**Universidad Autónoma de Ciudad Juárez**

**Ciudad Universitaria**

**Instituto de ingeniería y Tecnología**

# Propuesta de lenguaje de programación

George Iván Rodríguez Gómez 160000

Ricardo Mar Cupido 158883

Jonathan Noe Viramontes Ramírez 159951


## Teoría de lenguaje de programación

BATLAN

*Agosto del 2017*

**Nombre:** BATLAN

El lenguaje es llamado BATLAN en honor al superhéroe Batman debido a que sus características están inspiradas en las funcionalidades de su traje y el nombre podría ser traducido como Batman lenguage, inicialmente seria llamado batilenguaje pero era demasiado largo y se decido cambiar por algo más corto y quedo como BATLAN.

**Paradigma:** Imperativa, orientada a objetos, guiada por eventos y concurrente.

- **Imperativa:** La programación será flexible frente a los diferentes tipos de datos, pero en caso de ambigüedad se le informará al usuario la situación, el LP se encargará de definir los tipos manteniéndolo optimizado y compacto debido a que un traje inteligente cuenta con recursos disponibles limitados además requiere cálculos muy precisos; la asignación de variables, cambios en variables son indispensables en muchas de las operaciones que pueden ejecutarse.
- **Orientada a objetos:** La posibilidad de abstraer secuencias de movimientos básicos y complejos, con la posibilidad de acceder a características de las componentes del traje, así como de sus accesorios sin necesidad de recurrir a alguna biblioteca.
- **Guiada por eventos:** Debe de tener la capacidad de detectar los movimientos del usuario y activar las secuencias de código correctas de acuerdo al tipo de movimiento que esté realizando, por ejemplo, si el usuario decide mover el antebrazo los sensores del traje activaran las secuencias del código para mover el antebrazo.
- **Concurrente:** Debe de ser concurrente debido a que el exoesqueleto tendrá que desarrollar en algunas ocasiones dos o más funciones a la vez, por ejemplo, la posibilidad de entrar en un estado de defensa autónomo, es decir permitir al usuario ejecutar código referente al sistema defensivo u ofensivo del exoesqueleto mientras de manera independiente realice otras acciones.

**Aplicación:** Sera un lenguaje de programación destinado a los exoesqueletos y sus accesorios. Podrá ser utilizado para el ámbito laboral, militar o de la medicina

**Criterios de diseño**:

- **Legibilidad**: Este Lenguaje de programación está pensado para que los desarrolladores se adapten de forma muy rápida, el L.P. usara palabras reservadas en Idioma ingles debido a que este es un idioma estándar en muchos países y es usado actualmente por una cantidad muy grande de lenguajes de programación, creemos que si algo funciona bien para que cambiarlo
- **Ortogonalidad**: Se tendrá como propósito general que los diferentes procesos del lenguaje sean independientes unos de otros y evitar funciones similares, para que de esta forma no se cometan errores en la ejecución y sea más fácil de comprender para los desarrolladores. Será flexible en cuanto la posibilidad de

definir variables en cualquier punto del programa, sin embargo, aún se tendrán que definir a que tipo pertenecen

- **Capacidad de escritura:** Se implementarán funciones en concordancia al idioma manteniéndola clara y concisa para el sistema. Por ejemplo, walk from place1 to place2.

**Tokens:**

Todos los tokens podrán ser escritos en mayúsculas o minúsculas, el lenguaje de cualquier manera detectara que se trata de un token.

- **Palabras reservadas:** Estas palabras indican al computador la acción que va realizar y tienen una función muy específica dentro del compilador

main, private, public, import, if, then, else, try, except, while, for, while do, case, Switch, asm, arm, left, right, up, down, alfred (Alfed seria palabra con la que se comunicaría el asistente de voz del traje), mrj (De Mr. J, es el comando que activaría la secuencia de búsqueda de enemigos o amenazas), jump, walk, lights(), true, false, vehicle, climb, weapon, carry, bint, bdouble, bchar, bstring, bfloat, bshort, bbyte, blong, bint[], here,MsgWarning.

- **Constantes:** Son valores que siempre son lo mismo, no importa lo que pase no se pueden modificar.

PI, e, Fi, StepByStep

- **Literales:** Un valor literal es una constante formada por una secuencia de caracteres que no puede ser modificado en la ejecución del programa.

bstring user="batman123", bstring password="Ironmanapesta", bfloat gravity=9.81f,

- **Símbolos especiales:** Indican operaciones especiales que se van a realizar con las constantes o variables.

{,}, (,), @, +, -, *, /, \, >, <, %, ", ´, punto, coma, &, !, +=, -=, *=, /=, ==, =, !=, &&, ^, ||.

- **Nombre de variable:** Los identificadores o nombre de variable no pueden comenzar con un número, puede comenzar con un (_) o utilizar cualquier letra de la "a" a la "z" ya sea una minúscula o mayúscula, pero como en la mayoría de los L.P. esto sí importa. Tampoco se podrán usar palabras reservadas como variables ni símbolos especiales, ni dos variables o identificadores con el mismo nombre.

Por ejemplo:

bint radio, bfloat total, bstring _nombre, bchar _alt164, etc.

**Ejemplo de código en Batlan para responder una emergencia:**

```
Import situations
Import threading
Import defense
main emergencies (){
        under_attack: situation
        threads=list()
        defensive=threading.Thread(Under_attack)
        threads.append(defensive)
        defensive.start
        emergency: situation
        place1, place2:  location
        place1=emergency.place
        place2=here
        try{
            If (place1 =! place2)
                    {
                    go_walk:action
                    for (place1, place2, StepByStep)   go_walk.walk
                    }
            else
                    {
                    instruction:action
                    if (instruction.event()) { instruction.do}
                    }
            } except Error:SystemError
            {
            MsgWarning('El sistema no puede operar porque:',Error)
```

```
        }
}
```

**Categorías Léxicas**

**Tokens y ejemplos**

- **Palabras reservadas:**
  - **Numero entero:** bint
    - Positivo
    - Negativo
  - **Numero reales:** bfloat
    - Positivo
    - negativo
  - **Numero doble:** bdouble
  - **Variable texto:** bstring
  - **Variable carácter:** bchar
  - **Importar librería:** import
  - **Decision:** if, else
  - **iterativos**: for, while, do
  - **Partes del traje:** arm, leg, shoulder, etc.
  - **Eventos:** event, listener.
- **Funciones**
  - **Raiz cuadrada:** sqrt()
  - **Elevación a una potencia** pow(x,y)
  - **Localización actual:** here, location.
  - **Captura de errores:** try, except, Error, SystemError, Error
  - **Levantar**: up(5)
  - **Descender**: down(5)
- **Identificadores:**
  - bint numerotraje
  - bfloat localizacion
  - bstrIng usuario
- **Operadores:**
  - **+**
  - **-**
  - **/**
  - *****
  - **%** (MOD)
  - **=**
  - **^**
  - log2 ()
  - log10 ()
  - ln2 ()

    - ln10 ()
- **Constantes:**
    - PI
    - e
    - Fi
    - StepByStep

- **Simbolos especiales:**
    - {
    - }
    - [
    - ]
    - (
    - )
    - @
    - +
    - -
    - *
    - /
    - \
    - >
    - <
    - ^
    - %
    - "
    - ´
    - Punto
    - Coma
    - &
    - !
    - +=
    - -=
    - *=
    - /=
    - ==
    - =
    - !=:
    - &&
    - ||
- **Texto:** "Hola Mundo"

**"Tokenizar"**

| **Sentencia** | **Token** |
|---|---|

| | |
|---|---|
| import | **Palabra reservada** |
| situations | **Nombre de la biblioteca** |
| | **Espacio en blanco** |
| import | **Palabra reservada** |
| threading | **Nombre de la biblioteca** |
| | **Espacio en Blanco** |
| import | **Palabra reservada** |
| defense | **Nombre de la biblioteca** |
| | **Espacio en Blanco** |
| main | **Palabra reservada** |
| emergencies | **Nombre de la clase** |
| ( | **Simbolo especial: Paréntesis izquierdo** |
| ) | **Simbolo especial: Paréntesis derecho** |
| { | **Simbolo especial** |
| Under_attack | **Variable: Objeto** |
| : | **Simbolo especial** |
| situation | **Variable: Objeto** |
| Threads | **Variable: Objeto** |
| = | **Operador: Asignacion** |
| list | **Función** |
| ( | **Símbolo especial** |
| ) | **Símbolo especial** |
| defensive | **Variable: Objeto** |
| = | **Operador** |
| Threading | **Variable: Objeto** |
| . | **Símbolo especial** |
| Threads | **Función** |
| ( | **Símbolo especial** |
| Under_attack | **Constante: objeto** |
| ) | **Símbolo especial** |
| defensive | **funcion** |
| . | **Símbolo especial** |
| start | **Funcion** |
| emergency | **Objeto asignado** |
| : | **Símbolo especial: Carácter de asignación** |
| situation | **objeto** |
| place1 | **Variable float** |
| , | **Símbolo especial** |
| place2 | **Variable float** |
| : | **Símbolo especial** |
| location | **objeto** |
| = | **Operador:Asignación** |

| | |
|---|---|
| Emergency | **Variable: Objeto** |
| . | **Símbolo especial** |
| place1 | **Variable float** |
| here | **funcion** |
| try | **Palabra reservada** |
| { | **Símbolo especial** |
| If | **Palabra reservada** |
| ( | **Símbolo especial** |
| place1 | **Variable float** |
| = | **Símbolo especial** |
| ! | **Símbolo especial** |
| place2 | **Variable float** |
| ) | **Símbolo especial** |
| { | **Símbolo especial** |
| go_walk | **Variable: evento** |
| : | **Símbolo especial** |
| action | **Evento** |
| for | **Palabra reservada** |
| ( | **Símbolo especial** |
| place1 | **Variable float** |
| , | **Símbolo especial** |
| place2 | **Variable float** |
| , | **Símbolo especial** |
| StepByStep | **Constante** |
| ) | **Símbolo especial** |
| go_walk | **Variable: evento** |
| . | **Símbolo especial** |
| walk | **Objeto** |
| } | **Símbolo especial** |
| else | **Palabra reservada** |
| { | **Símbolo especial** |
| Instruction | **Variable: evento** |
| : | **Símbolo especial** |
| action | **Evento** |
| If | **Palabra reservada** |
| ( | **Símbolo especial** |
| instruction | **Variable: evento** |
| . | **Símbolo especial** |
| event | **evento** |
| ( | **Símbolo especial** |
| ) | **Símbolo especial** |
| ) | **Símbolo especial** |
| { | **Símbolo especial** |
| instruction | **Variable: evento** |
| . | **Símbolo especial** |

| | |
|---|---|
| do | **Palabra reservada** |
| } | **Símbolo especial** |
| } | **Símbolo especial** |
| } | **Símbolo especial** |
| except | **Palabra reservada** |
| Error | **Variable de error** |
| : | **Símbolo especial** |
| SystemError | **Palabra reservada** |
| { | **Símbolo especial** |
| MsgWarning | **Palabra reservada** |
| ( | **Símbolo especial** |
| ' | **Símbolo especial** |
| El sistema no puede operar porque: | **Texto** |
| ' | **Símbolo especial** |
| , | **Símbolo especial** |
| Error | **Palabra reservada** |
| ) | **Símbolo especial** |
| } | **Símbolo especial** |
| } | **Símbolo especial** |
| | |

**Errores léxicos**

- main emergencies{
  emergency: situation
  place1, place2: ubication
  place1= emergency.place
  place2= here
  go_walk:action

  }

- main medical attention (){
  under_attack:situation
  if (vital_signs=critical){
  administer_morphine:action
  }

- main defense(){
  under_attack=situation
  if(missile_detector= alert_actived){
  deploy_shafts:action
  }
  }

**Errores sintácticos**

- Emergencies main{
  emergency: situation
  place1, place2: ubication
  emergency.place= place1
  place2= here
  go_walk:action
  }

- main medical() attention{
  under_attack:situation
  vital_signs= (if = critical){
  administer_action:morphine
  }
- main under:attack() attention{
  under_medical:situation
  View.emergency_(for = critical){
  administer_action:

  }

  }