Proyecto de algoritmia y programación: Simulador de circuitos

Cristian Fernando Libreros Osorio, Ryan Eduardo Ramos.
Universidad del Valle sede Zarzal, Valle del Cauca, Colombia.
libreros.cristian@correounivalle.edu.co, ryan.ramos@correounivalle.edu.co
Programa:Algoritmia y programación
Profesor:Paul Angarita Jimenez

***Abstract*—In this document you will notice how an algorithm performed in Pseint works that although it does not specifically solve a job, it can be used in some circuits.**

I. Introducción

La idea del proyecto "simulador de circuitos" surgió debido a que se llegó a la conclusión de que muchas personas prefieren comprarse un buen dispositivo movil que un buen computador, o sencillamente, en algunos hogares se encuentra más fácil un dispositivo móvil que un computador. Debido a esto decidimos optar por una aplicación portable donde se pueda armar circuitos basicos y se puedan calcular algunas medidas basicas como voltaje, corriente, resistencia equivalente, potencia etc.

Julio 30, 2020

II. simulador de circuitos para dispositivos móviles

Para este proyecto hemos tenido algunas opciones de lengüaje que se utilizan para programar en dispositivos móviles android, siendo mas precisos hablaríamos de el famoso java que es uno de los lengüajes oficiales de android y uno de los mas mencionados o reconocidos.Otro lengüaje que aunque es un poco desconocido para las personas menos expertas en el ámbito de la programación, pero que igualmente se utiliza para programar en dispositivos móviles es el kotlin. Uno de los objetivos de este proyecto es llegar a las personas que desean un simulador portable, que funcione correctamente y esté optimizado para la mayoría de dispositivos, ya sea de gama alta o gama baja.
A continuación se explicarán los dos lengüajes que tenemos como opción a utilizar en nuestro proyecto.

A. Java

Es el lenguaje nativo que usa Android. Cualquier aplicación que use directamente el hardware y se comunique con el sistema operativo, usará este código. Es un lenguaje muy extendido, presente en muchos dispositivos, tanto por la red como a nivel local.

B. Kotlin

Este lenguaje formal tiene un tipado estático y puede ser utilizado en servidores, en sitios web y en el sistema operativo de Apple (iOS), siendo uno de los oficiales para desarrollar aplicaciones Android, como lo reconoció Google en el año 2017.

Kotlin se ejecuta en la Máquina Virtual de Java (JMV) y es interoperable con Javascript y frente a otros lenguajes de programación reduce la repetición de código, lo que a su vez ahorra recursos y tiempo, facilitando la localización de errores en caso de que se produzcan.

Además de la reducción de líneas de código, que se estima aproximadamente en un 40 porciento con respecto a otros lenguajes, Kotlin gestiona los nulos de forma segura, de tal forma que no se van a producir los Null Pointer Exception (NPE), aunque si se necesita esta característica de nulabilidad, bastará con añadir "?" tras el nombre del tipo.

Fundamentalmente tiene construcciones orientadas a objetos también permite construcciones de funciones simples y más complejas y puede mezclar elementos de ambos estilos.

Sin embargo para ello debemos saber como funciona cada uno de esos programas, como es su estilo, como se usa y demás, así que mientras tanto diseñaremos un prototipo en Pseint cuyo objetivo es realizar unos algoritmos que se implementen en nuestro proyecto que ayuden a calcular como decíamos anteriormente el voltaje, la corriente, etc... Para ello diseñamos un pequeño algoritmo que sería los divisores de voltaje, divisores de corriente para incluirlos después en nuestor proyecto pero en el lenguaje que vayamos a usar en Java y/o Kotlin
Ahora presentaré unas imágenes de nuestro pequeño algoritmo como funcionaría

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).
J. Doe and J. Doe are with Anonymous University.
Manuscript received April 19, 2005; revised August 26, 2015.

```
Algoritmo DivisorDeVoltaje
    Definir V,R1,R2 Como Real;
    Definir DV Como Real;
    Escribir 'Ingrese valor de la fuente V : ' ;
    Leer V;
    Escribir 'Ingrese valor de R1 : ' ;
    Leer R1;
    Escribir 'Ingrese valor de R2 : ' ;
    Leer R2;
    //El voltaje que se va hallar en este caso es en R1, si quiere hallarlo en R2 simplemente debe reemplazar el valor de R1 por el de R2 en el numerador//
    DV <- ((V*R1)/(R1+R2));

    Escribir 'El voltaje es : ',DV;


FinAlgoritmo
```

Fig. 1. Estructura del algoritmo

El primer algoritmo que implementaríamos sería un divisor de voltaje donde nos va a calcular el voltaje en la resistencia que nosotros queramos en este caso el voltaje en a resistencia uno, he de decir que este contiene un mensaje donde se aclara que el voltaje que se va a hallar es en R1.
Y el funcionamiento de nuestro algoritmo es el siguiente:

```
PSeInt - Ejecutando proceso DIVISORDEVOLTAJE
*** Ejecución Iniciada. ***
Ingrese valor de la fuente V :
> 20
Ingrese valor de R1 :
> 330
Ingrese valor de R2 :
> 220
El voltaje es : 12
*** Ejecución Finalizada. ***
```

Fig. 2. Divisor de voltaje

El siguiente algoritmo que usaríamos es el divisor de corriente que es importante para calcular la corriente que pasará a través de un elemento.

```
Algoritmo DivisorDeVoltaje
    Definir IR,R1,R2 Como Real;
    Definir DI Como Real;
    Escribir 'Ingrese valor de la corriente entrada/corriente total IR : ' ;
    Leer IR;
    Escribir 'Ingrese valor de R1 : ' ;
    Leer R1;
    Escribir 'Ingrese valor de R2 : ' ;
    Leer R2;
    //La corriente que se va hallar en este caso es en R1, si quiere hallarlo en R2 simplemente debe reemplazar el valor de R1 por el de R2 en el numerador//
    DI <- ((IR*R1)/(R1+R2));

    Escribir 'La corriente es : ',DI;


FinAlgoritmo
```

Fig. 3. Estructura del divisor de corriente

Estos 2 divisores son demasiado importantes para desarrolar en nuestro proyecto ya que como se van a armar/crear circuitos se va a necesitar medir cuanta tensión y corriente pasa o hay en cada componente del mismo circuito.

Vamos a probar nuestro algoritmo para saber que funciona correctamente

```
PSeInt - Ejecutando proceso DIVISORDEVOLTAJE
*** Ejecución Iniciada. ***
Ingrese valor de la corriente entrada/corriente total IR :
> 10
Ingrese valor de R1 :
> 330
Ingrese valor de R2 :
> 1000
La corriente es : 2.4812030075
*** Ejecución Finalizada. ***
```

Fig. 4. Divisor de corriente

Hago una aclaración aquí, es que usé el pseint para explicar estos algoritmos ya que es mucho más fácil entender por pseint que da como un toque agradable y visual en el código, de todas formas este algoritmo se implementará en su respectivo lenguaje ya sea usando el Java o el Kotlin.

Lo siguiente a realizar en nuestro proyecto es un algoritmo donde se usa X fórmula para hallar Voltaje, corriente, resistencia y potencia. Es decir que se pueda calcular el voltaje usando la ley de ohm .
Claramente para poder usar estos algoritmos debemos primero tener y conocer los OBJETOS y/o que el usuario va a poder utilizar como fuentes AC Y DC, resistencias, diodos, capacitores, etc etc.
Donde cada uno de los elementos que se puedan utilizar tendrán su propio código es decir en un menú aparecerán los elementos que podemos utilizar para nuestros circuitos en el cuál en la descripción de cada uno tendrá su nombre respectivo, su código y una pequeña teoría sobre su funcionamiento.
Cabe resaltar también que el usuario que desee realizar sus circuitos debería tener un poco de conocimiento acerca de circuitos (valga la redundancia) porque tendrá la libertad de usar los componentes que él desee sin embargo si alguno está mal colocado, claramente el resultado no será el esperado.
Aunque también se podría configurar para que en tal caso no se permita conectar el componente debido a que está mal.