Proyecto parcial de Lenguajes de Programación

Josue Cobos Salvador, Andrés Vargas, Enmanuel Magallanes

Escuela Superior Politécnica del Litoral

Resumen

En este documento se van a revisar las principales características del interpretador a implementar, incluirá detalles tanto léxico, sintáctico y semántico. Dart será el lenguaje para tomar en cuenta, por ende, en el documento se detallan la mayoría de los componentes de sintaxis y semántica del lenguaje.

Palabras clave: Lenguajes de programación, Léxico, Sintáctico, Semántico, Dart

Proyecto parcial de Lenguajes de Programación

Gracias a la tendencia actual por el crecimiento de la comunidad de código abierto, surgen soluciones específicas para problemas específicos, esto no es una excepción para los lenguajes de programación, Dart es un lenguaje de programación de código abierto con el que se desarrollan aplicaciones multiplataforma, permitiendo compilar aplicaciones web como móviles.

Dart nace como una iniciativa de Google buscando una alternativa de los problemas que presentaba javascript cerca del año 2011, este lenguaje tendría una máquina virtual nativa en google chrome para poder ser interpretado. La primera versión oficial se lanzó a finales de 2013. Este lenguaje puede ser usado en web, siempre y cuando sea transpilado a Javascript, o usando el framework AngularDart, también está presente en aplicaciones de escritorio y móviles usando su framework Flutter.

En cuanto al uso de Dart en la industria, para el 2020 según la encuesta anual de stackoverflow Dart guarda un respetable 6to puesto dentro de los lenguajes más queridos por la comunidad desarrolladora (cerca de 65000 encuestados).

**Modificaciones del Lenguaje**

Debido a la gran influencia de un gran deportista que práctica fútbol de manera profesional llamado Cristiano Ronaldo, mejor conocido como “El Bicho”, “El Comandante” o “CR7”, se ha optado por modificar la sentencia de impresión por pantalla:

print("Goaaal!");

por la expresión declarada abajo. Debido a que “SIUU” es el grito que este jugador realiza cada vez que anota un gol. Y se quiere que esta versión de dart haga referencia a él.

siu("Goaaal!");

# Componentes Léxicos

## Formas correctas para definir variables y su inicialización:

No es fuertemente tipado.

var name = "Cristiano Ronaldo";

Se usa la palabra clave ‘dynamic’ cuando el tipo de dato cambiará en el futuro.

*dynamic* name = "Cristiano Ronaldo";

Se puede inicializar la variable directamente con el tipo de dato

*String* name = "Cristiano Ronaldo";

## Tipos de datos (primitivos y estructurados)

|  |  |  |
| --- | --- | --- |
| Tipo | Descripción | Ejemplos |
| Números Enteros | Tipo de dato primitivo entero | *int* *goals*;  *int* *goals* = 5; |
| Double | Tipo de dato numérico con decimales | *double* distance = 5.78; |
| Booleanos | Tipo de dato primitivo asignado a valores lógicos primarios/valores de verdad | *bool* is\_mvp = true; |
| Strings | Tipo de dato estructurado definido para cadenas de caracteres. Cuya codificación corresponde a unidades de UTF-16. Podemos usar tanto  comillas dobles como simples. | *String* = "doble quote" |
| Listas | Tipo de dato estructurado que representa a una colección de valores agrupados en la estructura de datos lista. En Dart, estas listas son análogas a los arreglos en otros lenguajes de programación, son una serie de elementos indexados. | var best\_players = ['CR7',  'Messi', 'Mbappe'];  *List* best\_players2 = ['CR7',  'Kanté', 'Mbappe'];  *List*<*int*> numbers = [7, 10,  7]; |
| Mapas | Tipo de dato estructurado definido que difiere de las listas debido a que en lugar de usar índices per ser, utiliza pares clave valor para acceder a los datos almacenados en la  estructura. | var position = {      'MCO': 'Noboa'      'MD': 'Plata'      'MI': 'Ibarra'  } |

## Palabras reservadas importantes

|  |  |
| --- | --- |
| Palabra | Uso |
| break | Control condicional |
| dynamic | Tipo de variable dinámica |
| else | Control condicional |
| catch | Manejo de excepciones |
| const | Variable constante |
| continue | Control condicional |
| do | Control iterativo |
| for | Control iterativo |
| function | Define una función |
| if | Control condicional |
| return | Retorna valor |
| var | Tipo de variable |
| void | No retorna valores |
| while | Control iterativo |
| false | Booleano |
| finally | Manejo de excepciones |
| true | Booleano |
| siu | Sentencia de impresión |

## Operaciones

Teniendo en cuenta que:

var muster = 35;

| Tipo de operador | Denominación | Símbolo | Ejemplo | Resultado |
| --- | --- | --- | --- | --- |
| Aritmético | Suma | + | muster + 2; | 37 |
|  | Resta | - | muster - 2; | 33 |
|  | Multiplicación | \* | muster \* 3; | 105 |
|  | División | / | muster / 7; | 5 |
|  | Divide y retorna un resultado entero | ~/ | muster ~/ 7; | 11 |
|  | Asigna suma | += | muster += 6; | 41 |
|  | Asigna resta | -= | muster -= 7; | 28 |
|  | Asigna producto | \*= | muster \*= 2; | 70 |
|  | Asigna división | /= | muster /= 7; | 5 |
|  | Modulo | % | muster % 2 | 1 |
|  |  |  |  |  |
| De igualdad y relacionales | Es igual | == | muster == 35; | true |
|  | No es igual | != | muster != 44; | true |
|  | Menor que | < | muster < 44; | true |
|  | Menor o igual que | <= | muster <= 33; | false |
|  | Mayor que | > | 44 > muster; | true |
|  | Mayor o igual que | >= | muster>=23; | true |
|  |  |  |  |  |
| Incremento y decremento | Incrementa | ++ | ++muster; | 36 |
|  | Incrementa | ++ | muster++; | 36 |
|  | Decrementa | -- | --muster; | 34 |
|  | Decrementa | -- | muster--; | 34 |
|  |
|  |  |  |  |  |
| Lógicos | y | && | true && true | true |
|  | o | || | true || false | true |
|  | Negación | ! | !true | false |
|  |  |  |  |  |

## Estructuras de control

### If-else

void main() {

    var *num* = 12;

    if (*num* % 2==0) {

        siu("Even");

    } else {

        siu("Odd");

    }

}

### For

for( var i = 0 ; i <= 10; i++ ) {

    siu(i);

}

### While

var i = 0;

while(i <= 10) {

    siu(i);

    i++;

}

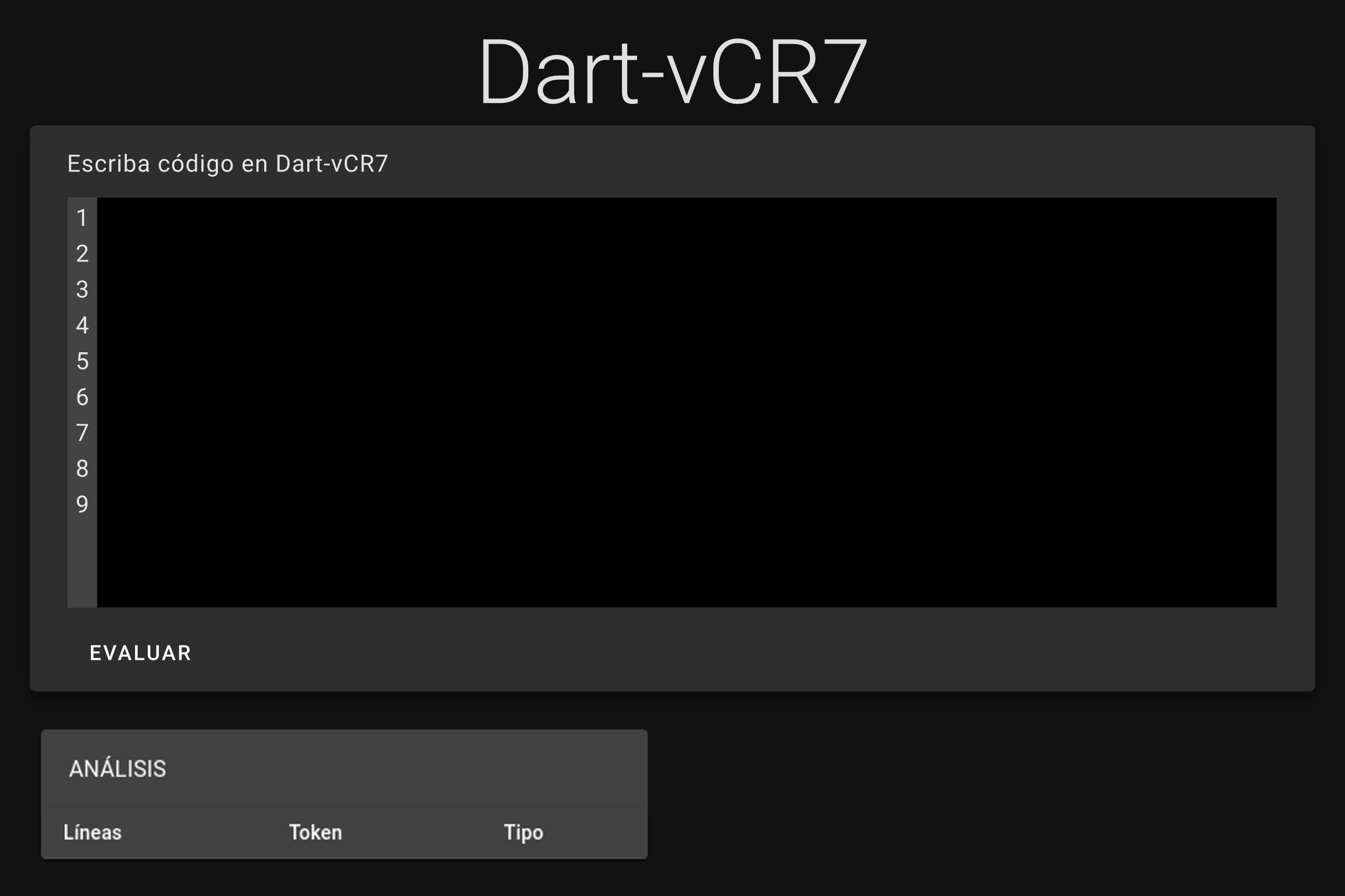
**Prototipo**

Figura 1: Prototipo - 1

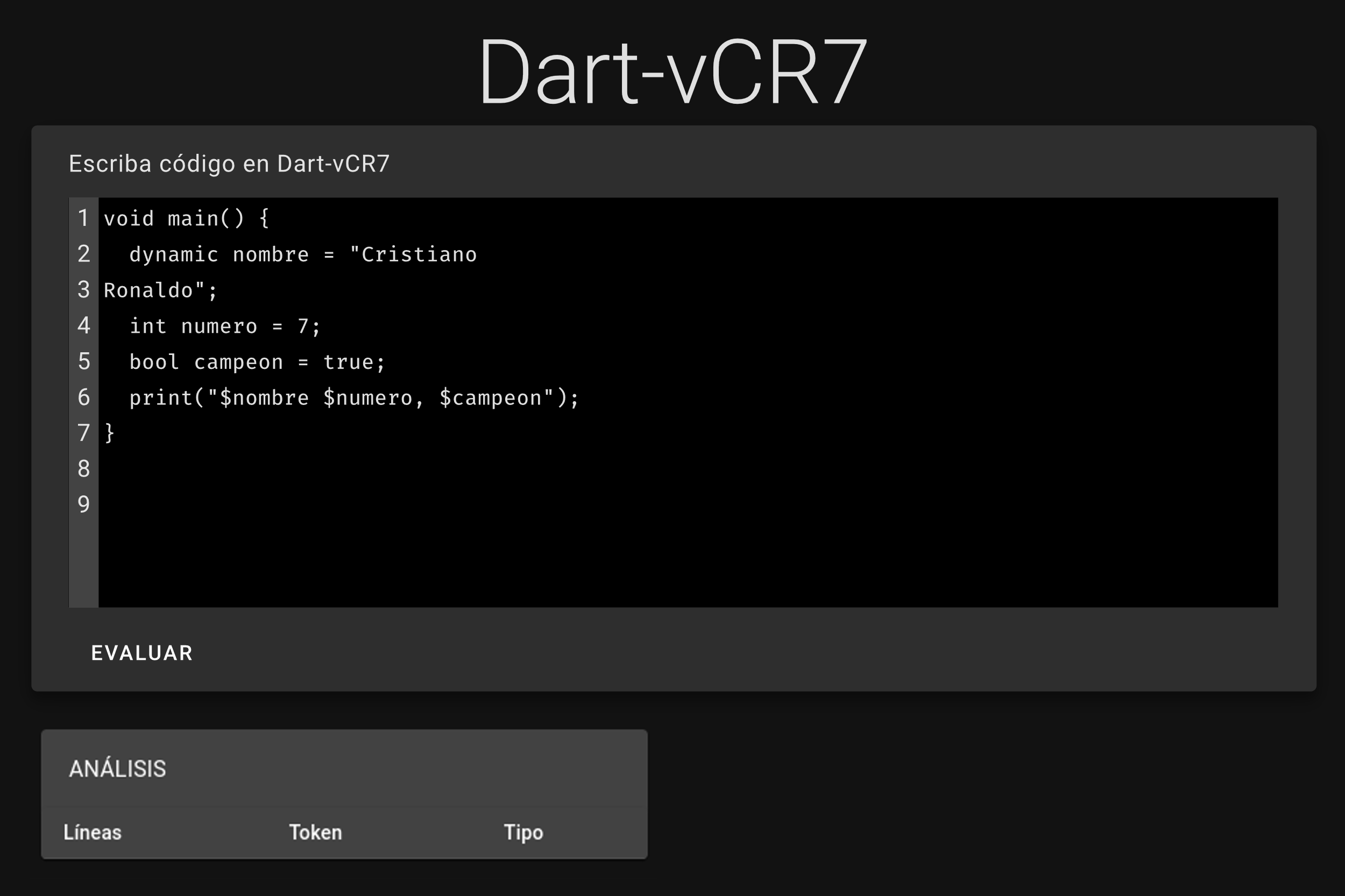
****

Figura 2: Prototipo - 2

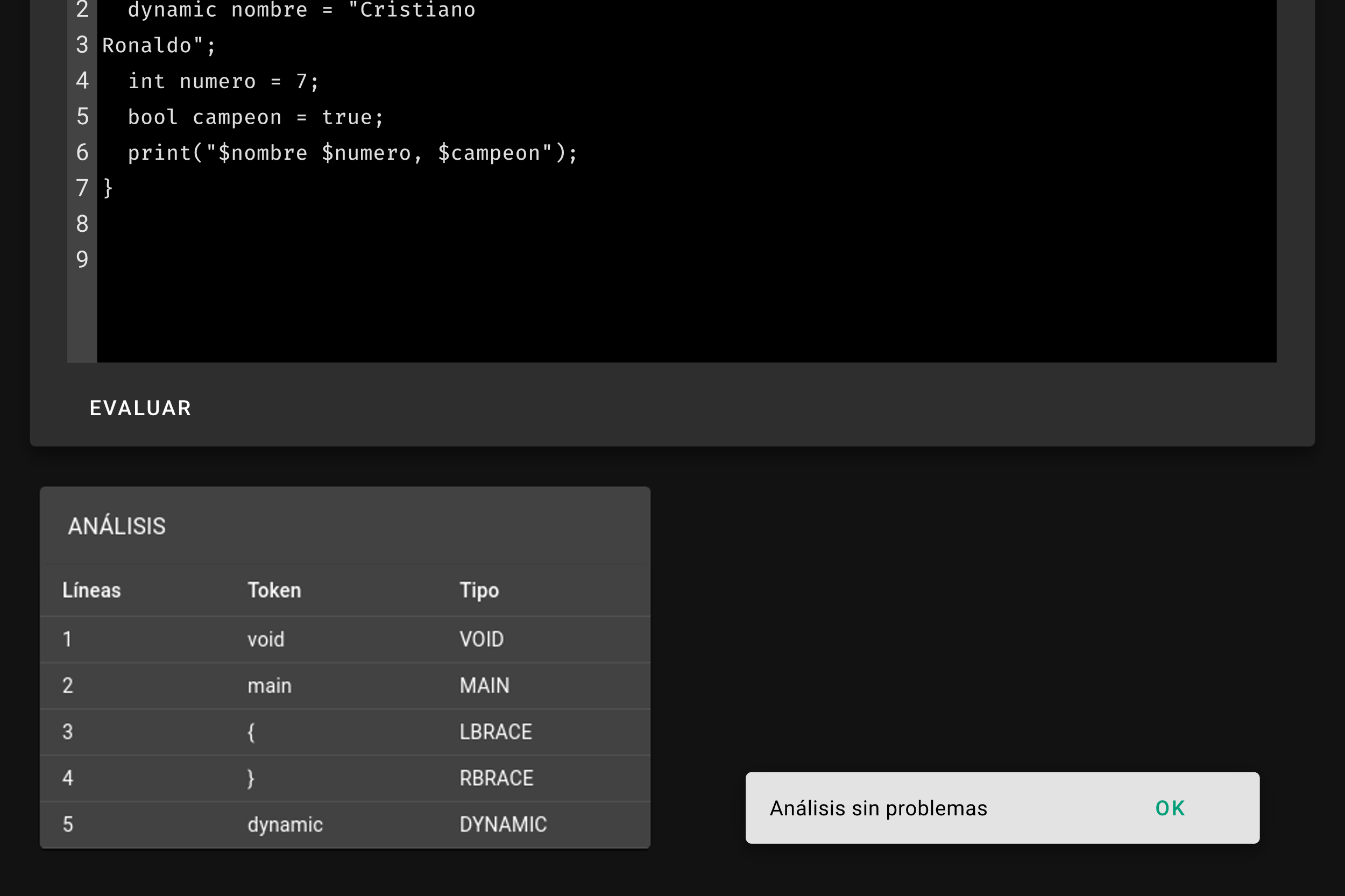


Figure 3: Protitpo - 3

## [Link a Figma](https://www.figma.com/proto/h192NYKgw3bYLJUVOiy6Jz/PROYECTO-PARCIAL-LP-1P?node-id=9%3A2&scaling=contain&page-id=0%3A1)

# Referencias

AEPI Escuela de programacion en Madrid. (14 de June de 2021). *Introducción al lenguaje Dart para Flutter*. Obtenido de Introducción al lenguaje Dart para Flutter: https://asociacionaepi.es/introduccion-al-lenguaje-dart-para-flutter/

Bracha, G. (2015). *The Dart programming language.* Addison-Wesley Professional.

Diví, V. (19 de junio de 2021). *inLab*. Obtenido de inLab: https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart

IONOS. (16 de Octubre de 2020). *Digital Guide IONOS*. Obtenido de Digital Guide IONOS: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/tutorial-de-dart/