**TECNOLÓGICO DE COSTA RICA**

**Escuela de Ingeniería en Computación**

**Proyecto de Ingeniería de Software**

**Profesora:**

María Estrada

**Entrega 1:**

Resumen técnico

de OCAML

**Estudiantes:**

Christian Leon Guevara - 2013371982

Gabriel Ramírez Ramírez - 201020244

**Fecha de entrega:**

22-12-2018

**Periodo Verano**

[**Introducción**](#_jed3dfm0gf1m)3

[**Comandos**](#_kxr9onqobcdf)3

[**Expresiones**](#_bwje23oe1w3x)4

[**Nombre de variable**](#_veue6h8x5npm)4

[**Declaraciones**](#_4qu9ol16i6o6)5

[**Parámetros**](#_ktb6agwoi55z)5

[**Tipos**](#_m17st8tzngw4)6

Lenguaje Triángulo

# Introducción

El lenguaje OCAML una abreviación de *Objetive CAML* y representa un lenguaje de programación avanzado de la familia ML desarrollado y distribuido en Francia.

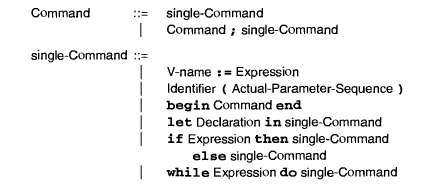
El lenguaje triángulo tiene el siguiente tipo de Declaraciones:

* Valor: es un valor real, por ejemplo, un entero numérico, un valor de verdad, un carácter, un registro o un arreglo.
* Variable: Entidad que contiene un valor.
* Procedimiento: Entidad que tiene un bloque de código que ejecuta una serie de instrucciones.
* Función: Entidad que tiene un bloque de código que será evaluado para retornar un valor
* Tipo: Determina un conjunto de características que el valor va a tener. Cada variable, valor y función lo tienen.

El lenguaje se puede dividir en múltiples secciones, a continuación, se explicará el comportamiento de algunos segmentos de la gramática.

# Comandos

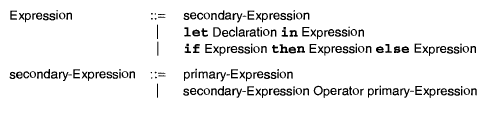
Un comando es una línea de código valida, que se puede concatenar una después de otra.

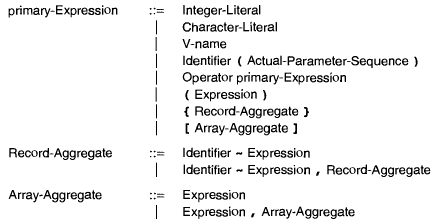


* La expresión “V-name := Expression” funciona como una asignación en donde “V-name” es la variable y “Expression” es la expresión.
* La expresión Identifier(Actual-Parameter-Sequence), funciona si Identifier es un procedimiento y Actual-Parameter-Sequence es una secuencia de parámetros válida.
* Se ejecuta un comando tras otro.
* En caso de que exista “begin C end”, se ejecutará solamente C.
* El if se ejecuta de tal manera que si la primera expresión es verdadera se ejecutará la segunda expresión, de lo contrario será la tercera expresión. El lenguaje debe verificar que la primera expresión tenga valor booleano para este efecto.
* El while funciona de la siguiente manera, se evalúa la expresión y si es true entonces se sigue ejecutando el siguiente single-Command. También hay que verificar que la expresión tenga valor booleano

# Expresiones

Una expresión es un bloque de código que puede devolver un valor.





* La expresión “Integer\_Literal” se refiere a un valor entero numérico.
* La expresión “Character\_Literal” se refiere a un valor de tipo carácter.
* La expresión “V-name” se refiere a un nombre de una variable.
* El bloque let permite declarar una variable A y asignarle un valor B para ese segmento de código en específico. Estas variables no tienen efecto fuera del segmento de código en el que se declararon.

# Nombre de variable

Un nombre de variable es un identificador al cual se le tiene asignado un tipo y se le puede asignar un valor.



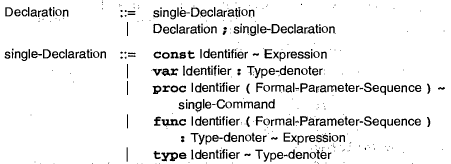
Un nombre de variable se puede expresar de 3 diferentes maneras:

* Simplemente escribiendo un identificador
* Haciendo la llamada a un registro (V-name.Identifier)
* Haciendo la llamada a un arreglo(V-name[Expression])

Las 2 últimas se pueden concatenar siempre y cuando siempre empiece con un identificador.

# Declaraciones

Sirve para enlazar 2 cosas, usualmente a alguna expresión.

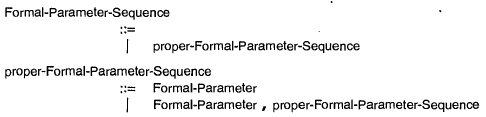


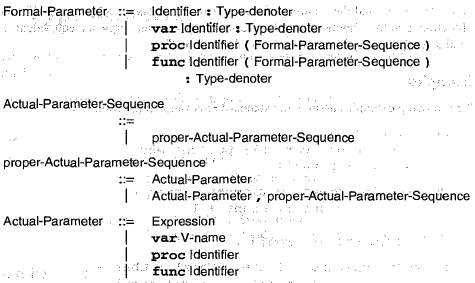
Las declaraciones funcionan de la siguiente manera:

* La declaración tipo const se enlaza un valor a un identificador, este valor no se puede alterar.
* La de tipo var, declara una variable y el tipo correspondiente; si se le puede asignar un valor después.

# Parámetros

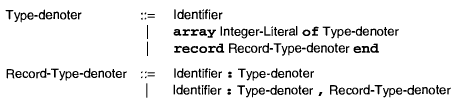
Es la especificación de identificadores en el cuerpo de una función o procedimiento. Existe la especificación de los parámetros y los parámetros reales. La especificación determina ciertas características de los parámetros a usar (como declarando una variable) y los parámetros reales son cualquier expresión o llamada a función que devuelva un valor.





# Tipos

Un tipo es una clase de etiqueta que se le da a algo para otorgarle ciertos atributos a ese algo. Se le puede enlazar un tipo a valores, constantes, variables y funciones.



* Los registros son un encapsulamiento de variables, en donde son accedidos por medio de un identificador.
* El Identifier se refiere a un Tipo ya definido por el sistema como Integer, Boolean, etc.

Bibliografía

* Watt, David A., and Deryck F. Brown. *Programming Language Processors in Java: Compilers and Interpreters*. Pearson Education (Singapore) Pty. Ltd., 2003.