

## **Manual de Usuario**



## **Título del Trabajo**

Manual de Usuario

## **Asignatura**

Teoría de Lenguajes de Programación

## **Profesor**

M. en C. Luis R. Basto Diaz

## **Integrantes:**

- Víctor Manuel Lara Lopez (152126359)
- Antonio Alfonso Cetzal Patron (14000516)
- Miguel Angel Quiñones Ramirez (17115665)

**entrega:** 27/11/19

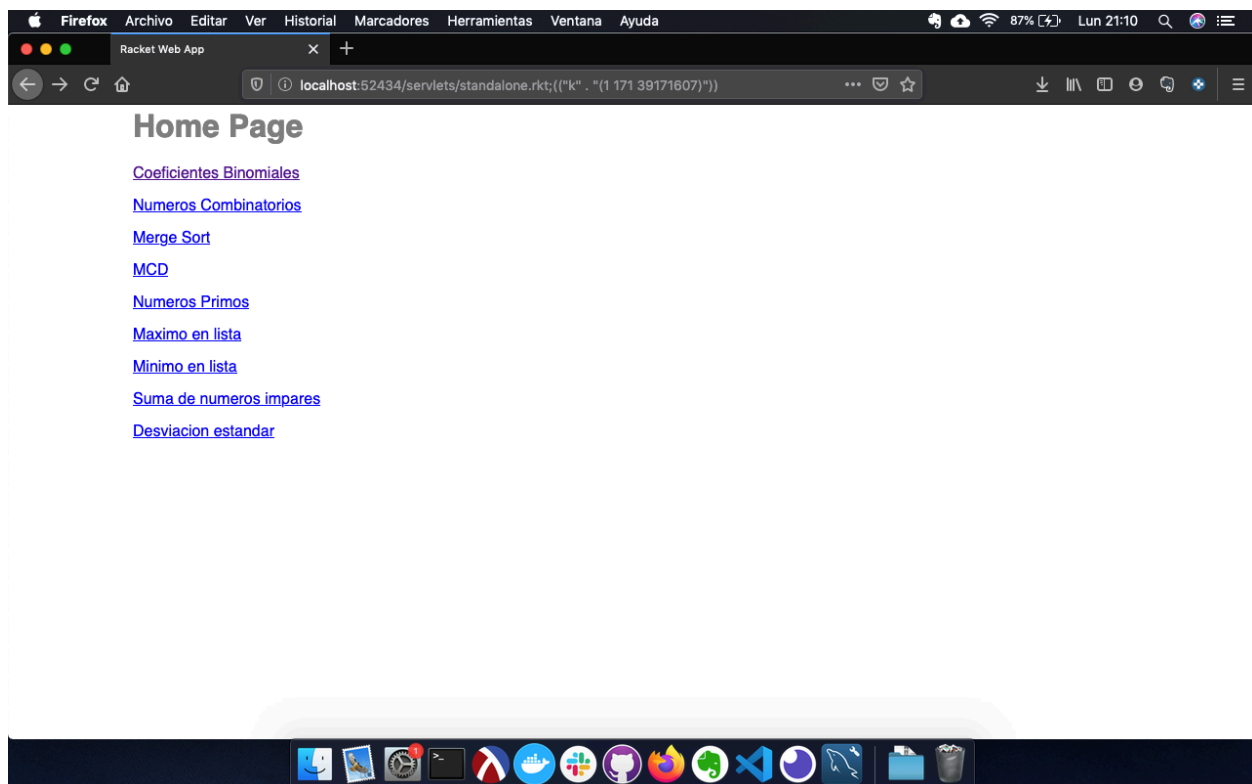
---

# Utilización de la Aplicación

**Nota:** La aplicación esta conformada por los 10 algoritmos y su despliegue vía web, para la utilización de la aplicacion la ejemplificaremos utilizando dos algoritmos, puesto que las acciones de estos dos se repiten en todos los casos dentro de los 10 algoritmos

## Pantalla de inicio

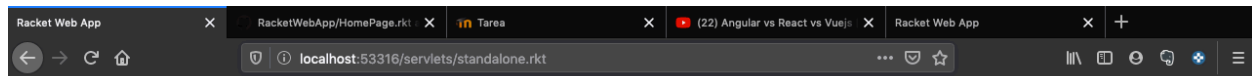
Una vez iniciada la aplicación web, se desplegara la pantalla de inicio que contiene el indice de los algoritmos que podemos utilizar.



Para poder acceder a alguno basta con un click sobre el algoritmo de nuestra preferencia

## Aplicaciones de 1 o 2 datos

Una vez dentro de la aplicación (Ej. "1-cofBin.rkt") la manera de utilizar es ingresando los valores de 'n' en el primer recuadro y 'k' en el segundo, una vez hecho esto procedemos a oprimir el boto "Enviar consulta" para que la aplicación proceda a hacer los cálculos correspondientes:

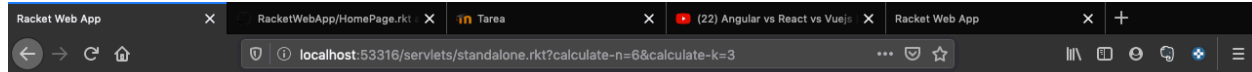


## Coeficientes binomiales (n,k)

6	N	3	K	Enviar consulta
---	---	---	---	-----------------

### Algoritmo

```
(define (fact a)
  (if (= a 0) 1
      (* a (fact (- a 1)))))
(define (coefBin n k)
  (cond
    [(< n 0) 0]
    [(< k 0) 0]
    [(<= n k) 0]
    [else (/ (fact n) (* (fact (- n k)) (fact k)))]))
```



## Coeficientes binomiales (n,k)

20	Resultado	Enviar consulta
----	-----------	-----------------

### Algoritmo

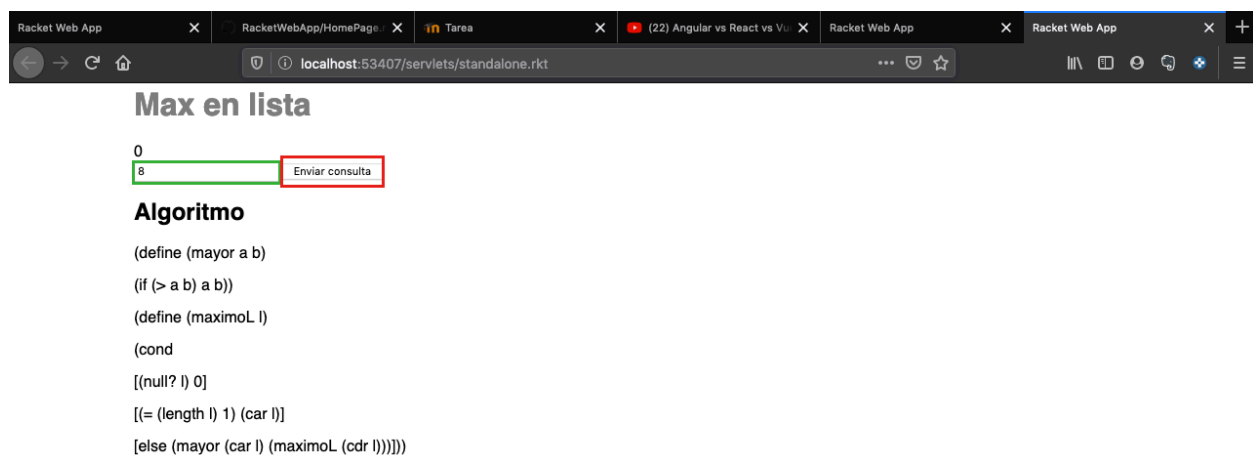
```
(define (fact a)
  (if (= a 0) 1
      (* a (fact (- a 1)))))
(define (coefBin n k)
  (cond
    [(< n 0) 0]
    [(< k 0) 0]
    [(<= n k) 0]
    [else (/ (fact n) (* (fact (- n k)) (fact k)))]))
```

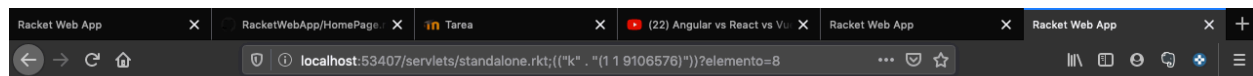


Como podemos observar en la parte superior a los recuadros de información nos aparece el resultado mientras que en la parte inferior tenemos el algoritmo que utilizamos para el cálculo de las dos variables.

## Aplicaciones con listas

Ahora bien para ejemplificar las aplicaciones que utilizan listas, tomaremos el archivo “7-maxList.rkt” y lo ejecutaremos de la misma forma como lo hicimos antes con los otros archivos .rkt. Una vez ejecutado introduciremos los elementos de la lista uno por uno, cada vez que ingresemos un elemento oprimiremos el botón de “Enviar consulta”:





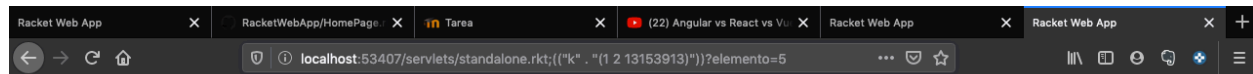
## Max en lista

**8** Resultado  
5 Enviar consulta

**8** lista

### Algoritmo

```
(define (mayor a b)
  (if (> a b) a b))
(define (maximoL l)
  (cond
    [(null? l) 0]
    [(= (length l) 1) (car l)]
    [else (mayor (car l) (maximoL (cdr l)))]))
```



## Max en lista

**8** Resultado  
10 Enviar consulta

**5 8** lista

### Algoritmo

```
(define (mayor a b)
  (if (> a b) a b))
(define (maximoL l)
  (cond
    [(null? l) 0]
    [(= (length l) 1) (car l)]
    [else (mayor (car l) (maximoL (cdr l)))]))
```

**Max en lista**

**10** Resultado

10 5 8 Lista

Enviar consulta

**Algoritmo**

```
(define (mayor a b)
  (if (> a b) a b))
(define (maximoL l)
  (cond
    [(null? l) 0]
    [(= (length l) 1) (car l)]
    [else (mayor (car l) (maximoL (cdr l)))]))
```

Como podemos observar en la parte superior se nos muestra el resultado el cual es el elemento máximo de la lista que vamos ingresando y este se va actualizando así como la lista que se nos presenta en la parte inferior antes del algoritmo.

**Nota:** En cualquiera de los algoritmos no se debe enviar un carácter ni un espacio en blanco!