.

```
1 (defn perfecto
2   [n]
3     (loop [i 1 divisores[]]
4       (if (= i n)
5         (if (= (reduce + divisores) n)
6             (println "Es perfecto")
7             (println "No es perfecto"))
8         (do
9           (if (= (mod n i) 0)
10              (recur (inc i) (into divisores (set [i]))))
11              (recur (inc i) divisores))))))
12
13 (perfecto 28)
```

8.

```
1 (defn cuadrante
2   [a b]
3   (if (and (> a 0) (> b 0))
4       (println "Pertenece al pimer cuadrante"))
5   (if (and (< a 0) (> b 0))
6       (println "Pertenece al segundo cuadrante"))
7   (if (and (< a 0) (< b 0))
8       (println "Pertenece al tercer cuadrante"))
9   (if (and (> a 0) (< b 0))
10      (println "Pertenece al cuarto cuadrante"))
11  )
12
13
14 (cuadrante 3 -2)
```