



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

UNIVERSIDAD DE SONORA

TAREA 3: IMPLEMENTACIÓN DE OBJETOS EN  
RACKET

## Lenguajes de Programación

*Gutierrez Navarro Gustavo*

September 15, 2023

## 1 Razonamientos de las configuraciones

- **Primera configuración:** El punto mas importante durante la elaboración de este ejercicio, ocurrio cuando me percate que al crear vectores cuyos elementos fueran otros vectores, la creación de estos se daban desde dentro hacia afuera, es decir primero los vectores internos y luego los externos. Esta idea fue la idea que me permitio solucionar el problema.
- **Segunda configuración:** La pieza fundamental para elaborar este programa fue el uso de *letrec* y adicionalmente, ajustando la creación de entornos para que se ajustara correctamente a la configuración.
- **Tercera configuración:** Despues de un tiempo intentando recrear la configuración, me percaté que las funciones lambda manejan su propio entorno de variables, esta característica me permitio completar mi programa.
- **Cuarta configuración:** Desde que empeze a resolver este ejercicio, me pude percatar que muy probablemente el programa se trataba de un contador. Esta idea la segui desarrollando, encontrandome con varios problemas los cuales puede resolver investigando en las notas de clase, y en especial con el segundo ejemplo de la implementación de objetos en racket de la propia tarea 3.

## 2 Sección de objetos

- **Objetos básicos:**
  1. Describe qué hace el programa. El programa se encarga de inicializar una variable *init* para posteriormente, tener la posibilidad de incrementar en 1 su valor u obtener el valor actual de la variable. Estas operaciones se dan mediante las palabras claves "inc" y "get" que son pasadas como parametros.
  2. ¿Qué *clases* hemos codificado? La clase se llama *o - state - 2* y podria ser clasificado como un contador.
  3. ¿Qué *objetos* hemos codificado? El único objeto que hemos codificado es *o* una instancia de la clase anteriormente nombrada.
  4. ¿Qué *atributos* hemos codificado? Nuestra clase tiene un solo atributo llamado *init* el cual podriamos decir que es de tipo entero y se inicializa en el constructor.
  5. ¿Qué *metodos* hemos codificado? Tenemos en si un solo método el cual va actuar diferente dependiendo del parametro que le pasemos, sin embargo si lo interpretamos como un objeto, podemos ver que tenemos dos métodos, *inc* y *get*, el primero se encarga de incrementar el atributo *init* en 1 y el segundo nos devuelve su valor actual.

6. Describe cómo se implementaron las *clases*. La clase se implemento mediante el uso de una función que devuelve una función lambda la cual hace uso de la variable *init* para asemejar a cuando una clase crea una instancia con varios parametros.
7. Describe cómo se implementaron los *objetos*. Los objetos estan presentes cuando definimos una variable y le asignamos de valor la llamada a la función (la clase), esto permite que cada una de las variables que creemos tenga su propio entorno distinto al resto, permitiendo que exista el encapsulamiento.
8. Describe cómo se implementaron los *atributos*. El único atributo que se implemento *init*, dicho valor se puede inicializar con la primera llamada a la función ya que es la encargada de iniciar el entorno donde va a trabajar el objeto creado.
9. Describe cómo se implementaron los *metodos*. Los metodos se implementaron mediante la función lambda, que utiliza su único parametro *m* para pasarle el nombre del metodo que se quiera usar, de esta manera podemos escribir cuantos metodos queramos a manera de un switch case dentro de la función lambda.

• **Atributos estáticos:**

1. Describe qué hace el programa. Este programa funciona un poco distinto al anterior. Este programa maneja dos variables importantes, un contador que maneja el número de veces que la clase ha sido creada (en otras palabras cuantas instancias del objeto han sido creadas) y otra que se inicializa en un número y que tiene un método para incrementar su valor en 1. La primera variable, el contador funciona de manera global y compartida entre todas las estancias de la clase mientras que el acumulador es individual para cada objeto.
2. ¿Cuántos objetos son creados? 2 objetos son creados  $o - 1$  y  $o - 2$ .
3. ¿Cuál es el valor del amount de  $o-1$  al final de la ejecución del programa? 5.
4. ¿Cuál es el valor del amount de  $o-2$  al final de la ejecución del programa? 8.
5. ¿Cuál es el valor del counter de  $o-1$  al final de la ejecución del programa? 2.
6. ¿Cuál es el valor del counter de  $o-2$  al final de la ejecución del programa? 2.