MANUAL TÉCNICO

Comunicación Web

Se utilizo la clase socket para realizar las operaciones de comunicaciones y como el fin es mostrar una herramienta que se pueda comunicar entre sí, se utilizó las ip de los dispositivos en que se desarrolló, para el servidor se utilizó la ip 192.186.1.28:8080 y para la aplicación movil se utilizo la ip 192.168.1.40:4555.

Especificaciones técnicas

- Lenguaje JAVA, versión JDK 16
- JCup para herramientas sintácticas
- JFlex para herramientas léxicas
- Clase Socket de java para la comunicación web
- AudioInput para reproducir sonidos
- Reproductor manejado en hilos
- JChart para gráficar

Detalles: cómo se utilizo la red local de internet, cuándo hay internet con pérdidas de paquetes, la aplicación móvil puede fallar, por lo que se recomienda que esté en un lugar con buena conexión.

Gramáticas Léxicas

```
LineTerminator = \r|\n|\r\n
WhiteSpace = {LineTerminator}| [ \t\f]

/*Reserved Words*/
Lista = [lL][i][s][t][a]
Nombre = [nN][o][m][b][r][e]
Random = [rR][a][n][d][o][m]
Circular = [cC][i][r][c][u][l][a][r]
Pistas = [pP][i][s][t][a][s]
True = [tT][r][u][e]
False = [fF][a][l][s][e]
String = [\"][^\"]*[\"]
Simbolo = ([a-zA-Z])([a-zA-Z0-9]|"_")*



    /*Simbolos*/
    "["
```

"]"
"|"
"{"
"}"
":"
","

Lenguaje de Phyton

```
WhiteSpace = [ \r\b\f]
/*Commentary*/
Commentary = >>[^\n]+
BlockCommentary = <-([^-]|-[^>]|-+[^->])*-*->

/*Numbers*/
Number =  (0|([1-9][0-9]*))
Decimal = (([1-9][0-9]*|0?)[.])[0-9]{1,5}
Simbolo = ([a-zA-Z])([a-zA-Z0-9]|"_")*
String = [\"][^\"]*[\"]
Char = [\'][^\']*[\']


/*Condiciones*/
        [sS]"ino"
        [sS]"witch"
        [cC]"aso"
        [sS]"alir"

        [dD]"efault"
        /*Notas*/
        [dD]"o#"
        [rR]"e#"
        [mM]"i#"
        [fF]"a#"
        [sS]"ol#"
        [lL]"a#"
        [sS]"i#"

        [dD]"o"
        [rR]"e"
        [mM]"i"
        [fF]"a"
        [lL]"a"
        [sS]"ol"

        [sS]"i"
```

```
/*Reserved Words*/
[vV]"erdadero"|[tT]"rue"
[fF]"also"|[fF]"alse"
[pP]"ista"
[eE]"xtiende"
[rR]"eproducir"
[eE]"sperar"
[oO]"rdenar"
[aA]"scendente"
[dD]"escendente"
[pP]"ares"
[iI]"mpares"
[pP]"rimos"
[sS]"umarizar"
[lL]"ongitud"
[mM]"ensaje"
[pP]"rincipal"
[kK]"eep"
[vV]"ar"
[eE]"ntero"
[dD]"oble"
[bB]"oolean"
[cC]"aracter"
[cC]"adena"
[aA]"rreglo"
{String}
{Char}
[pP]"rincipal"
[rR]"etorna"


/*Ciclos*/
[pP]"ara"
[mM]"ientras"
[hH]"acer" {
[cC]"ontinuar"
{Simbolo}

/*Operadores relacionales*/
"="
"=="
"!="
">="
"<="
"!!"
">"
"<"
```

```
/*Operadores de Incremento/Decremento */
"+="
"++"
"--"


/*Operadores aritmeticos*/
"^"
"+"
"-"
"*"
"/"
"%"

/*Operadores logicos*/
"!&&"
"&&"
"!||"
"||"
"&!"
"!"

/*Especiales*/
"\t"
"\n"
/*Signos*/
"["
"]"
"{"
"}"
"("
")"
";"
","
```

Lenguaje de comunicación

Solicitud = [sS][o][l][i][c][i][t][u][d]
Tipo = [tT][i][p][o]
Simbolo = ([a-zA-Z])([a-zA-Z0-9]|"_")*
Nombre = [nN][o][m][b][r][e]
Listas = [lL][i][s][t][a][s]
Lista = [lL][i][s][t][a]
Pistas = [pP][i][s][t][a][s]

```
Aleatoria = [aA][l][e][a][t][o][r][i][a]
Duracion = [dD][u][r][a][c][i][o][n]
Canal = [cC][a][n][a][l]
Numero = [nN][u][m][e][r][o]
Nota = [nN][o][t][a]
Frecuencia = [fF][r][e][c][u][e][n][c][i][a]
Datos = [dD][a][t][o][s]
Octava = [oO][c][t][a][v][a]
Pista = [pP]"ista"
Do = [dD]"o"
Do_ = [dD]"o#"
Re = [rR]"e"
Re_ = [rR]"e#"
Mi = [mM]"i"
Mi_ = [mM]"i#"
Fa = [fF]"a"
Fa_ = [fF]"a#"
Sol = [sS]"ol"
Sol_ = [sS]"ol#"
La = [lL]"a"
La_ = [lL]"a#"
Si = [sS]"i"
Si_ = [sS]"i#"
Circular = [cC]"ircular"
Note = {Do}|{Do_}|{Re}|{Re_}|{Mi}|{Mi_}|{Fa}|{Fa_}|{Sol}|{Sol_}|{La}|{La_}|{Si}|{Si_}

/*Datos*/
String = [\"][^\"]*[\"]
Decimal = (([1-9][0-9]*|0?)[.])[0-9]{1,5}
Entero = [0-9]+
```

Gramáticas Sintácticas

Lenguaje de pistas

node -> main_body

main_body -> OPEN_CURLY productions

productions -> LISTA SEMI_COLON parameters

```
//An arrayList of playlist
productions_re -> COMA LISTA SEMI_COLON parameters

                |/*empty*/ {

//Parameters should end with a }
parameters -> OPEN_CURLY status


//Name of the playlist
status -> NOMBRE SEMI_COLON STRING




//Throw an object that goes filling this parameters
new_status -> COMA random
        |COMA circular
        |COMA pista
        |CLOSE_CURLY
                ListaReproduccion lista = new ListaReproduccion(cur_token.left,
cur_token.right)
                RESULT = lista


pista -> PISTA SEMI_COLON look_pista


circular -> CIRCULAR SEMI_COLON options


random -> RANDOM SEMI_COLON options


//ArrayList<String>
look_pista -> OPEN_BRACKET pistas
```

```
//Strings
pistas -> SIMBOLO



//Strings
pistas_re -> COMA SIMBOLO

        |/*empty*/ {


//boolean
options -> TRUE
        |FALSE
```

Lenguaje tipo Phyton

```
kar  KEEP
        |/*empty*/


principal  kar PRINCIPAL OPEN_PARENTHESIS CLOSE_PARENTHESIS tab_re SPACE {:
                RESULT = new Principal(cur_token.left, cur_token.right)

        |kar PRINCIPAL error CLOSE_PARENTHESIS SPACE { esperaba \"(\"","Sintaxis
incorrecta")
        |kar PRINCIPAL OPEN_PARENTHESIS error SPACE { esperaba \")\"","Sintaxis
incorrecta")
        |kar PRINCIPAL OPEN_PARENTHESIS CLOSE_PARENTHESIS error { esperaba
un salto de linea","Sintaxis incorrecta")


/*Declaracion de variables, asignaciones, incrementos, for, while, do_while, if, switch,
funciones especiales*/

/*TAB*/
tab  TAB tab_re



tab_re  TAB tab_re
        |/*empty*/
```

main_body instruction

instruction tab principal instruction

|pista instruction

|tab throw_function tab_re SPACE instruction

|tab function instruction

|tab func_especiales SPACE instruction

|tab v instruction

|tab for instruction

|tab while instruction

|tab do_while instruction

|tab if instruction

|tab switch instruction

|tab EXIT tab_re SPACE instruction

|tab RETORNA value tab_re SPACE instruction

|tab CONTINUE tab_re SPACE instruction

|default instruction

|cases instruction

|tab SPACE instruction
|SPACE instruction
|/*empty*/ { list = new ArrayList() RESULT=list

/*NOTAS MUSICALES*/
notas DO

|RE
|MI
|FA
|SOL
|LA
|SI
|DOR
|RER
|MIR
|FAR
|SOLR
|LAR
|SIR

/*FUNCIONES ESPECIALES*/

ordenar_options ASCENDENTE

|DESCENDENTE

|PARES

|IMPARES

|PRIMOS

reproducir_syntax notas COMA value COMA value COMA value CLOSE_PARENTHESIS

esperar_syntax value COMA value CLOSE_PARENTHESIS

sumarizar_syntax value CLOSE_PARENTHESIS

longitud_syntax value CLOSE_PARENTHESIS

mensaje_syntax value CLOSE_PARENTHESIS

ordenar_syntax value COMA ordenar_options CLOSE_PARENTHESIS

```
                |array_assign COMA ordenar_options CLOSE_PARENTHESIS


func_especiales  REPRODUCIR OPEN_PARENTHESIS reproducir_syntax
                |SUMARIZAR OPEN_PARENTHESIS sumarizar_syntax

                |LONGITUD OPEN_PARENTHESIS longitud_syntax

                |ESPERAR OPEN_PARENTHESIS esperar_syntax

                |MENSAJE OPEN_PARENTHESIS mensaje_syntax

                |ORDENAR OPEN_PARENTHESIS ordenar_syntax




/*FUNCTION*/

kf  KEEP type_function function_mode


        |type_function function_mode

        |KEEP


//Para arreglos
function_mode  OPEN_BRACKET CLOSE_BRACKET function_mode

                |/*empty*/


function  kf word_functions OPEN_PARENTHESIS  multiple_v tab_re  SPACE



type_function  data_type



multiple_v  function_variables multiple_v_re

        |CLOSE_PARENTHESIS
```

```
multiple_v_re  COMA function_variables multiple_v_re
               |CLOSE_PARENTHESIS



function_variables  data_type SIMBOLO
               |data_type ARREGLO SIMBOLO function_mode


/*LLAMAR FUNCIONES*/
method_value  condition_re method_value_re


               |CLOSE_PARENTHESIS

               |array_assign method_value_re
               i


method_value_re  COMA condition_re method_value_re
               |COMA array_assign method_value_re
               |CLOSE_PARENTHESIS{:
               ArrayList<Operation> lista = new ArrayList()
               RESULT = lista



word_functions  SIMBOLO
               |REPRODUCIR
               |ESPERAR
               |ORDENAR
               |LONGITUD
               |MENSAJE


throw_function  word_functions OPEN_PARENTHESIS method_value


/*PISTA*/
pista  PISTA SIMBOLO d


d  EXTIENDE SIMBOLO d_

          |tab_re SPACE
```

```
d_   COMA SIMBOLO d_
          | tab_re SPACE


/*INCREMENTACION DE VARIABLES*/
increm_variables  SIMBOLO increm

                    |SIMBOLO



increm  INCREASE { = "++"
          |DECREASE { = "--"
          |error


/*Variables for*/

v_for  type SIMBOLO assign_for

          |SIMBOLO assign_for


assign_for  equal condition_re



/*VARIABLES*/

v  type variables assign_special /*Declaracion*/
          |variables assign /*Solo asignacion*

          |variables increm SPACE



data_type  ENTERO
          |DOBLE
          |BOOLEAN
          |CADENA
          |CARACTER


variables  SIMBOLO variables_re
```

```
                |ARREGLO SIMBOLO variables_re dimension

                |SIMBOLO dimension


variables_re  COMA SIMBOLO variables_re


k       KEEP VAR
        |VAR



type  k data_type

        |k



/*ASIGNAR VALORES*/
assign_special  assign
                |tab_re SPACE


assign  equal condition_re assign_special

        |EQUAL array_assign tab_re SPACE{:



equal  EQUAL
        |EQUAL_MORE


//Depricated
array  ARREGLO dimension
        |/*empty*/


dimension   OPEN_BRACKET value CLOSE_BRACKET  dimension_re


dimension_re   OPEN_BRACKET value CLOSE_BRACKET dimension_re
```

|/*empty*/

/*Asignacion de valores a arreglos*/
/*PENDIENTE*/
/*{{5,10,15},{1,3,8},{20,25,30}}*/

array_assign OPEN_CURLY assign_arr CLOSE_CURLY// nD

|OPEN_CURLY mult_values CLOSE_CURLY //1D

//Genera una lista de Dimensiones de grado 1
new_assign CLOSE_CURLY COMA OPEN_CURLY mult_values new_assign

|CLOSE_CURLY

assign_arr OPEN_CURLY assign_arr assign_arr_re {: //Como es la parte final le damos vuelta a los arreglos siempre

|OPEN_CURLY mult_values new_assign {: //SOlo para dimensiones de grado 1

assign_arr_re CLOSE_CURLY COMA OPEN_CURLY assign_arr assign_arr_re

|CLOSE_CURLY{:

mult_values value mult_values_re

mult_values_re COMA value mult_values_re

```
        /*VALORES*/
value  NUMBER

        |MINUS NUMBER

        |DECIMAL

        |MINUS DECIMAL

        |STRING

        |CHAR

        |throw_function

        |value ADD value

        |value MINUS value

        |value MULTIPLY value

        |value DIV value

        |value POW value

        |value MODULE value

        |OPEN_PARENTHESIS value CLOSE_PARENTHESIS
        |SIMBOLO dimension_re

        |TRUE
        |func_especiales

        |FALSE

/*SI*/

if  block_if
        |block_elseif
        |block_else


block_if  SI OPEN_PARENTHESIS condition tab_re SPACE


block_elseif  ELSE SI OPEN_PARENTHESIS condition tab_re SPACE
```

block_else ELSE tab_re SPACE

/*CONDICION*/
/*this production end in close_parenthesis*/

condition condition_re CLOSE_PARENTHESIS

condition_re condition_re AND condition_re

|condition_re NAND condition_re

|condition_re OR condition_re

|condition_re NOR condition_re

|condition_re XOR condition_re

|OPEN_PARENTHESIS condition_re CLOSE_PARENTHESIS
|NOT condition_re {

|comparation

comparation comparation EQUALIZATION comparation

|comparation DIFFERENTIATION comparation

|comparation GREATER comparation

|comparation GREATER_THAN comparation

|comparation LESSER comparation

|comparation LESSER_THAN comparation

|OPEN_PARENTHESIS comparation CLOSE_PARENTHESIS{ = e1

|NULL_ value

|value

/*SWITCH*/

switch SWITCH OPEN_PARENTHESIS value CLOSE_PARENTHESIS tab_re SPACE

default tab DEFAULT tab_re SPACE

cases tab CASE value tab_re SPACE

/*FOR*/

for FOR OPEN_PARENTHESIS for_instructions SPACE

for_instructions v_for COLON for_condition

for_condition condition_re COLON for_increm

for_increm increm_variables CLOSE_PARENTHESIS tab_re

/*WHILE*/

while WHILE OPEN_PARENTHESIS condition tab_re SPACE{

/*DO WHILE*/

do_while HACER tab_re SPACE

Comunicación de Lenguaje

main_body -> solicitudes

|body

body -> lista_listas
|lista_estructura

|lista_pistas

|pista_estructura

|crear_solicitud

//Para realizar lista de pistas

lista_pistas -> etiqueta_pista pista_lista_

pista_lista_ -> LESS PISTA params_pista_lista pista_lista_

|etiqueta_pista_cl

etiqueta_pista -> LESS PISTAS GREATER

etiqueta_pista_cl -> LESS DIV PISTAS GREATER

//Para realizar estructura de pista

open_lista -> LESS PISTA NOMBRE EQUAL STRING GREATER

close_pista -> LESS DIV PISTA GREATER

pista_estructura -> open_lista canales

canales -> canal canales
|close_pista

open_canal -> LESS CANAL NUMERO EQUAL NUMBER GREATER

close_canal -> LESS DIV CANAL GREATER

canal -> open_canal notas

notas -> nota notas

|close_canal

nota -> LESS NOTA DURACION EQUAL NUMBER FRECUENCIA EQUAL valor GREATER

valor -> DECIMAL

|NUMBER

//Para realizar estructura de listas

lista_estructura -> fact_lista pista_lista

fact_lista -> LESS LISTA params_lista

pista_lista -> LESS PISTA params_pista_lista pista_lista

|etiqueta_lista_cl

params_pista_lista -> param_nombre params_pista_lista

|param_duracion params_pista_lista

|GREATER

params_lista -> param_nombre params_lista

|param_random params_lista

|param_circular params_lista

|GREATER

etiqueta_lista_cl -> LESS DIV LISTA GREATER

//Fin Listas

//Etiquetas

//para realizar lista_listas
lista_listas -> etiqueta_lista listas_re

listas_re -> throw_lista listas_re

|etiqueta_listas_cl

etiqueta_lista -> LESS LISTAS GREATER

etiqueta_listas_cl -> LESS DIV LISTAS GREATER

throw_lista -> LESS LISTA param_nombre param_pistas GREATER

param_nombre -> NOMBRE EQUAL STRING

param_pistas -> PISTAS EQUAL NUMBER

param_random -> ALEATORIA EQUAL STRING

param_duracion -> DURACION EQUAL NUMBER

param_circular -> CIRCULAR EQUAL STRING

//FIN etiquetas

//Para realizar peticiones

solicitudes -> solicitud

solicitud -> etiqueta_solicitud lista

lista -> tipo nombre

tipo -> etiqueta_tipo tipos etiqueta_tipo_cl

tipos -> LISTA
|PISTA

nombre -> solicitud_nombre nombre

|etiqueta_solicitud_cl

solicitud_nombre -> etiqueta_nombre STRING etiqueta_nombre_cl

|etiqueta_nombre error etiqueta_nombre_cl

//Creacion de pistas

crear_solicitud -> etiqueta_solicitud big_data

big_data -> throw_tipo big_data_re

big_data_re -> container_data big_data_re
|etiqueta_solicitud_cl

container_data -> etiqueta_datos container_big_data etiqueta_datos_cl

container_big_data -> throw_canal throw_nota throw_octava throw_duracion

etiqueta_datos -> LESS DATOS GREATER

etiqueta_datos_cl -> LESS DIV DATOS GREATER

throw_duracion -> etiqueta_duracion NUMBER etiqueta_duracion_cl

etiqueta_duracion -> LESS DURACION GREATER

etiqueta_duracion_cl -> LESS DIV DURACION GREATER

throw_octava -> etiqueta_octava NUMBER etiqueta_octava_cl

etiqueta_octava -> LESS OCTAVA GREATER

etiqueta_octava_cl -> LESS DIV OCTAVA GREATER

throw_nota -> etiqueta_nota NOTE etiqueta_nota_cl

etiqueta_nota -> LESS NOTA GREATER

etiqueta_nota_cl -> LESS DIV NOTA GREATER

throw_canal -> etiqueta_canal NUMBER etiqueta_canal_cl

etiqueta_canal -> LESS CANAL GREATER

etiqueta_canal_cl -> LESS DIV CANAL GREATER

throw_tipo -> etiqueta_tipo SIMBOLO etiqueta_tipo_cl

//Fin solicitudes

etiqueta_solicitud -> LESS SOLICITUD GREATER

etiqueta_solicitud_cl -> LESS DIV SOLICITUD GREATER

etiqueta_tipo -> LESS TIPO GREATER

etiqueta_tipo_cl -> LESS DIV TIPO GREATER

etiqueta_nombre -> LESS NOMBRE GREATER

etiqueta_nombre_cl -> LESS DIV NOMBRE GREATER