

C++

Juan Diego Barrado Daganzo, Javier Saras González y Daniel González Arbelo
4º de Carrera

26 de febrero de 2024*

1. Especificaciones técnicas del lenguaje

1.1. Identificadores y ámbitos de definición

1.2. Tipos

Los tipos básicos del lenguaje son los enteros y los booleanos (caracteres?), pudiendo formar arrays de estos tipos. Cada uno se declarará usando estas palabras reservadas:

- **Enteros:** `int`
- **Booleanos:** `bool`
- **Array:** `Tipo[DIMENSION] var`

El lenguaje también posee otros tipos compuestos como clases y registros. Adicionalmente, se ha considerado un tipo para las funciones y para el vacío¹. Las palabras reservadas para ello son las siguientes:

- **Clases:** `clas`
- **Registros:** `estruct`
- **Funciones:** `func`
- **Tipo vacío:** `void`

Por último, los punteros a cualquiera de estos tipos se declararán poniendo el caracter '~' al final de la palabra reservada para el tipo.

- **Puntero:** `Tipo~ var`

Como nota especial, la declaración de un puntero a estructuras de tipo array, se haría como `Tipo[DIMENSION] var`.

*Este documento se actualiza, para consultar las últimas versiones entrar en el enlace <https://github.com/JuanDiegoBarrado/PracticaPL>

¹El tipo vacío es necesario para poder

1.3. Instrucciones del lenguaje

El lenguaje tiene el siguiente repertorio de instrucciones:

- Instrucción de asignación: `:=`

```
1      int var := 3
```

- Instrucciones condicionales: `if-else`, `switch`

```
1      if (var > 3) {
2          ...
3      }
4      else {
5          ...
6      }
```

- Instrucción de bucle: `while`

```
1      while (var > 0) {
2          ...
3      }
```

Se incluyen además las instrucciones `breic` y `continui`.

1.4. Sucio

- Las constantes se declaran como los `# define` de C++, pero con `# difain`.
- Los nombres de variable siguen los mismos patrones que en C++.
- ~~Los arrays como las variables normales, pero con corchetes indicando la dimensión (como en C++).~~
- Indicamos que una variable es puntero con `int~ var`.
- Paso por valor y por referencia igual que en C++.
- ~~Los bloques tienen como separadores las llaves.~~
- ~~Los struct se escriben `estruct`.~~
- ~~El identificador de tipo es `clas`.~~
- ~~El identificador de los tipo entero es `int` y el de los booleanos es `bul`.~~
- Operadores infijos:
 - Aritméticas como C++, añadimos el operador exponencial.
 - Asignación es `:=`.
 - Igualdad es `=` y la desigualdad como en C++.
 - Los operadores booleanos son: `an`, `or`, `sor`, `not` y menor mayor es como en C++.
 - Operador parentesis y corchetes también.
- Para definir los tipos `taipdef`.
- Para definir las funciones `func nombre(args): tipoRetorno {...}`.
- Las funciones de entrada salida son `cein` y `ceaut` con las funciones de modificación de entrada salida que vayamos necesitando. La intrucción de retorno para las funciones es `return valor`
- Acceso a punteros `~puntero`

■ Instrucciones del lenguaje:

- Asignación: `:=`
- Condicionales `if () { ... } else { ... }` y `switch () { caseis () ... }`
- Bucle indefinido `guail`
- Bucle definido `for`.
- Sentencias de control de bucles: `break` y `continuu`.