

# Chemist v0.9

## Trabajo Práctico Final - ALP

Camilo Garcia Gonzalez

24 de Febrero del 2021

### 1 Chemist.

Se describe a continuación un lenguaje simple para la realización de cálculos y simulación de experimentos químicos. Se inicia dando su sintaxis abstracta completa, la cual se basa en un pequeño lenguaje imperativo al estilo visto en clase, y lo extiende introduciendo conceptos como elemento, molecula, muestra, etc. Estos conceptos serán explicados posteriormente mediante un programa de ejemplo.

### 2 Sintaxis abstracta.

```
symbol    ::= string

atom       ::= symbol double nat

molecule ::= nat          * atom
           | molecule + molecule

sample     ::= molecule double double double

numexp     ::= nat
           | double
           | var
           | numexp + numexp
           | numexp * numexp
           | numexp - numexp
           | numexp / numexp
           | var      ! string
```

```

boolexp ::= True
          | False
          | var
          | ! boolexp
          | boolexp & boolexp
          | boolexp | boolexp
          | numexp == numexp
          | numexp > numexp
          | numexp < numexp
          | numexp >= numexp
          | numexp <= numexp
          | molecule == molecule

exp ::= numexp
      | boolexp

comm ::= skip
        | var = exp
        | var <= sample
        | symbol = molecule
        | comm . comm
        | if boolexp then comm
        | if boolexp then comm else comm
        | while boolexp do comm
        | return exp

```

### 3 Un programa de ejemplo.

Por ejemplo, con este lenguaje podría escribirse un simple programa que calcula el peso de 15 litros de dióxido de carbono a una temperatura de 100°C y una presión de 756mmHg, de la siguiente manera:

```

1 C + 2*0 = C02.
2
3 x <= C02 {v=15; t=100; p=756}.
4
5 res = (C02_w * x_p * x_v) / (62,36 * (x_t + 273)).
6
7 return res.

```

## 4 Explicación programa anterior.

### 4.1 El operador =.

El operador = estaría sobrecargado para asignación de variables y para definición de nuevos elementos químicos en base a otros ya definidos, como se muestra en la línea 1). La idea es que en el evaluador estén cargados todos los elementos de la tabla periódica y el programador pueda usarlos como datos básicos para la definición de moléculas compuestas.

### 4.2 El operador <=.

Se introduce el concepto de "muestra" que consiste en simular la manipulación de una determinada cantidad de un elemento con la intención de realizar cálculos y experimentos. Para esto, debe especificarse el elemento en cuestión y deben darse las condiciones en que se encuentra dicho elemento (encerradas entre {}), estas condiciones son: presión, volumen y temperatura.

En la línea 3) del programa, se establece que x representa una muestra de 15 litros de dióxido de carbono, a una temperatura de 100 grados centígrados y una presión de 756 mmHg.

**Nota:** resta establecer el correcto manejo de las unidades dentro del lenguaje.

### 4.3 Los operadores \_.

Dado un elemento cualquiera, supongamos por ejemplo Hidrógeno, podemos acceder a ciertas propiedades del elemento con los operadores H.w y H.n (peso atómico y número atómico respectivamente). A su vez, dada una muestra cualquiera m, podemos acceder a sus propiedades antes definidas mediante m.v, m.p y m.t.

**Nota:** Una posible extensión sería agregar algunas propiedades interesantes más a cada elemento, como por ejemplo, período (niveles de energía, esto es, las diferentes distancias entre órbitas de electrones), grupo (número de valencia). Aunque esto sería válido sólo para los elementos que consisten de un átomo (C, H, O) y no para compuestos (CO<sub>2</sub>, H<sub>2</sub>O).

### 4.4 Evaluación de un programa.

La línea 3) realiza el cálculo deseado en base a la ley de los gases ideales y lo asigna a la variable res, la cual mediante la línea 4) resulta ser el valor final del programa.

**Nota:** En principio no se considera la introducción de funciones, dado el gran salto en complejidad que esto representa, por lo que cada programa vendría a ser como una única función cuyo valor de retorno es el valor del programa y está dado por la palabra reservada **return**.