# Visión del Proyecto/Propósito

Realizar un compilador para un lenguaje gráfico que permita a los usuarios programar música mediante la selección la utilización de estructuras de control y elementos musicales, todo con el fin de aprender a programar de una manera más interactiva y dinámica mediante la exploración de composiciones musicales.

# Objetivo del Lenguaje

El objetivo del lenguaje es enseñar programación a niños de secundaria mediante la utilización de bloques gráficos para que puedan entender estructuras de control, lógica de programación y que al mismo tiempo exploten su lado creativo que sería la producción de obras musicales o imitación de las mismas mediante estructuras de control.

# Requerimientos del Proyecto

## Componentes del Léxico

El lenguaje estará basado en la librería de blockly de google la cual tomaremos como base para utilizar y extenderemos agregando bloques para producción de notas, blockly ya bloques que engloba el léxico y semántica necesarios para poder funcionar. De momento blockly cuenta con:

* Lógica: if do, comparación (equals, gt,gte,lt,lte, notequals), ands y ors de expresiones lógicas, nots, y switches que en su caso se llaman tests.
* Loops: Repeat n veces, repeat mientras que exista cierta condición (while), count de n a m sumando el iterador por cierta cantidad (for), for each para una lista de objetos y un break para salirse de cualquier loop.
* Matemáticas: Maneja enteros y dobles en uno de sus bloques a los cuales se le puede sumar, restar, dividir, multiplicar y exponenciar, los bloques de operaciones binarias pueden recibir más bloques de operaciones binarias dentro de ellos. Al igual existen bloques para hacer expresiones unitarias los bloques de números que pueden contener un resultado de una operación binaria a su vez. Existen condiciones matemáticas y generación de números aleatorios, todo es con números de tipo flotantes o enteros.
* Texto: Se puede generar texto, concatenar texto, buscar dentro del texto, revisar la longitud del texto, búsquedas dentro del texto, e impresiones de texto.
* Listas: Se pueden crear listas de ítems que toman en cuenta variables de bloques, se pueden manipular las listas y buscar dentro de ellas.
* Variables: Se pueden crear y asignar variables.
* Funciones: Existen funciones que reciben parámetros y regresan un valor o existen funciones que reciben parámetros y no regresan nada. Existen condicionales que regresan un valor en específico.

Se busca extender este léxico a crear nuevos bloques:

* Melodía: Este bloque va a declarar el tempo que va a llevar la melodía que básicamente es que tan rápido se va a tocar la canción.
* Nota: Este bloque recibe a la nota y el tiempo que se tocará la nota.
* Tono: Este bloque permitirá escoger las notas a tocar de acuerdo a la llave de sol (do, re, mi, fa, sol, la, si y vacío).

## Diagramas de Sintaxis

<MELODÍA>

BLOQUE

FOR EACH

PRINT

<BLOQUE >

NOTA

SETVAR

IF

REPEAT N TIMES

REPEAT WHILE

COUNT WITH

IN LIST SET

EXPRESION

<SETVAR>

<EXPRESION >

EXPRESION LOGICA

EXPRESION MATEMÁTICA

EXPRESION TEXTO

EXPRESION LISTA

<EXPRESION LOGICA>

COMPARACION

CONJUNCION LOGICA

NOT

TEST

IS EMPTY

<COMPARACION>

EXPRESION

EXPRESION

<CONJUNCION LOGICA>

EXPRESION LOGICA

EXPRESION LOGICA

<NOT>

EXPRESION LOGICA

<TEST>

EXPRESION LOGICA

EXPRESION

EXPRESION

<EXPRESION MATEMATICA>

OP BINARIA

OP UNITARIA

OP LISTA

LENGTH

<OP BINARIA>

EXPRESION MATEMATICA

EXPRESION MATEMATICA

<OP UNITARIA>

EXPRESION MATEMATICA

<OP LISTA>

EXPRESION LISTA

<EXPRESION TEXTO>

CREATE TEXT

APPEND

FIRST OCURR

SUBSTR

<CREATE TEXT>

EXPRESION

<APPEND>

<LENGTH>

EXPRESION TEXTO

EXPRESION LISTA

<FIRST OCURR>

EXPRESION TEXTO

EXPRESION TEXTO

<SUBSTR>

EXPRESION TEXTO

EXPRESION MATEMATICA

EXPRESION MATEMATICA

<EXPRESION LISTA>

CREATE LIST

GET

<CREATE LIST>

EXPRESION

<IS EMPTY>

EXPRESION LISTA

EXPRESION TEXTO

<GET>

EXPRESION LISTA

EXPRESION MATEMATICA

<IN LIST SET>

EXPRESION LISTA

EXPRESION MATEMATICA

EXPRESION

<IF>

EXPRESION LÓGICA

BLOQUE

<REPEAT N TIMES>

EXPRESION MATEMATICA

BLOQUE

<REPEAT WHILE>

EXPRESION LOGICA

BLOQUE

<COUNT WITH>

EXPRESION MATEMATICA

EXPRESION MATEMATICA

EXPRESION MATEMATICA

BLOQUE

<FOR EACH>

EXPRESION LISTA

BLOQUE

<PRINT>

EXPRESION

< NOTA>

## Principales Características Semánticas

En el caso de utilizar variables en aquellas funciones que estén esperando un valor que no sea del mismo tipo al esperado de entrada entonces se marcará un error al momento de compilar el programa.

Solamente se utilizarán números de tipo flotante y enteros, la duración de las notas a tocar van de acuerdo a la estructura musical y se multiplican por el tempo que se está llevando. No son segundos ni enteros los que se utilizarán en la duración de las notas sino que irán de acuerdo con la librería musical de jython. Cualquier nota que se ejecute fuera de la melodía es como si nunca se hubiera tocado y no se escuchará en la canción, las notas se tocan de acuerdo al tono que se eligió y si se quiere realizar una pausa se tendrá que incluir el tono vacío.

El output de sonido se hará de acuerdo a estándares MIDI y siguiendo la librería de notas presentadas en jython music.

## Descripción de Funciones Especiales del Lenguaje

Todas las funciones que utilizamos son las incluidas en blockly, solamente agregaremos los bloques musicales los cuales nos permitirán utilizar las estructuras de control para crear música. La melodía denota cuando comienza la producción musical y dentro de ella pueden venir bloques lógicos, una vez que se ingresa el primer bloque de nota entonces comienza el output de sonido, la melodía va avanzando de acuerdo se vayan incluyendo más notas y de acuerdo a su duración con el tempo es lo que tomará como total a toda la composición musical cuando se ejecuta el programa.

## Tipos de Datos

* Booleanos
* Enteros
* Flotantes
* Listas
* Tonos
* Duraciones
* Tempos
* Instrumentos