

Universidad de San Carlos de Guatemala  
División de Ciencias de la Ingeniería  
Ingeniería en Ciencias y Sistemas  
Organización de Lenguajes y Compiladores 2  
Ing. Moisés Granados



# MANUAL TÉCNICO

Proyecto #1



Carné 201830221  
Willians Alberto Orozco López



## Definición del proyecto

El proyecto consiste en construir un sistema en el cual por medio de un lenguaje de alto nivel se reciba fragmentos de canciones, de notas o clases de ritmos preestablecidos haciendo a este lenguaje de definición de canciones, pistas, ritmos, etc. bastante completo y extendido pudiéndose definir largas pistas musicales sin tanto código involucrado, de una manera clara y definida por métodos locales del propio lenguaje y que se puede transmitir a un sistema consumidor que pueda hacer uso de las mismas.

## Herramientas utilizadas

- JavaFX
- JFlex y Cup
- JDK 8
- Scene Builder
- RichTextFX
- Line Chart FX
- Android Studio
- SDK 21 (lollipop)
- MediaPlayer de Android

## Definición de tokens según lenguaje

### Lenguaje de definición de pistas

Lexema	Token ID	Tipo
PISTA   Pista   pista	PISTA	Palabra reservada
EXTIENDE   Extiende   extiende	EXT	Palabra reservada
entero	ENTERO	Palabra reservada
doble	DECIMAL	Palabra reservada
boolean	BOOLEAN	Palabra reservada
caracter	CHAR	Palabra reservada



cadena	CADENA	Palabra reservada
verdadero   true	TRUE	Palabra reservada
false   falso	FALSE	Palabra reservada
KEEP   Keep   keep	KEEP	Palabra reservada
VAR   Var   var	VAR	Palabra reservada
SI   Si   si	SI	Palabra reservada
SINO   Sino   sino	SINO	Palabra reservada
SWITCH   Switch   switch	SWITCH	Palabra reservada
CASO   Caso   caso	CASO	Palabra reservada
SALIR   Salir   salir	SALIR	Palabra reservada
CONTINUAR   Continuar   continuar	CONT	Palabra reservada
DEFAULT   Default   default	DEFAULT	Palabra reservada
PARA   Para   para	PARA	Palabra reservada
MIENTRAS   Mientras   mientras	MIENTRAS	Palabra reservada
HACER   Hacer   hacer	HACER	Palabra reservada
RETORNA   Retorna   retorna	RETORNA	Palabra reservada
REPRODUCIR   Reproducir   reproducir	REPRODUCIR	Palabra reservada
do   do#   re   re#   mi   mi#   fa   fa#   sol   sol#   la   la#   si	NOTAS	Palabra reservada
ESPERAR   Esperar   esperar	ESPERAR	Palabra reservada
ORDENAR   Ordenar   ordenar	ORDENAR	Palabra reservada
ASCENDENTE   Ascendente   ascendente	ASC	Palabra reservada
DESCENDENTE   Descendente   descendente	DESC	Palabra reservada
PARES   Pares   pares	PAR	Palabra reservada
IMPARES   Impares   impares	IMPAR	Palabra reservada
PRIMOS   Primos   primos	PRIMOS	Palabra reservada



SUMARIZAR   Sumarizar   sumarizar	SUMARIZAR	Palabra reservada
LONGITUD   Longitud   longitud	LONG	Palabra reservada
MENSAJE   Mensaje   mensaje	MSG	Palabra reservada
PRINCIPAL   Principal   principal	PRINCIPAL	Palabra reservada
==	IGUAL	Operador relacional
!=	DIFF	Operador relacional
>	MAYOR	Operador relacional
<	MENOR	Operador relacional
>=	MAYOR_I	Operador relacional
<=	MENOR_I	Operador relacional
!!	NULO	Operador relacional
&&	AND	Operador lógico
!&&	NAND	Operador lógico
	OR	Operador lógico
!	NOR	Operador lógico
&	XOR	Operador lógico
!	NOT	Operador lógico
+	SUMA	Operador aritméticos
-	RESTA	Operador aritméticos
*	POR	Operador aritméticos
/	ENTRE	Operador aritméticos
%	MOD	Operador aritméticos
^	POT	Operador aritméticos
+=	SUMA_S	Operador de incremento
--	DEC	Operadores de decremento
++	INC	Operadores de incremento



[	COR_A	Agrupación
]	COR_C	Agrupación
:	DOS_P	//
,	COMA	//
{	LLAVE_A	Agrupación
}	LLAVE_C	Agrupación
\n	SAL	Salto de línea
\t	TAB	Tabulación
([aA-zZ] "_")([aA-zZ] "_" [\0-9])*	ID	Identificador
\".\."	STRING	String
\'.\'	CARÁCTER	Carácter

## Lenguaje de definición de listas

Lexema	Token ID	Tipo
LISTA   Lista   lista	LISTA	Palabra reservada
LISTAS   Listas   listas	LISTAS	Palabra reservada
NOMBRE   Nombre   nombre	NOMBRE	Palabra reservada
RANDOM   Random   random	RANDOM	Palabra reservada
CIRCULAR   Circular   circular	CIRCULAR	Palabra reservada
[	COR_A	Agrupación
]	COR_C	Agrupación
:	DOS_P	Definición
,	COMA	Numeración
{	LLAVE_A	Agrupación



}	LLAVE_C	Agrupación
---	---------	------------

## Lenguaje de definición de solicitudes

Lexema	Token ID	Tipo
LISTA   Lista   lista	LISTA	Palabra reservada
LISTAS   Listas   listas	LISTAS	Palabra reservada
NOMBRE   Nombre   nombre	NOMBRE	Palabra reservada
TIPO   Tipo   tipo	TIPO	Palabra reservada
PISTA   Pista   pista	PISTA	Palabra reservada
PISTAS   Pistas   pistas	PISTAS	Palabra reservada
SOLICITUD   Solicitud   solicitud	SOLICITUD	Palabra reservada
DURACION   Duracion   duracion	DURA	Palabra reservada
ALEATORIA   Aleatoria   aleatoria	ALEA	Palabra reservada
NOTA   Nota   nota	NOTA	Palabra reservada
CANAL   Canal   canal	CANAL	Palabra reservada
FRECUENCIA   Frecuencia   frecuencia	FRECUENCIA	Palabra reservada
OCTAVA   Octava   octava	OCT	Palabra reservada
DATOS   Datos   datos	DATOS	Palabra reservada
<	INICIO_S	Etiqueta
>	FIN_S	Etiqueta
</	INIF_S	Etiqueta
=	IGUAL	Definición



# Definiciones de gramáticas

## Gramática para definición de pistas

### Bloque de instrucciones

instrucciones	instrucciones instruccion
	instruccion

### Instruccion

instrucción	si
	sino
	sinosi
	asignacion
	declaracion
	llamada
	switch
	para
	hacer

### Sentencia Si, Sino si, Sino

si	SI PAR_A condicion PAR_C INDENT instrucciones DEDENT
sino	SI PAR_A condicion PAR_C INDENT instrucciones DEDENT SINO INDENT instrucciones DEDENT
sinosi	SI PAR_A condicion PAR_C INDENT instrucciones DEDENT SINO instruccion

```
SI (variable1 == 10)
  VAR ENTERO n1 = 10
  VAR ENTERO n2 = 22
  n1 = 50.5 * 20+115*n1
SINO SI (variable2 == 10)
  MENSAJE("un mensaje simple")
SINO
  KEEP VAR ENTERO n3 = 0;
```



## Switch

switch	SWITCH PAR_A ID PAR_C INDENT casos DEDENT
	SWITCH PAR_A ID PAR_C INDENT casos default DEDENT
	SWITCH PAR_A ID PAR_C INDENT default DEDENT
casos	casos caso
	caso
default	DEFAULT INDENT instrucciones DEDENT
caso	CASO valor_puntual INDENT instrucciones DEDENT

```
SWITCH(variable)
  CASO valor1
    VAR ENTERO a1 = 1
    SALIR
  CASO valor 2
    VAR ENTERO a2 = 2
    SALIR
  DEFAULT
    MENSAJE("No coincide");
```

## Para

para	PARA PAR_A asignacion PYC condicion PYC paso PAR_C INDENT instrucciones DEDENT
------	--

```
PARA (ENTERO a = 3; a < 5; a++)
  MENSAJE("Saludando 2 veces")
```

## Mientras

mientras	MIENTRAS PAR_A condicion PAR_C INDENT instrucciones DEDENT
----------	--

```
MIENTRAS (valor1==0)
  MENSAJE("Hola mundo")
  valor1--;
```





## Hacer

hacer	HACER INDENT instrucciones DEDENT MIENTRAS PAR_A condicion PAR_C
-------	--

```
HACER
    MENSAJE("imprimiendo mensaje #"+conteo)
    conteo++
MIENTRAS(conteo == 3)
```

## Declaracion

declaracion	keep VAR tipo ids asignar
	VAR tipo ids asignar
tipo	entero
	doble
	cadena
	boolean
	caracter
ids	ids COMA ID
	ID
asignar	ASIGNAR expresion
	$\lambda$
keep	$\lambda$
	KEEP

## Asignacion

asignación	ID ASIGNAR expresion
------------	----------------------

## Expresion

expresión	expresión operador_aritmetico expresión
	expresion operador_lógico expresión



	expresion operador_relacional expresión
	termino
	NULO ID
	NOT expresion
	MENOS expresion
termino	ID
	ID dimensiones
	BOOLEAN
	NUMERO
	NUMERO_D
	STRING
	CARACTER
	metodos_retorno
	llamada
metodos_retorno	ORDENAR PAR_A arreglo COMA orden PAR_C
	SUMARIZAR PAR_A arreglo PAR_C
	LONGITUD PAR_A arreglo_cadena PAR_C

```
(a4 > 56) || (!!a3) || (!a3!=false) || metodo_retorno_booleano(3,4)
```

## Pista

pistas	pistas pista
	pista
pista	PISTA ID extiende INDENT lista_declaracion DEDENT
lista_declaracion	lista_declaracion declaraciones
	declaraciones



declaraciones	declaracion
	metodo
extiende	$\lambda$
	EXT ids

## Metodo

metodo	KEEP tipo ID PAR_A param PAR_C INDENT instrucciones DEDENT
	tipo ID PAR_A param PAR_C INDENT instrucciones DEDENT
	ID PAR_A param PAR_C INDENT instrucciones DEDENT
	KEEP ID PAR_A param PAR_C INDENT instrucciones DEDENT
	KEEP PRINCIPAL PAR_A param PAR_C INDENT instrucciones DEDENT
	PRINCIPAL PAR_A param PAR_C INDENT instrucciones DEDENT
param	$\lambda$
	parametros
parametros	parametros COMA parametro
	parametro
parametro	tipo ID
ids	ids COMA ID
	ID

## Mensaje

mensaje	MSG PAR_A expresion PAR_C
---------	---------------------------



## Gramática para definición de pistas

### Listas

<b>listas</b>	LLAVE_A LISTA DOS_P LLAVE_A atributos LLAVE_C LLAVE_C
---------------	---

### Atributos

<b>atributos</b>	atributos COMA atributo
	atributo

### Atributo

<b>atributo</b>	NOMBRE DOS_P STRING
	RANDOM DOS_P booleano
	CIRCULAR DOS_P booleano
	PISTAS DOS_P COR_A pistas COR_C

### Pistas

<b>pistas</b>	pistas COMA ID
	ID

### Booleano

<b>booleano</b>	TRUE
	FALSE

```
{
    lista:
    {
        nombre: "lista",
        circular: true,
        random: true,
        pistas: [Pista1,Pista2]
    }
}
```



## Gramática para definición de solicitudes

### Solicitudes

<b>solicitudes</b>	solicitudes solicitud
	solicitud

### Solicitud

<b>solicitud</b>	MENOR_Q SOLICITUD MAYOR_Q parametros FIN SOLICITUD MAYOR_Q
------------------	--

### Parámetros

<b>parametros</b>	tipo nombre data
	tipo

### Data

<b>data</b>	λ
	datos

### Tipo

<b>tipo</b>	MENOR_Q TIPO MAYOR_Q tipos FIN TIPO MAYOR_Q
-------------	---

### Tipos

<b>tipos</b>	PISTANUEVA
	PISTA
	LISTA

### Nombre

<b>nombre</b>	MENOR_Q NOMBRE MAYOR_Q STRING FIN NOMBRE MAYOR_Q
---------------	--



## Datos

datos	datos dato
	dato

## Dato

dato	MENOR_Q DATOS MAYOR_Q listado_datos FIN DATOS MAYOR_Q
------	---

## Listado datos

listado_datos	listado_datos atributo
	atributo

## Atributo

atributo	canal nota octava duracion
----------	----------------------------

## Canal

canal	MENOR_Q CANAL MAYOR_Q NUMERO FIN CANAL MAYOR_Q
-------	--

## Nota

nota	MENOR_Q NOTA MAYOR_Q notas FIN NOTA MAYOR_Q
------	---

## Octava

octava	MENOR_Q OCTAVA MAYOR_Q NUMERO FIN OCTAVA MAYOR_Q
--------	--

## Duración

duracion	MENOR_Q DURACION MAYOR_Q NUMERO FIN DURACION MAYOR_Q
----------	--

## Notas

notas	NOTAS
	SIN



# Diagrama de clases

