Ambiente de simulación sobre señales, sistemas y control dinámico en dispositivos móviles

Andrés Felipe Jiménez Vega

20441717505

Universidad Antonio Nariño

Programa Ingeniería Electrónica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Ibagué, Colombia

2022

Ambiente de simulación sobre señales, sistemas y control dinámico en dispositivos móviles

Andrés Felipe Jiménez Vega

Proyecto de grado presentado como requisito parcial para optar al título de:
Ingeniero Electrónico

Director (a):
Ph.D Sergio Alejandro Orjuela

Línea de Investigación:
Automatización y control
Grupo de Investigación:
Gepro

Universidad Antonio Nariño

Programa Ingeniería Electrónica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Ibagué, Colombia

2022

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

_____,

Cumple con los requisitos para optar

Al título de _____.

_____

Firma del Tutor

_____

Firma Jurado

_____

Firma Jurado

Ciudad, Día Mes Año.

Contenido

Pág.

Lista de Figuras

Lista de Tablas

Preliminares

(Dedicatoria)

El presente trabajo integral de grado lo dedico a todas las personas quienes fueron participes de esta investigación. A mi familia que gracias a su apoyo pude concluir mi carrera. A mi padre y a mi abuelita por brindarme los recursos necesarios y estar a mi lado apoyándome y aconsejándome siempre. A mi madre por hacer de mí una mejor persona mediante sus consejos, apoyo y amor. A mis hermanos que siempre estuvieron en este proceso. Por último, al PhD. Sergio Alejandro Orjuela por brindarme de su tiempo y sabiduría para la ejecución de esta investigación.

Agradecimientos

A mi familia por brindarme los recursos y medios adecuados para lograr esta carrera.

A mis amigos que siempre me apoyaron y estuvieron en las buenas y en las malas.

Al PhD. Sergio Alejandro Orjuela, por impartirme gran parte de su conocimiento.

Resumen

Este Trabajo Integral de Grado (TIG) hace parte de un proyecto modular de la Fundación Conciencia Activa, el cual tiene como objetivo de una aplicación para dispositivos móviles empleando el framework Kivy de código abierto de Python. El motivo principal para su desarrollo es brindar a los estudiantes de ingeniería electrónica la posibilidad de consolidar y ampliar conocimientos a través de esta herramienta de ambiente de estudio. Se tuvo en cuenta su ejecución móvil debido a que la mayoría de los estudiantes cuentan con un teléfono funcional y compatible. La aplicación será desarrollada en un ambiente de simulación el cual ofrece interfaces gráficas de dibujo, comunicación e interacción a cerca de señales, sistemas y control dinámico por medio de la librería Control System Library que el lenguaje Python ofrece, proporcionando diversas funciones para el análisis y diseño de sistemas de control, utilizando la iniciativa CDIO, así como la metodología Scrum en el proceso y tomando en consideración las múltiples fuentes de información como libros y artículos acerca del avance de la tecnología móvil, el desarrollo de aplicaciones en el framework Kivy.

Palabras clave: framework kivy, Control System Library, Python, Iniciativa CDIO.

Abstract

This Integral Degree Project (TIG) is part of a modular project of Fundación Conciencia Activa, which aims to develop an application for mobile devices using the Python open source Kivy framework . The main reason for its development is to provide electronic engineering students with the possibility of consolidating and expanding their knowledge through this study environment tool. Its mobile execution was taken into account because most of the students have a functional and compatible phone. The application will be developed in a simulation environment which offers graphical interfaces for drawing, communication and interaction about signals, systems and dynamic control through the Control System Library that the Python language offers, providing various functions for the analysis and design of control systems, using the CDIO initiative, as well as the Scrum methodology in the process and taking into consideration the multiple sources of information such as books and articles about the advancement of mobile technology, application development in the Kivy framework.

Keywords: framework kivy, Control System Library, Python, CDIO Initiative.

Introducción

La educación moderna requiere fortalecer contenidos considerados en clase empleando las tecnologías que brindan los avances recientes. En el caso de los móviles, se lleva a cabo por medio de procesadores ágiles que pueden ser una herramienta de apoyo de suma importancia si se potencia su empleabilidad hacia el campo educacional.

La temática principal de este TIG envuelve la necesidad de sustentar y fortalecer el conocimiento e implementarlo a los avances tecnológicos de hoy en día, incorporando la programación científica mediante el lenguaje de programación Python, y así estar al día con la evolución de las tecnologías de la información y la comunicación (TIC). Para este caso, se plantea el uso de dispositivos móviles por parte de estudiantes de control y se sugiere por tanto el desarrollo de una aplicación móvil a través de la plataforma kivy para asignaturas base de carácter complejo de las cuales se selecciona señales, sistemas y control por su crucial empleo en la carrera de ingeniería electrónica y por su afinidad con procesos numéricos, esta pretende afianzar los conocimientos en estas áreas fundamentales sin estar conectado a una red WiFi, e implementando la metodología de desarrollo SCRUM y la iniciativa para la ruta didáctica CDIO.

El documento se presenta de la siguiente manera, la justificación contesta la pregunta del por qué se lleva a cabo la investigación y quienes se encuentran de manera directa e indirectamente beneficiados con esta, de igual manera, en el marco teórico se plantea la teoría fundamental, la cual sustenta ampliamente la investigación, sirviendo de referencia para enriquecer y completar el proyecto.

1. Objeto de estudio

1.1 Antecedentes

El uso de simuladores interactivos para la enseñanza virtual y sus características, permite desde un principio representar un cambio en el entorno de la enseñanza y aprendizaje mediante la modelización de casos de la vida real. Por lo tanto, las necesidades y retos a los cuales se enfrenta la educación generan la obligación de diseñar mejores estrategias utilizando las herramientas que nos proporciona el m-learning. De esa manera, muchas estrategias educativas han adoptado modelos de aprendizaje que hacen uso de las (TIC) para reforzar el proceso de conocimiento:

Artículos internacionales muestran que en el año 2016 en Praga – República Checa se realizó la 17th International Conference on Mechatronics – Mechatronika (ME). Dentro de todas sus exposiciones se habla acerca de un Sistema PSE un ambiente de simulación en tiempo real de control para sistemas mecatrónicos desarrollado en el lenguaje de programación orientado a objetos Python usando el framework Kivy y las librerías NumPy y SciPy las cuales permiten resolver sistemas lineales, no lineales usando bloques similares al ambiente de Simulink, pero en dispositivos móviles (Fabo, 2016).

En Argentina, la Universidad Nacional de la Plata, en la facultad de informática, desarrolló una plataforma llamada Lihuen, basada en el sistema operativo Debian, bajo la licencia de distribución GNU. En la cual se utilizó el framework kivy para desarrollar aplicativos multiplataforma con Python para la simulación de aplicaciones educativas, este incluye matemáticas, química, idiomas, entre otros, con propósito de potencializar el estudio en áreas a fines (Lihuen, 2022).

1.2 Objetivo general

Desarrollar una aplicación para dispositivos móviles, haciendo uso de herramientas de programación científica, para simular ejercicios de señales, sistemas y control dinámico.

1.2.1 Objetivos específicos

- Establecer una ruta didáctica para implementar la simulación sobre señales, sistemas y control, diseñando una interfaz amigable para la compresión de cada uno de los pasos para la simulación de cada ejercicio.
- Desarrollar algoritmos para la operación de métodos numéricos necesarios para simular ejercicios básicos de señales y sistemas, seleccionados de libros contemplados en la base de datos de la universidad Antonio Nariño.
- Diseñar prácticas de laboratorio para la utilización de la aplicación desarrollada acorde con los resultados de aprendizaje del programa.
- Crear una guía para la implementación de simulaciones de asignaturas como señales y sistemas en dispositivos multiplataforma usando las librerías NumPy, Matplotlib, SciPy, SymPy y el framework Kivy.

1.3 Justificación

La finalidad de este trabajo integral de grado es afianzar la diversidad de conocimientos acerca de señales, sistemas y teoría de control para permitirle al estudiante o lector entender el funcionamiento de estas áreas, de manera ágil y eficiente por medio del desarrollo de una aplicación móvil realizada en el lenguaje de programación de alto nivel orientado a objetos Python, asistido por el framework de código abierto Kivy para alcanzar la complejidad de funcionalidades, y algunas librerías de cálculo científico que este ofrece como son: NumPy, utilizada para trabajar con matrices; SciPy, la cual ofrece distintas herramientas dedicadas específicamente a la resolución de problemas de computación científica; SymPy, usada para resolver ejercicios de ecuaciones diferenciales de cualquier orden y Matplotlib, empleada para la creación de gráficos estáticos, animados e interactivos.

La aplicación se llevará a cabo por medio de la herramienta de código abierto "Python-for-android" aplicada gracias a la virtualización de un sistema informático, el cual fue realizado por medio del programa "Virtual Box" bajo el sistema operativo "Ubuntu", para finalmente mediante la herramienta Buildozer generar el archivo .apk con proceso de automatización al compilar, seguidamente se podrá instalar la aplicación en dispositivos móviles y ejecutarla sin problema alguno.

2. Marco teórico

Para el desarrollo de la aplicación móvil “SimulaSS”, se mencionan los principales temas que se van a abordar, estos son: fundamentos y repaso sobre métodos numéricos, señales, sistemas y teoría de control. En estos temas se realiza una explicación convencional y se enfoca en su solución usando Python ya que este lenguaje hace referencia a la ciencia y análisis de datos gracias a sus librerías de cálculo científico tales como: NumPy, SciPy, Matplotlib y SymPy. Principalmente se habla de cómo implementar cada tema en Python. Si el lector desea profundizar en cada uno de estos temas, puede dirigirse al **Anexo (A): Anexo: Implementación Framework Kivy,** allí encontrara una explicación más detallada sobre cada uno, acompañado de una serie de ejercicios resueltos de manera convencional y otra aplicando el lenguaje de programación Python. La figura 2 representa el GUI del menú principal que contiene la aplicación móvil “SimulaSS”.

Figura 2 GUI menú principal. Fuente: autor

2.1 Métodos numéricos

Los métodos numéricos son técnicas que se utilizan para obtener de manera aproximada la solución a un procedimiento matemático por medio de la aplicación de cálculos y algoritmos de carácter lógico usando operaciones puramente aritméticas, las cuales idealizan y conciben métodos eficientes y menos complejos que aprueban los resultados expresados matemáticamente. La implementación de métodos numéricos en la aplicación móvil "SimulaSS", se basa en algebra lineal y ecuaciones diferenciales tal cual se aprecia en la figura 2-1 donde el usuario puede interactuar con su contenido y ver la solución de ejercicios dando uso de las librerías de calculo cientifico numpy y sympy.

Figura 2-1 GUI sub menú métodos numéricos. Fuente: autor

2.1.1 Algebra Lineal

Python ofrece algunos módulos para realizar operaciones matemáticas básicas, en el cual se dará uso de las matrices debido a que es una estructura de datos bidimensionales donde los números se organizan en filas y columnas y su aplicación se basa en operaciones lineales. El principal módulo que proporciona Python para realizar operaciones de álgebra lineal es NumPy y para implementarlo en código fuente de la aplicación móvil se usa el comando (**import numpy as np**). Seguido de eso, numpy ofrece la clase (**np)** para crear listas o matrices dando uso a operaciones básicas matriciales para entender la descomposición de señales, la figura 2-1-1 representa el GUI álgebra lineal el cual incluye ejercicios resueltos sobre matrices y sus operacioens basicas, en la siguiente versión el usuario podra ingresar parametros.

Figura 2–1-1 GUI algebra lineal. Fuente: autor

2.1.2 Ecuaciones Diferenciales Ordinarias (EDO)

Es un tipo de ecuación diferencial la cual involucra o relaciona de manera directa variables independiente x, funciónes en y(x) y varias derivadas de y(x), por lo tanto, una ecuación es ordinaria si hay únicamente una sola variable no dependiente, es decir, contiene sólo derivadas en proporción de una o más variables dependientes con respecto a una sola variable independiente. Sympy, es una librería de cálculo simbólico perteneciente a Python se importa usando el comando **from sympy import Eq, dsolve, init_printing, symbols, Function**. Dando uso de esta librería junto a sus módulos se resuelven ecuaciones diferenciales ordinarias para describir de forma matemática los sistemas físicos en el dominio del tiempo. la figura 2-1-2 representa el GUI ecuaciones diferenciales ordinarias, el usuario interactúa con ejercicios resueltos, en la siguiente versión el usuario podra ingresar parametros.

Figura 2-1-2 GUI Ecuaciones diferenciales ordiarias. Fuente:autor

2.1.3 Aplicaciones (EDO) con Python

Las aplicaciones diferenciales ordinarias se pueden aplicar en el área de la física, biología, química e ingeniería como tal. En este inciso se presentan algunas aplicaciones y gráficas interactivas que se pueden realizar usando ecuaciones diferenciales en el lenguaje de programación Python, explorando métodos simbólicos y numéricos con las librerías que ofrece para su desarrollo tales como lo son NumPy, SymPy, SciPy y Matplotlib. Estas apicaciones son incluidas en la aplicación móvil "SimulaSS", la figura 2-1-3 representa el GUI aplicaciones EDO, en el cual el usuario puede interactuar con el ejercicio del circuito RLC, ingresando parametros y obteniendo un resultado gráfico como objetivo dando uso de los modulos numpy y matplotlib y ejecutandose con exito en la apk, sin embargo el modulo scipy no es compatible con p4a "python for android" por lo tanto no realiza la simulación en dispositivos móviles.

Figura 2-1-3 GUI Aplicaciones EDO. Fuente:autor

Tabla 2.3.1.

Aplicación de ecuaciones diferenciales

Aplicación de Ecuaciones Diferenciales Ordinarias	
FUNCIÓN	**ECUACIÓN**
Sistema Masa Resorte	$F(t) - c\dot{x}(t) - kx(t) = m\ddot{x}(t)$
Circuito RLC en serie	$VR + VL + VC = E(t)$
Grafica interactiva de Batman	La ecuación de Batman fue creada por el profesor Matthew Register.

*Nota: En el **anexo(A),** se encuentra como se desarrolla cada aplicación de ecuaciones diferenciales de forma convencional y también en el lenguaje de programación Python. Estos ejercicios se implementan en la aplicación móvil, dando uso del framework Kivy.*

2.2 Señales

Una señal representa una variación cuantificable en el tiempo, esta esta denotada por la letra (t), la señal está cambiando constantemente, porque la información se ve perturbada en movimiento, no importa cuán lento o rápido, esto produce variación en la señal, la figura. 2-4-1 (a) es una representación genérica de la amplitud en un intervalo de tiempo realizada en el lenguaje de programación orientado a objetos Python para luego simular en la aplicación móvil dando uso del framework Kivy, en la sección 2-2-3, muestra un paso a paso de la implementación en la aplicación móvil “SimulaSS” y un mock up del UI (interfaz de usuario) para cada screen (pantalla) sobre las señales, su clasificación según su variable, las funciones impulso unitario y escalón unitario.

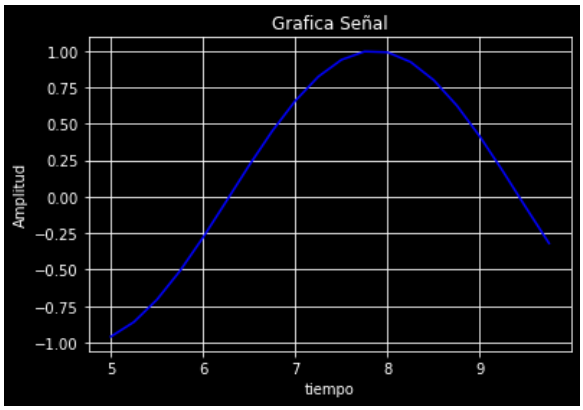


Figura. 2-4-1 (a) gráfica de una señal. Fuente: autor

Las señales se pueden clasificar según los parámetros que se le indiquen mediante transferencia de información. A continuación, en la tabla. 2-4-1 puede encontrar la clasificación de las señales según su variable y en la figura 2-4-1(b) encuentra el GUI del sub menu clasificacion de señales dando uso de los modulos numpy y matplotlib y ejecutandose con exito en la apk.

Tabla 2.4.1

Clasificación de las señales según sus variables

Clasificación de señales		
Clasificación por su variable independiente	Señal continua	Señal discreta
Clasificación por su reflejo en tiempo continuo (t) y tiempo discreto [n]	Señal Par	Señal impar
Clasificación por su repetitividad	Señal periódica	Señal no periódica

Figura. 2-4-1 (b) GUI sub menú. Fuente: autor

2.2.1 Función impulso unitario

La función de impulso unitario δ(t), cumple una función fundamental en el análisis de señales de sistemas, donde su duración en el tiempo es muy corta, tendiendo a cero y su amplitud se acerca a infinito, la cual en tiempo continuo se define como:

$$\delta(t) = \{\infty,\ t\ =\ 0\ 0,\ t\neq 0\ \} \quad \textbf{(1)}$$

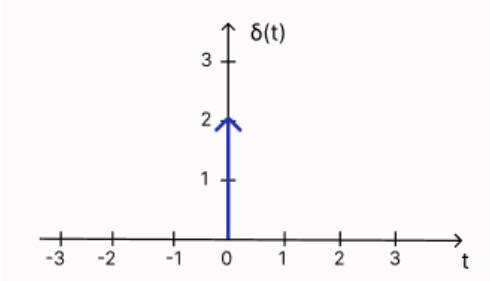


Figura. 2.2.1(a) función impulso unitario Fuente: Schaum

Al desplazar la función de impulso unitario en $\delta\,(t - t0)$, se puede definir de la siguiente manera:

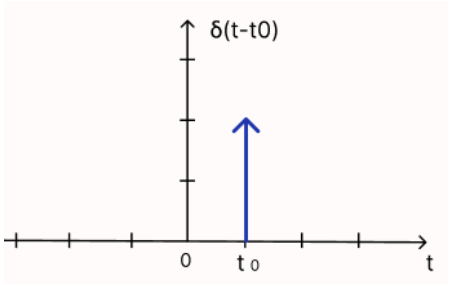


Figura.2.2.1 (b) Función impulso unitario desplazado Fuente: Schaum

La función impulso unitario $\delta(t)$ cuenta con algunas propiedades adicionales, esta son:

$$\delta(at) = \frac{1}{|a|}\delta(t) \quad \textbf{(2)}$$

$$\delta(-\,t) = \delta(t) \quad \textbf{(3)}$$

$$x(t)\delta(t) = x = (0)\delta(t) \quad \textbf{(4)}$$

$$x(t)\delta(t - t0) = x = (t0)\delta(t - t0) \quad \textbf{(5)}$$

2.2.2 Función escalón unitario

La función escalón unitario, es una función matemática la cual tiene como característica, tener un valor 0 para todos sus valores negativos y 1 para todos sus valores positivos, la cual en tiempo continuo se define como en la ecuación **(6).**

$$u(t) = \{1,\ t > 0\ 0,\ t < 0\} \quad \textbf{(6)}$$

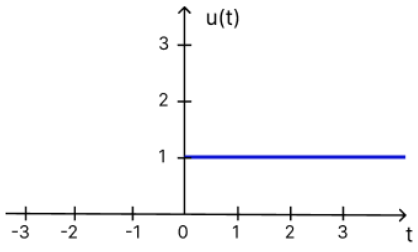


Figura. 2.2.2(a) Función escalón unitario Fuente: Schaum

Al desplazar la función de escalón unitario en $u(t - t0)$ **(7)**, se puede definir como en $t = 0$ y su valor es indefinido.

$$u(t - t0)\{1,\ t > t0\ 0,\ t < t0\ \} \qquad \textbf{(7)}$$

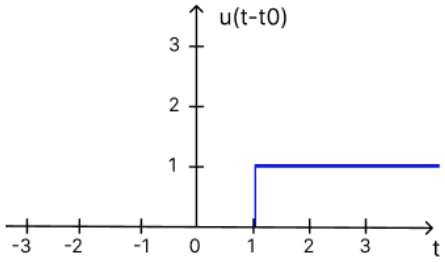


Figura. 2.2.2(b) desplazamiento $u(t - t0)$ Fuente: Schaum

2.2.3 *Señales*

Para la implementación de la sección. 2-2, se da uso de la metodología de desarrollo CDIO para dispositivos móviles, en la sección. 4-4, se encuentra el paso a paso para este desarrollo. A continuación, en la figura. 2-2-3, se encuentra GUI para cada screen (pantalla) Señales (a), Clasificación de señales (b), Función impulso unitario y escalón unitario (c).

2.3 Sistemas

Un Sistema básicamente es una interconexión entre componentes, transformando las señales que ingresan en dispositivos o módulos con parámetros de I/O, dando como resultado una o más señales de salida. Un ejemplo claro en el cual se presenta la transformación de señales son los transductores, estos pertenecen al área de los sistemas de automatización y

control donde su funcionamiento principal es transformar una magnitud física en una señal eléctrica y poder detectar magnitudes físicas como lo son: la temperatura, presión o caudal de fluidos etc. Registrando estas respuestas en dispositivos de almacenamiento de datos.

Un sistema continuo h(t), es aquel donde las señales continuas de entrada son transformadas en señales continuas de salida y un sistema discreto h[n], es aquel que transforma las entradas de tiempo discreto por salidas de tiempo discreto. Estos sistemas están representados en la figura. 2.4.2.

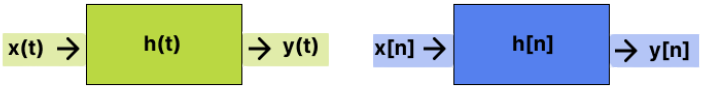


Figura. 2.4.2 Sistema continuo y sistema discreto Fuente: autor

El sistema continuo está representado simbólicamente $t,\ x(t) \rightarrow y(t)$ donde $x(t)$ es la entrada y $y(t)$ es la salida y el sistema discreto se representa de la misma manera: $n,\ x[n] \rightarrow y[n]$, siendo n el número de muestras ingresadas, adoptando valores enteros. Dicho de otra manera, que la señal sea continua o discreta depende de la varianza de la variable independiente. En la (sección 2-3-1) se encuentra con algunos términos básicos acerca de las propiedades de los sistemas (Oppenheim), los cuáles se están implementados en la aplicación móvil “SimulaSS”.

2.3.1 Propiedades básicas de los Sistemas

Sistemas con y sin memoria. Los sistemas sin memoria son aquellos en los que la salida en un momento dado depende solo de la entrada en ese momento. Es decir, y(t) no depende de t-t¡ o y[n] no depende de n-n¡. Sistemas latentes, acaparamiento, etc. Hay sistemas con memoria. La memoria del sistema tiene que ver con el almacenamiento de energía.

Invertibilidad y sistemas inversos. Un sistema es llamado invertible si diferentes entradas producen diferentes salidas, por tanto, es posible conectar otro sistema en serie a su salida de forma que la señal resultante del conjunto sea igual a la señal de entrada original.

Causal. Un sistema es causal si su salida en un determinado momento depende solamente de esa entrada en ese momento o momentos anteriores. También se conoce como sistemas no anticipados.

Estabilidad. Un sistema es estable si señales no divergentes ofrecen respuestas no divergentes, quiere decir, que una respuesta infinita en la salida no es generada por ningún valor finito de entrada. Intuitivamente, un sistema es inestable si una entrada pequeña puede causar una salida cada vez más grande.

Invariancia en el tiempo. Un sistema es invariante en el tiempo si un corrimiento en el tiempo de la señal de entrada genera un corrimiento igual en el tiempo de la señal de salida.

Linealidad. Un sistema es lineal si la suma ponderada de señales de entrada genera una salida que es la suma ponderada de las señales de salida que producirían cada una de las señales de entrada en solitario.

2.4 Sistemas de Control

Un sistema de control es un sistema de control automático, donde las variables de salida se comportan según las órdenes dadas por las variables de entrada tal cual se aprecia en la (figura 2-4) un diagrama de bloques simple de un sistema de control. Los sistemas de control se encuentran en gran cantidad, en diferentes sectores como lo es los sistemas robóticos, sistemas de vehículos espaciales, en los procesos industriales el cual requiera el control de temperatura, presión, humedad, flujo, etc. Antes de continuar con el análisis de los sistemas de control se deben tener claras algunas definiciones como lo es: planta, proceso, variables controladas, control retroalimentado, etc (Ogata).

Figura. 2-4 Diagrama de bloques sistema de control Fuente: autor

Básicamente, un sistema de control es una combinación de varios elementos conectados como una unidad para dirigirse o regularse a sí mismo o cualquier otro sistema con el fin de proporcionar una salida específica. Entonces, el sistema de control se usa para dirigir el funcionamiento de un sistema físico para llevar a cabo el objetivo deseado. En la (sección 2-4-1) encuentra una breve explicación a cada termino, los cuales están implementados en la aplicación móvil “SimulaSS”.

2.4.1 Conceptos y términos básicos de un Sistema de Control

Variable controlada. La variable controlada es un parámetro importante del proceso ya que esta es la cantidad o condición que se mide y controla, manteniéndola estable (sin variación), pues su variación modificaría las condiciones que requiere el sistema.

Señal de control o variable manipulada. La señal de control o variable manipulada como también es llamada, es la cantidad que el controlador modifica para afectar el valor de la salida del sistema.

Planta. La Planta es el elemento físico que requiere ser controlada.

Proceso. El proceso de control es un lazo diseñado para mantener la variable controlada en el set point. Algunos procesos son los procesos químicos, económicos y biológicos.

Sistema. Los sistemas no son solamente físicos tal cual se mencionó en la (Sección 2-3), este concepto de sistemas se aplica en fenómenos abstractos y dinámicos, los cuales se encuentran en la economía. Entonces, el concepto Sistema tiene una amplia definición el cual comprende sistemas físicos, biológicos y económicos.

Perturbaciones. Las perturbaciones son señales que pueden afectar negativamente el valor de salida del sistema. Si ocurren perturbaciones dentro del sistema, se denominan perturbaciones internas, mientras que las externas se generan fuera del sistema y constituyen la entrada.

Control retroalimentado. El control de retroalimentación se refiere a la operación de reducir la diferencia entre una salida del sistema y alguna entrada de referencia en presencia de perturbaciones. Se debe hacer hincapié en la naturaleza del ciclo de retroalimentación, que incluye los resultados del control para determinar las acciones de control.

Un sistema de control es un sistema de control dinámico en el que las variables de salida actúan en el orden especificado por las variables de entrada. En esta sección, describiremos diferentes tipos de sistemas operativos que se construyen solo porque tienen estos dispositivos, pero que deben seguir una lógica con al menos 3 elementos básicos:

Tabla 2.5 (a)

Elementos de un sistema de control

Elementos de un Sistema de Control	
Variable controlada	Una variable controlada es un elemento que se desea controlar, se puede concluir que es la salida del proceso.
Actuador	Los actuadores son instrumentos los cuales siguen las ordenes de un sistema de control, el cual realiza acciones que repercuten en tiempo real.
Punto de referencia o set-point	El set point es el valor de objetivo al que un controlador de temperatura intenta mantener las variables del proceso.

2.4.2 Tipos de sistemas de control

En sistemas de control está el **control de lazo abierto**, el cual es un sistema de **bucle o lazo abierto** es aquel que la salida del proceso registrado no es comparada con la señal de referencia y también el **control de bucle o lazo cerrado**, este es un sistema en lazo cerrado toma la salida del proceso y la compara con la señal de referencia para conocer en todo momento la evolución de la variable (REALIMENTADO, s.f.).

2.4.3 Controlabilidad y observabilidad

La controlabilidad y la observabilidad son conceptos importantes sobre la representación del espacio de estado de un sistema tal como se lee en un sistema de control. En este inciso se da una explicación grosso modo por separado acerca de la controlabilidad y la observabilidad. Para los sistemas de control en Python se da uso de la librería Control, la cual su función principal es realizar operaciones básicas para analizar y diseñar sistemas de control de retroalimentación, donde su instalación en Python es con el siguiente comando: pip install control. Esta Liberia ofrece funciones similares a Matlab. Sus características principales se muestran en la tabla 2.5.1.

Tabla 2.5.1

Características
Sistemas lineales input/output en espacio de estado y dominio de frecuencia
Algebra de diagrama de bloques: interconexiones en serie, en paralelo y de retroalimentación
Tiempo de respuesta: inicial, paso, impulso.
Respuesta de frecuencia: diagramas de Bode y Nyquist
Análisis de control: estabilidad, alcanzabilidad, observablidad, márgenes de estabilidad.
Diseño de control: colocación de valores propios, regulador cuadrático lineal
Diseño del estimador: estimador cuadrático lineal (filtro de Kalman)

Nota: En esta sección, el lector encuentra las características principales que se pueden realizar con la librería Control y sus funciones principales.

2.5 Problemas resueltos

En este inciso puede encontrar de manera ordenada una serie de ejercicios básicos acerca de las asignaturas Señales, Sistemas y Control. Los cuales harán parte de la aplicación móvil "SimulaSS".

2.5.1 Señales básicas

El siguiente ejercicio pertenece a la fuente: Señales y Sistemas Schaum 2da edición Hwei P. Hsu - Capítulo 1 Señales y Sistemas – Pag 24, ejercicio 1.22 (Hsu, 1995)

1. La (figura 2-5-1) muestra una señal de tiempo continuo $x(t)$. Realice la grafica para cada una de las siguientes señales:

a) $x(t)u(1-t)$
b) $x(t)[u(t)-u(t-1)]$
c) $x(t)\delta(t-\frac{3}{2})$

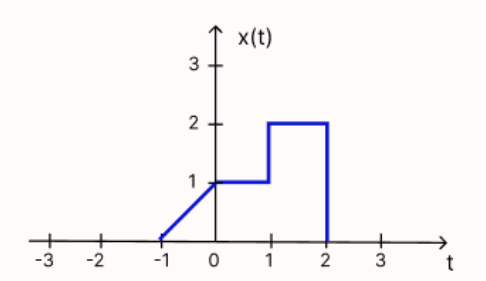


Figura. 2-5-1

a) Como se mencionó anteriormente en la ecuación **(3)**, la señal $x(t)u(1-t)$, al ser una función escalón unitario, se reescribe de la siguiente manera:

$$u(1-t)=\{1, t<1\ 0, t>1\}$$

La gráfica de la señal $x(t)u(1-t)$, es la representación de la **figura 2-5-1 (a)**:

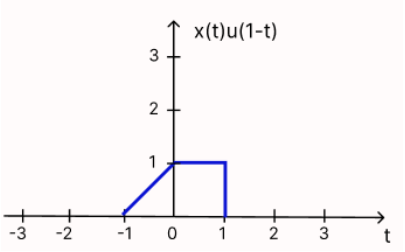


Figura. 2-5-1 (a)

b) La señal $x(t)[u(t) - u(t - 1)]$, es una función escalón unitario, se puede reescribir en base a las ecuaciones **(2)** y **(3),** Quedando de la siguiente manera:

$$u(t) - u(t - 1) = \{1, 0 < t \leq 1\ 0\}$$

La gráfica de $x(t)[u(t) - u(t - 1)]$, es la representación de la **figura 2-5-1 (b)**:

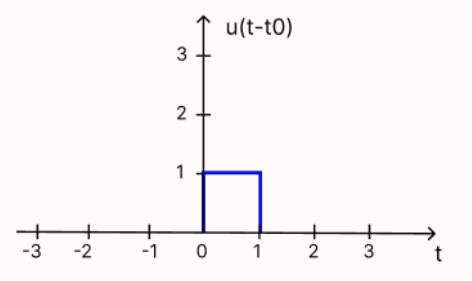


Figura. 2-5-1 (b)

c) Por la ecuación (5) se puede reescribir $x(t)\delta(t - \frac{3}{2})$ de la siguiente manera y se ilustra en la figura 2-5-1 (c):

$$x(t)\delta\left(t - \tfrac{3}{2}\right) = x(\tfrac{3}{2})\delta\left(t - \tfrac{3}{2}\right) = 2\delta\left(t - \tfrac{3}{2}\right)$$

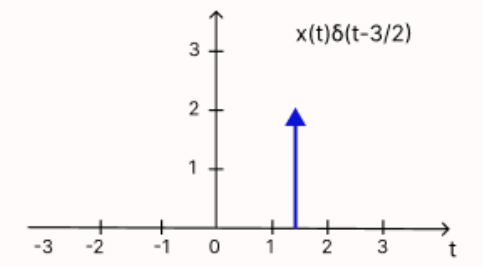


Figura. 2-5-1 (c)

3. Herramientas

3.1 Lenguaje y sistema operativo

Para el correcto funcionamiento de dispositivos móviles sobre este Trabajo Integral de Grado (TIG), el cual fue dirigido a estudiantes de ingeniería electrónica, donde su contenido se basa en señales, sistemas y control, fue ejecutado por completo en el lenguaje de programación orientado a objetos Python en su versión 3.8.8 64-bit. Python es un potente lenguaje de programación el cual es fácil de aprender y de open source, el cual tiene estructuras de datos eficaz de alto nivel, donde este permite trabajar más rápido e integrar sus sistemas de manera más efectiva (Python, s.f.).

De esta forma Python es sumamente utilizado en Back end development, Data science y App development, esto lo hace ser uno de los lenguajes de programación más populares según lo indica la revista **ITPro Today** (Tozzi, 2022) la cual atiende las necesidades de información de los profesionales de Information Technology (TI).

En todo el proceso de este TIG, se desarrolló con el sistema operativo de Windows. Pese a que no es un sistema operativo óptimo para el desarrollo de aplicaciones, como lo es Linux, el cual posee una potente línea de comandos, lo cual facilita el desarrollo de aplicaciones web tanto móviles en Python y por eso se toma como primera opción. Se trabaja con Microsoft Windows por practicidad, experiencia y dominio.

3.2 Programas utilizados

Para el desarrollo de la aplicación móvil “SimulaSS”, fue necesario implementar algunos softwares, librerías y submódulos, para diseño y programación, es importante resaltar su función principal dentro de este trabajo integral de grado, tal como se ilustra en la tabla 3-2.

Tabla 3-2

Softwares utilizados para el desarrollo de la aplicación móvil

Programas Utilizados
Visual Studio Code
Spyder
Figma
Oracle VM VirtualBox

Nota: En este inciso se mencionan los softwares, las cuales son necesarias para el desarrollo de la aplicación móvil, en la sección 4.1, se explica el funcionamiento de cada uno dentro para el desarrollo del apk.

3.3 Librerías

Python cuenta con una variedad de librerías y submódulos, estos ayudan a un desarrollo óptimo. En este caso, para la simulación en la aplicación móvil, se tendrán en cuenta las librerías y módulos, estos ofrecen el manejo de vectores, matrices y gráficos en dos dimensiones, a continuación, en la tabla 3-3 encuentra una breve explicación:

Tabla 3-3

Librerías y framework utilizadas para el desarrollo de la aplicación móvil

Librerías
Kivy - KivyMD
NumPy
SymPy
Scipy
Matplotlib

Nota: En este inciso se mencionan las librerías, las cuales son necesarias para el desarrollo de la aplicación móvil, en la sección 4.2, se explica el funcionamiento de cada librería en el desarrollo del apk.

3.4 Entorno virtual

En esta sección, se habla del uso del entorno virtual con Python. Básicamente se implementa en la aplicación móvil **"SimulaSS",** por buenas prácticas de programación debido a que se aíslan las librerías junto a las versiones que requiere de otros proyectos. Se utiliza el módulo **venv** como un Script en la ruta de la carpeta fuente. Su instalación es con el siguiente comando: pip install virtualenv.

4. Iniciativa para implementación de ejercicios prácticos de señales, sistemas y control en dispositivos móviles

Para desarrollar prácticas de asignaturas de señales, sistemas y control, apoyadas en dispositivos móviles. Este capítulo describe la iniciativa CDIO y cómo implementar las herramientas mencionadas en el capítulo anterior.

4.1 Diseño e implementación de algoritmos para la operación de métodos numéricos necesarios en simulaciones de ejercicios básicos de señales, sistemas y control en dispositivos móviles.

En esta sección se describe cómo alcanzar el objetivo “Desarrollar algoritmos para la operación de métodos numéricos necesarios para simular ejercicios básicos de señales y sistemas”. En el cual se da una breve explicación de cómo implementar las herramientas mencionadas en el inciso 3.1, ilustrando el GUI de cada software, para realizar la simulación de los distintos ejercicios y como fueron desarrollado.

4.1.1 *Visual Studio Code (VSC)*

Se utiliza este software de edición de texto estructurado a Python junto a su útil librería para el desarrollo de aplicaciones multiplataforma (móvil, ordenador o Tablet) para poder acceder de manera práctica a un archivo editable de extensión [.py], de igual modo y bajo la misma productividad el framework Kivy genera un archivo de extensión [.kv]. Se opta por usar este editor debido a su clara legibilidad y versatilidad en la compilación de proyectos, en este,

se importarán todas las dependencias que se usarán para el desarrollo de la aplicación móvil conjuntamente a cada módulo que ofrece la librería mencionada, tal y como se aprecia en la figura. 4-1-1, donde el lector puede las clases que se crean.

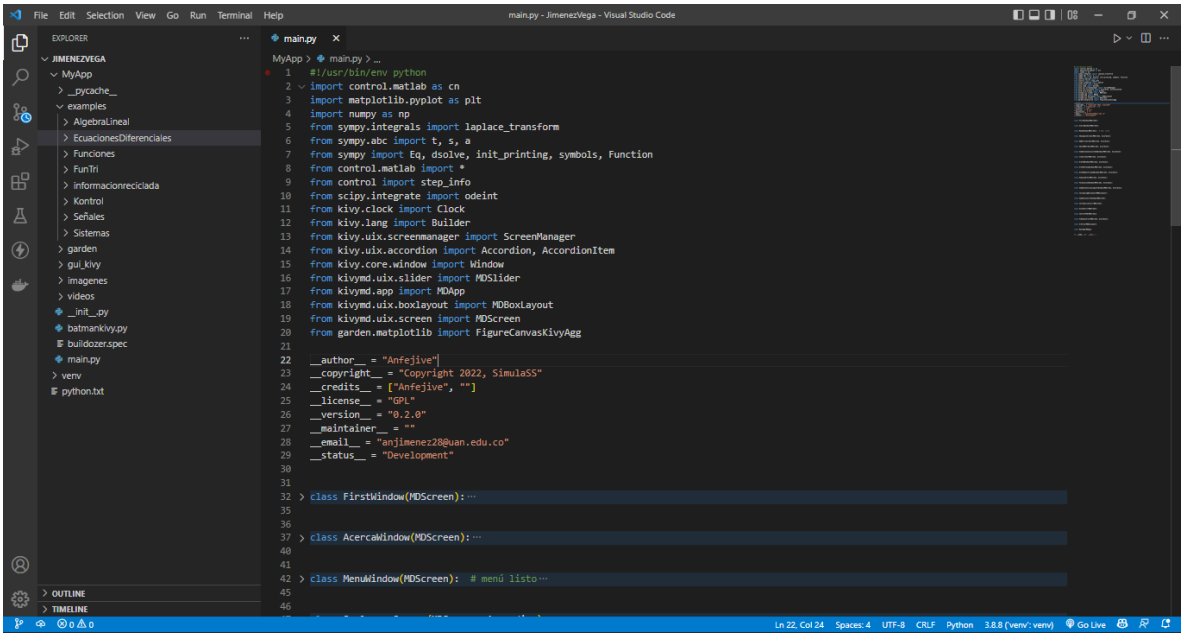


Figura 4-1-1 Sintaxis Visual Studio Code. Fuente: autor

4.1.2 Spyder

Spyder, es un potente IDE entorno de desarrollo interactivo para Python, el cual su función principal en este trabajo integral de grado, es probar cada ejercicio de una manera más interactivo, gracias a que posee funciones avanzadas de edición, pruebas interactivas, depuración, introspección y un entorno informático numérico. En la figura 4-1-3**,** se encuentra el entorno de desarrollo Spyder, este es un Script de Python en el cual se desarrolló la simulación de cada ejercicio seleccionado.

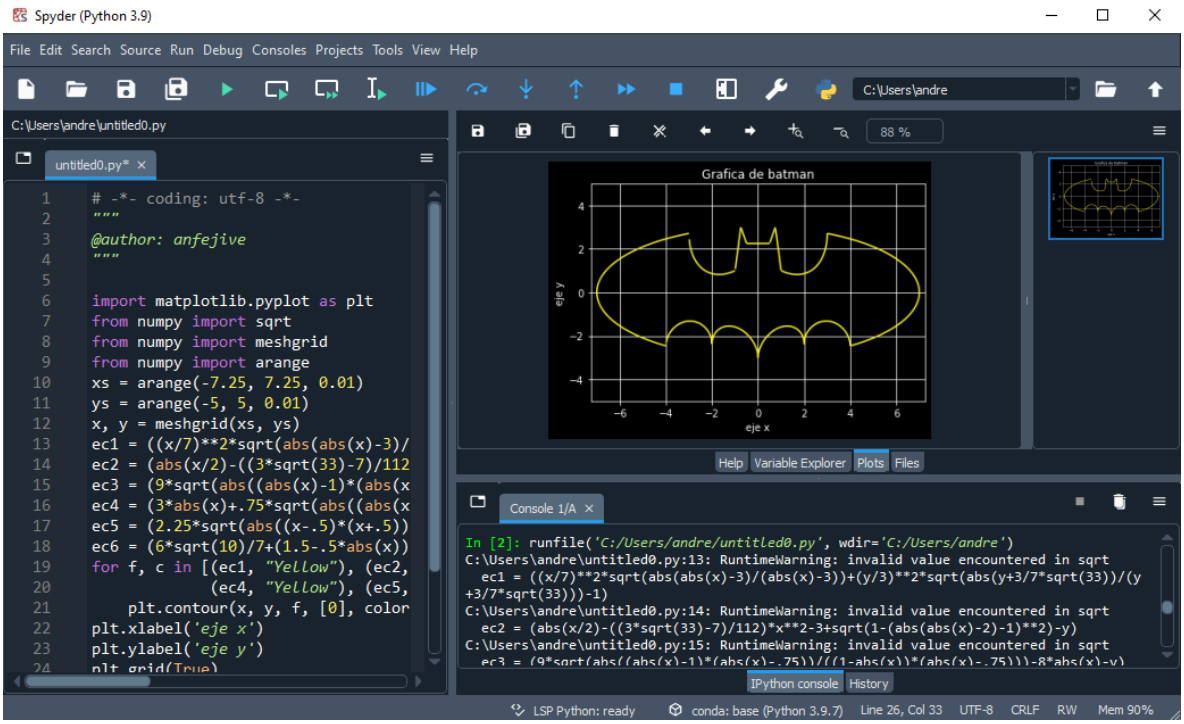


Figura 4-1-3 Sintaxis Spyder. Fuente: autor

4.1.3 Figma

Figma es una herramienta de diseño de vectores en linea especialmente diseñada para programas API web y móviles, incluso tiene colaboración en tiempo real, historial de versiones, sistema de comentarios, uso de fuente de iconos, diseño de interfaces flexibles. El uso que se le dio para el desarrollo de este trabajo integral de grado, fue para diseñar imágenes propias y realizar el Mock up de la aplicación móvil. En la figura 4-1-4 el lector puede observar la GUI de esta herramienta.

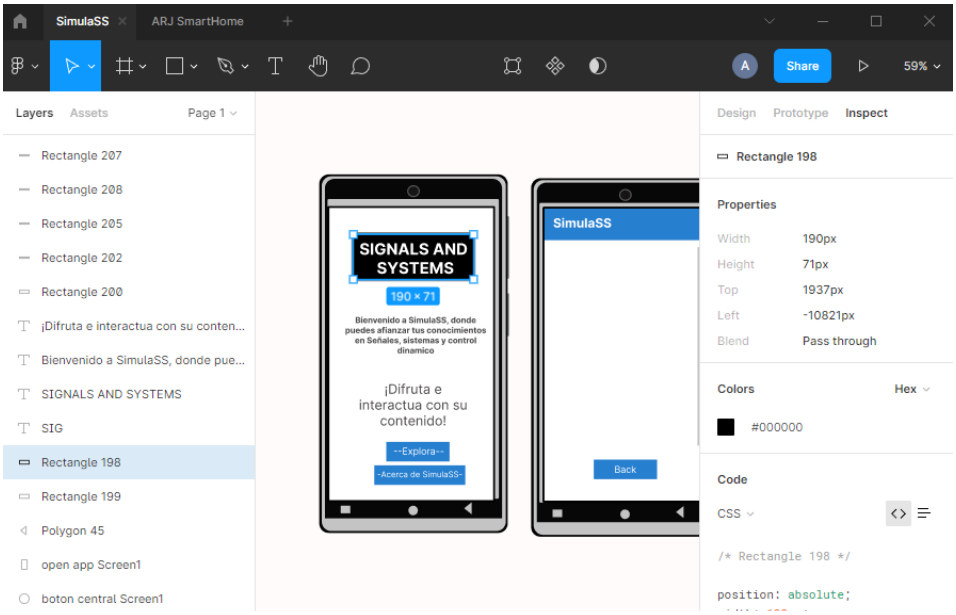


Figura 4-1-4 Interfaz gráfica de Figma. Fuente: autor

4.1.4 Oracle VM VirtualBox

El software Oracle VM VirtualBox facilita trabajar con discos virtuales permitiendo utilizar cualquier sistema operativo como si fuera una simple ventana.

Estos discos virtuales deben crearse o descargarse, ya continuación configurarlo para su óptimo funcionamiento, y suelen tener un tamaño elevado, por eso es recomendado hacer una buena distribución en el almacenamiento que se va a trabajar. Se ha utilizado para trabajar con un disco virtual de Ubuntu (distribución de Linux) preparado para la conversión de Python en Android.

Este software es necesario para el desarrollo de la aplicación móvil ya que una vez teniendo la estructura del código realizado con el framework Kivy de Python. Se construye en aplicativo móvil de Android en formato apk. El software Oracle VM VirtualBox a diferencia de otros

discos virtuales es gratuito, no requiere tener licencia para tener acceso en su aplicación.En la figura 4-1-5 el lector puede observar la GUI de esta herramienta.

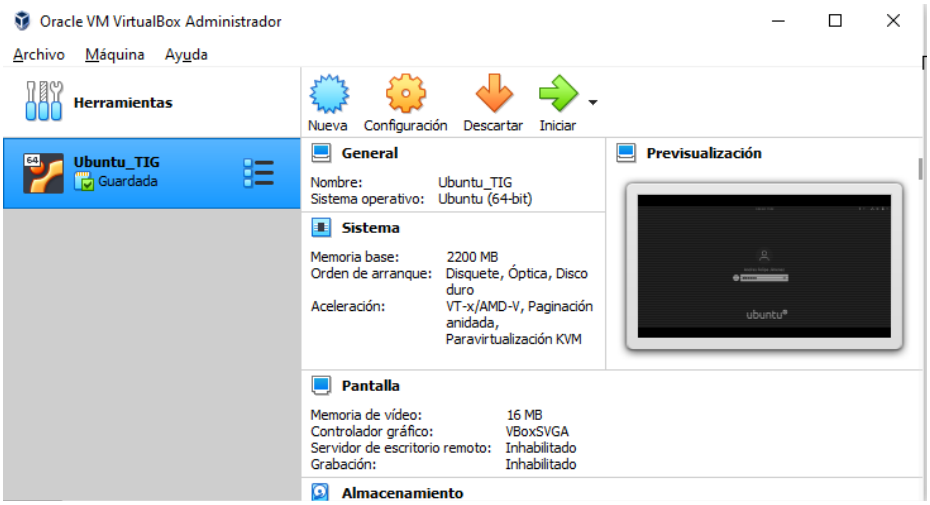


Figura 4-1-5 Ubuntu VX Oracle Fuente: autor

4.2 Cálculos y operaciones en Python para simulación de: señales, sistemas y control, también para el desarrollo de métodos numéricos en Python, donde su solución es implementada en el framework Kivy - KivyMD y dando uso de las librerías NumPy, SciPy, SymPy y Matplotlib.

4.2.1 Kivy

Kivy es un framework o dependencia que pertenece a Python, su función principal es crear interfaces gráficas de usuario. Kivy se ejecuta en sistemas operativos Linux, Windows, OS X, Android, iOS y Raspberry Pi. Puede ejecutar el mismo código en todas las plataformas compatibles (Kivy). Su instalación, se realiza desde la consola de Windows o directamente desde el terminal del IDE preferido, se debe escribir el siguiente comando: python -m pip install kivy. El módulo kivy.uix, ofrece widgets los cuales son elementos de una interfaz gráfica de usuario (GUI),

este módulo contiene clases para crear y administrar widgets y pantallas, como lo es el widget ScreenManager y Accordion.

El ScreenManager es un widget utilizado para administrar múltiples pantallas, este widget solo permite mostrar una pantalla a la vez, ofrece las clases Screen para habilitar cada pantalla usada en la aplicación móvil. Para implementarlo en el archivo de extensión [.py], se escribe la siguiente línea de código (Kivy): from kivy.uix.screenmanager import ScreenManager, Screen. En la figura. 4-2-1 (a), puede observar cómo se implementa el widget mencionado.

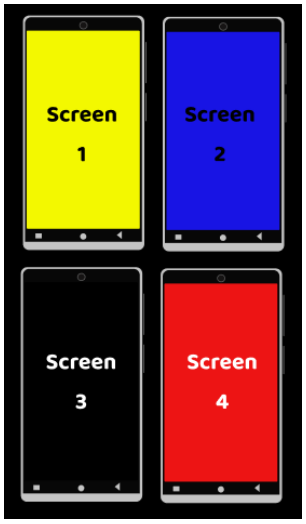


Figura 4-2-1 (a) ScreenManager Fuente: autor

El widget accordion, es un tipo de menú donde las opciones se apilan ya sean vertical u horizontalmente y al oprimir el elemento este se abre de inmediatamente para enseñar su contenido. Para implementarlo en el archivo de extensión [.py], se escribe la siguiente línea de

código (Kivy). from kivy.uix.accordion import Accordion, AccordionItem. En la figura. 4-2-1 (b), se puede observar cómo se implementa el widget mencionado.

Figura 4-2-1 (b) *Accordion* Fuente: autor

4.2.2 KivyMD

Es una biblioteca que contiene una colección de widgets que son compatibles para el uso efectivo de Material Desing, es decir, ofrece un set de herramientas como lo son widgets y layouts, entre otras, que permiten desarrollar interfaces gráficas de calidad siguiendo altos estándares establecidos sin afectar su rendimiento o la simplicidad de uso. Esta librería ofrece varias clases, entre esas se encuentran MDApp, MDScreen, MDSlider, MDBoxLayout.

4.2.3 NumPy

Numpy es una biblioteca de Python utilizada para trabajar con matrices. También tiene funciones para trabajar en el dominio del álgebra lineal, la transformada de Fourier y las matrices. Para su instalación desde la consola de Windows o directamente desde el terminal del

IDE preferido, se debe escribir el siguiente comando: pip install numpy. Como se mencionó anteriormente, NumPy es una librería que trabaja con matrices, donde este ofrece unos módulos importantes como lo es el módulo np.array.

El módulo np.array, pertenece a la librería NumPy y este permite crear un arreglo multidimensional, este módulo también ofrece arreglos de tipo booleano, entero, flotante y complejo. Se puede instalar tan solo escribiendo el comando: import numpy as np.

4.2.4 SymPy

Sympy es una librería de Python para realizar cálculos simbólicos. SymPy tiene una amplia gama de funciones para aritmética simbólica básica, calculo, álgebra, las matemáticas discretas, la física cuántica. Su uso principal en la aplicación móvil fue para la solución de ecuaciones diferenciales ordinarias. Para su instalación desde la consola de Windows o directamente desde el terminal del IDE preferido, se debe escribir el siguiente comando: pip install sympy.

SymPy es una librería de cálculo simbólico y ayuda a solucionar EDO como se mencionó anteriormente, esta cuenta con el módulo Eq, dsolve, symbols y functions, los cuales hacen más fácil el desarrollo de las EDO.

4.2.5 SciPy

Es una biblioteca de código abierto procedente del lenguaje de alto nivel Python la cual ofrece diversos archivos de herramientas dedicadas a problemas de computación científica implementando varios grupos de algoritmos matemáticos optimizados para su uso eficaz, además, proporciona estructuras de datos para operaciones ampliamente eficientes y aplicables

ante cualquier valor de un atributo. Su operación es eficaz ante los arreglos de la librería NumPy debido a que pone en práctica muchas de sus funcionalidades; para la generación de sus respectivas gráficas es necesario enlazar la librería Matplotlib.

4.2.6 Matplotlib

Matplotlib es una librería completa para crear gráficos estáticos, animados e interactivos en Python. Matplotlib está escrito principalmente en python, algunos segmentos están escritos en C, Objective-C y JavaScript para la compatibilidad de la plataforma (w3schools). Su principal uso en la aplicación móvil fue la graficación de todas las señales comprendidas.

La librería matplotlib ofrece la función plot() para graficar la función a la cual se desee llegar. Para importar la librería, se llama plt de la siguiente manera: import matplotlib.pyplot as plt.

4.3 De Python a kivy, de clases a widgets

Para visualizar los ejercicios realizados en Python acerca de algebra lineal, señales sistemas y control al lenguaje kivy, la comunidad proporciona la librería de Python garden.matplotlib, esta normalmente se aplica en análisis y visualización de datos, asimismo proporciona la clase FigureCanvasKivyAgg y esta se usa para crear un gráfico en matplotlib.

4.3.1 Portabilidad a dispositivos móviles

Se desarrolla la aplicación móvil utilizando el Framework Kivy, para luego construirlo en aplicativo móvil de Android en su formato. apk, utilizando la herramienta de construcción de código abierto "Python-for-android" construido bajo la distribución "Ubuntu" de Linux. Por tal

razón, se virtualizó el sistema operativo "Ubuntu" utilizando el programa de "Virtual Box" y se construyó la aplicación móvil para Android, utilizando la mencionada herramienta de construcción.

4.3.2 Ventaja de sus librerías contra otras

- La ventaja de las librerías implementadas en la aplicación móvil **"SimulaSS",** se basa principalmente en la conversión que se realizó de Python a Kivy.
- Su funcionamiento es completamente sin conexión a internet.
- Funciona en el sistema operativo Android en todas sus versiones
- Es multiplataforma ya que se puede ejecutar como aplicación de Windows.

4.4 Diseño de una ruta didáctica para implementar la simulación de ejercicios prácticos de señales, sistemas y control.

Para el desarrollo de la ruta didáctica de **"SimulaSS"**, se escogió la iniciativa CDIO (concebir, diseñar, implementar y operar), aplicándola en la infografía basada en dispositivos móviles como se aprecia en la figura. 4.4**.**

Este identifica una iniciativa educativa orientada al rediseño de los currículos en ingeniería, brindando a los estudiantes una guía especializada y rediseñada bajo esta iniciativa en la cual es las siguientes sub secciones el lector podrá identificar de una forma mas clara. Para el desarrollo de la aplicación móvil **SimulaSS**, se implementa esta metodología para obtener una alternativa diferente de estudio de los estudiantes de ingeniería electrónica de la universidad Antonio Nariño sede Ibagué en las asignaturas de señales, sistemas y control, acudiendo al mobile-learning accediendo al contenido de aprendizaje utilizando dispositivos

móviles. Por consiguiente, se da un paso hacia la implementación de contenido, en dispositivos móviles.

4.4.1 Concebir

Identificación y carectización del tema a investigar, para implementar en la aplicación móvil SimulaSS se basa en Señales, Sistemas y Control, por tal motivo el usuario encuentra en la aplicación móvil un menú principal, el cual su contenido se basa en métodos numéricos, señales y sistemas, teoría de control y finalmente la selección de algunos Ejercicios Interactivos, tal cual se aprecia en la figura. 4.4.1**.** Se identifica la metodología de desarrollo de software Scrum, por su versatilidad e interacción con el cliente y el desarrollador en cada momento donde las herramientas para el desarrollo de software constan de: visual studio code, figma y spyder las cuales se explicaran en la sección del diseño según la iniciativa CDIO.

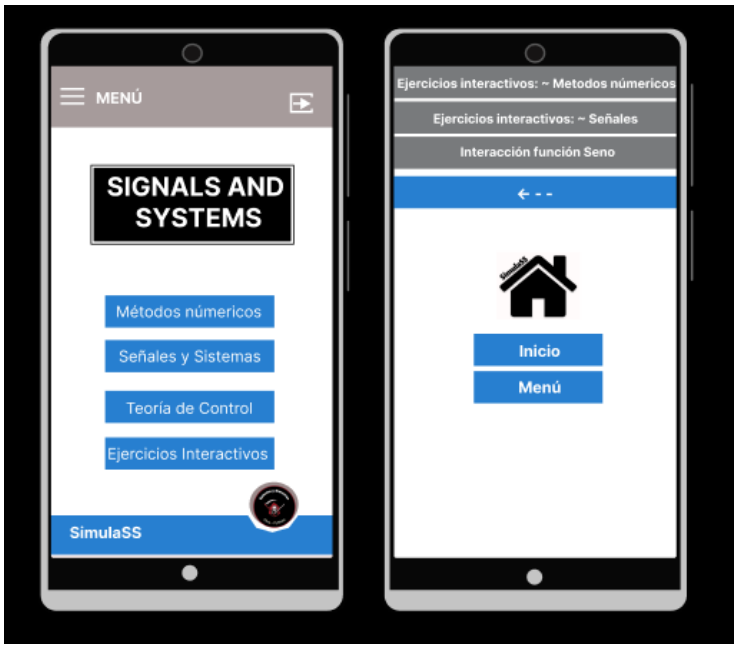


Figura. 4.4.1 *menú principal e implementación de ejercicios* Fuente: autor

4.4.2 Diseñar

El diseño de la arquitectura de la aplicación móvil consta del diagrama de clases UML y el diseño realista (Mock Up) realizados totalmente en Figma la herramienta de vectores online, estos diseños se implementan para entender la interacción de SimulaSS. En esta sección se procede a hablar sobre las herramientas de software implementadas como lo es el editor de texto Visual Studio Code para estructurar el lenguaje de programación Python generando un archivo de extensión .py y un archivo de extensión. kv, que facilita dar apariencia a las clases creadas para los widgets y finalmente el entorno de desarrollo científico Spyder el cual facilita bastante la compilación de los ejercicios seleccionados gracias a que posee capacidades de edición avanzadas, pruebas interactivas, depuración y autoprueba y un entorno de computación númerica.

4.4.3 Implementar

Al momento de realizar la implementación del contenido y los ejercicios en la aplicación móvil, se realizan pruebas de validación y depuración para obtener un código más legible y entendible organizando cada gui haciendo referencia a los temas principales. Se hace portable la aplicación móvil utilizando el Framework Kivy, para luego construirlo en aplicativo móvil de Android en su formato .apk, dando uso de la herramienta de construcción de código abierto "Python-for-android" construido bajo la distribución "Ubuntu" de Linux. Por tal razón, se virtualizo el sistema operativo "Ubuntu" utilizando el programa de "Virtual Box" y se construyó la aplicación móvil para Android, utilizando la mencionada herramienta de construcción.

4.4.4 Operar

La aplicación móvil SimulaSS opera en laboratorios de electrónica, la cual será usada por personal como profesores, tutores, estudiantes e inclusive aficionados, en donde podrán afianzar sus conocimientos en el área de señales, sistemas y control y realizar simulaciones gráficas para realizar sus respectivos análisis.

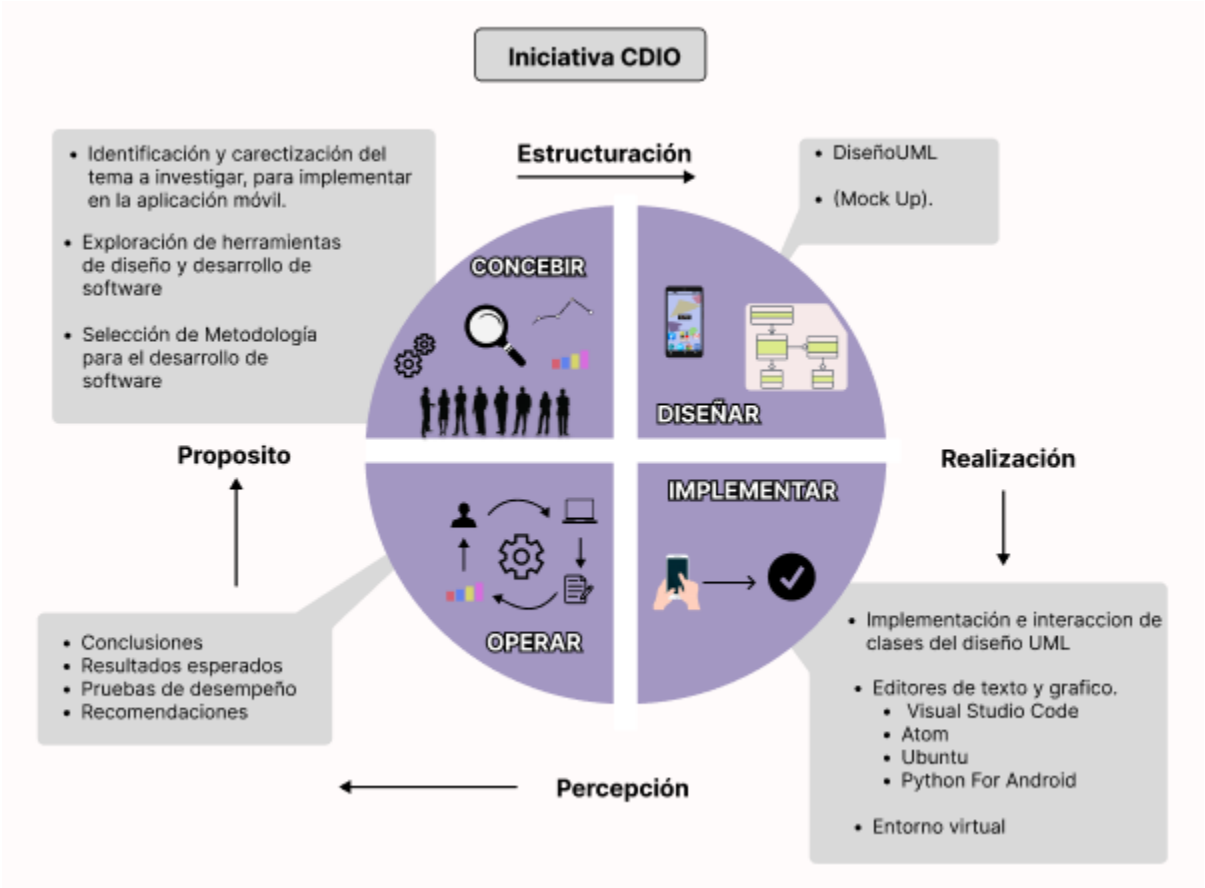


Figura. 4.4 Iniciativa CDIO para dispositivos móviles Fuente: autor

4.5 Utilización de la aplicación para el diseño de prácticas de laboratorio

La aplicación móvil **"SimulaSS"**, se utiliza en prácticas de laboratorios para profesores, tutores, inclusive estudiantes y aficionados los cuales deseen verificar las simulaciones que están desarrollando en el aula de clases.

4.6 Descripción metodológica

Este trabajo integral de grado, se basa en la metodología de desarrollo de software Scrum, esté es un framework de desarrollo ágil. El cual puede ser visto como el arranque inicial para decidir el proceso de desarrollo de un nuevo proyecto, se caracteriza por ser un modelo el cual define un conjunto de prácticas y roles como lo son el Scrum master, el Product Owner y el Equipo Scrum. Este proyecto dispone de ciclos de trabajo llamados Sprints los cuales permiten asignar un tiempo de trabajo donde su duración habitual consta de 4 semanas. Para este TIG fue necesario implementar 4 Sprints de un mes de duración cada uno, en cada Sprint se va consiguiendo el objetivo principal el cual es desarrollar una aplicación móvil para afianzar conocimientos en señales, sistemas y control dinámico usando el lenguaje de programación Python junto al framework Kivy.

ScrumMaster: Es la persona encargada de dirigir el proyecto. En este caso es el director de la tesis.

ProductOwner: Nicho al cual va dirigida la aplicación. Este trabajo integral de grado, va dirigido a estudiantes de diversas áreas de la ingeniería la cual implique ver las asignaturas: señales, sistemas y control.

TeamScrum: Son los encargados de recibir los parámetros que dirige el ScrumMaster. En este caso, es el desarrollador de la aplicación móvil.

4.7 Mock Up

El presente Mock up fue diseñado totalmente en el editor de gráficos vectoriales Figma, tomando como ejemplo de dispositivo móvil un Xiaomi Redmi Note 9S el cual su sistema operativo es Android. Si el lector desea ver cada Screen, puede dirigirse al Anexo **(B): Anexo Mockup App Móvil con Figma.** En la figura. 4.8 (a), se encuentra el logo principal de **"SimulaSS"**, el cual se ve reflejado al instalarse como apk en la figura, 4.8 (b**).**

Figura. 4.8 (a) *Logo de la aplicación móvil* Fuente: autor

Figura. 4.8 (b) *Presentación de la Apk en dispositivos móviles* Fuente: autor

4.8 Diagrama de clases UML

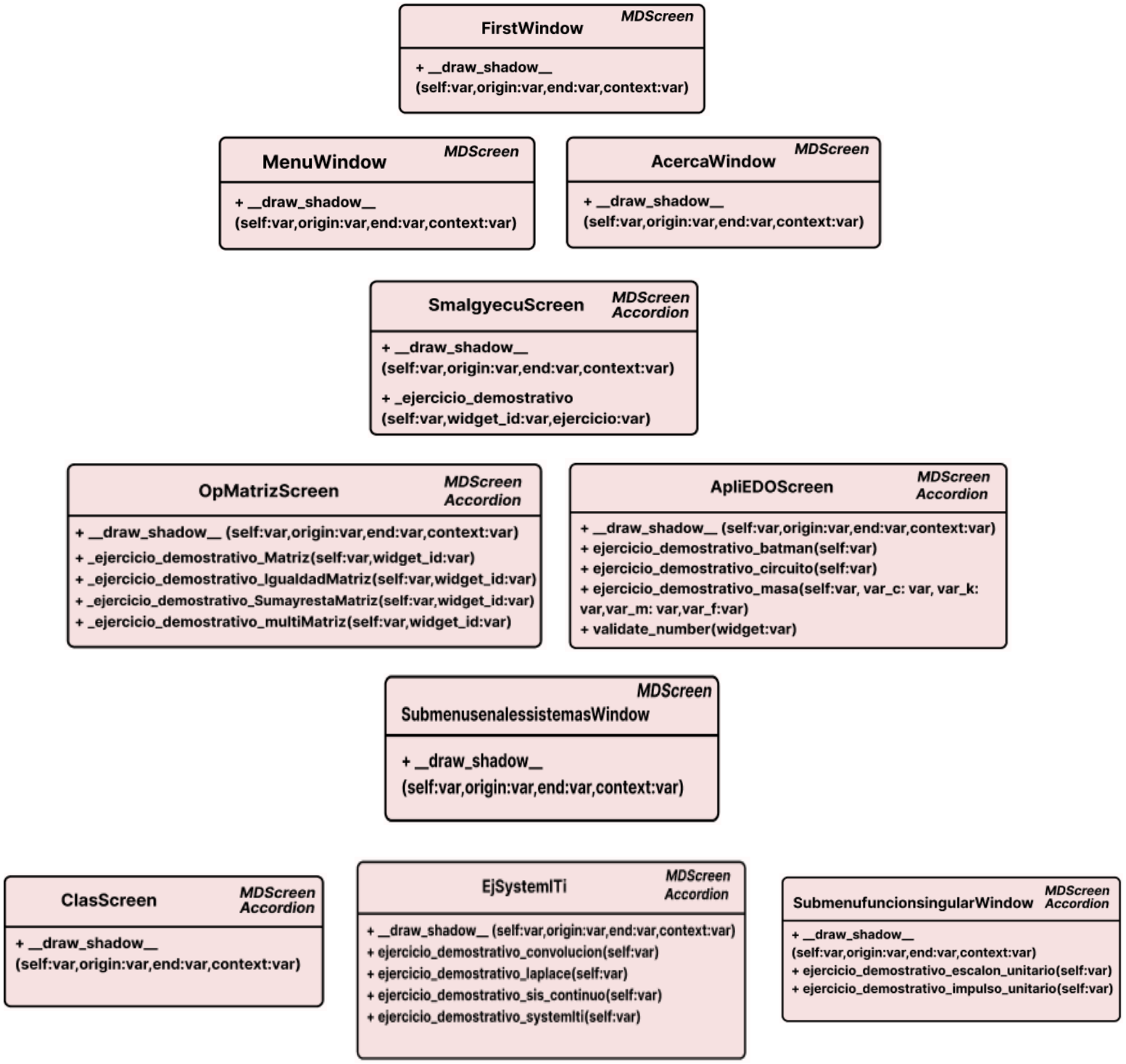


Figura. 4-9 Diagrama de clases UML Fuente: autor

En la figura. 4-9, Se ecuentra el diagrama de clases UML, en este diagrama encuentra las clases, funciones y métodos especiales, los cuales fueron adoptadas para su desarrollo. La interacción entre clases se basa principlamnte en los botones, ya que con estos el usurio puede interactuar con diferentes simulaciones que la apk presenta, llevandolo de un secreen a otro.

4.8.1 Métodos especiales:

Los métodos especiales en el lenguaje de programación orientado a objetos Python, se identifican por tener doble guíon bajo en cada lado de su nombre como lo son: __draw_shadow__ y el metodo __init__, estos son los unicos metodos implementados en las clases para el desarrollo de la aplicación móvil, algunos de los botones que son mencionados más adelante, también son clases.

__draw_shadow__ : genera una sombra de imagen.

__init__: Este metodo inicializa los atributos del objeto el cual fue creado.

5. Ejercicios Practicos

En este capítulo se dará una explicación paso a paso del funcionamiento de un sistema masa-resorte-amortiguador, realizado de forma manual (convencional) y usando el lenguaje de programación Python perteneciente al blog que Posee Hans-Peter Halvorsen acerca de Python (Halvorsen H.-P.). Se toma el procedimiento realizado por el autor principal, el objetivo es pasar el ejercicio al dispositivo móvil usando el framework Kivy e ingresar parámetros para variar su salida en representación de una gráfica la cual muestra la posición x1 y la velocidad x2 con respecto al tiempo (t).

5.1 Circuito RLC en serie

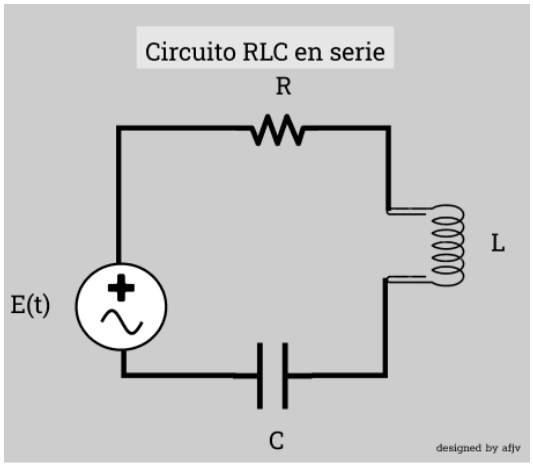


Figura. 5-1 (a) circuito RLC en serie. Fuente: autor

Circuito RLC en serie consiste de una fuente de entrada, una resistencia R, un inductor L y un condensador C, se desea encontrar la ecuación diferencial la cual permita calcular la carga en el circuito como la corriente que fluye por el circuito. Para dar solución a este ejercicio se deben usar algunas leyes las cuales surgen de la física, como lo es *la Ley de tensiones de Kirchhoff* y básicamente esta ley nos indica que si se suman las caídas de voltajes de los

elementos (RLC), entonces esa suma debe ser igual al voltaje el cual se le está suministrando al circuito, se puede expresar de la siguiente manera:

$$VR + VL + VC = E(t)$$

La ley de Kirchhoff es la que vamos a emplear para resolver el ejercicio genérico el cual se puede apreciar en la figura. 5-1 (a) y para eso, damos inicio calculando cada uno de los voltajes que se tiene por cada elemento. Para solucionar cada voltaje se emplea la *Ley de Ohm,* la cual es muy importante ya que esta relaciona el voltaje, la corriente y la resistencia, la cual nos indica lo siguiente:

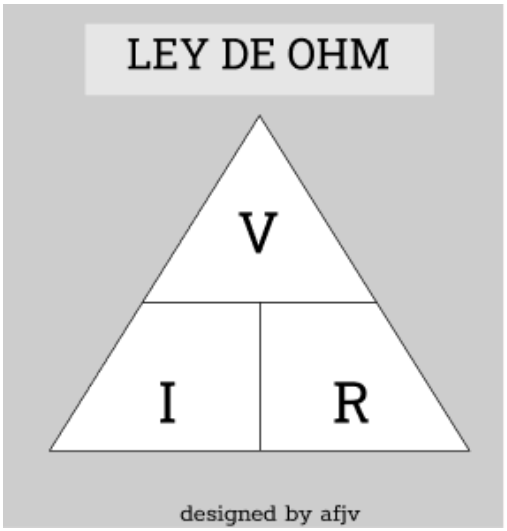


Figura. 5-1 (b) ley de Ohm. Fuente: autor

$$VR(t) = R.i(t)$$

La *ley de Ohm* como lo indica la ecuación y la figura 5-1 (b), dice que el voltaje en la resistencia *VR(t)* es igual al producto de la resistencia (*R)* por la corriente (I) que fluye por cada elemento. También se puede expresar la Ley de Ohm de una forma equivalente, para asimismo expresarlas en forma de ecuación diferencial, utilizando la carga *(q)* en lugar de la corriente *(i),* para eso debemos recordar cual es la derivada de la corriente *(i)* y esta se expresa de la siguiente manera:

$$i(t) = \frac{dq}{dt} \longrightarrow q'(t) \textbf{ (8)}$$

Debido a lo anterior, ahora se puede reescribir nuevamente la *Ley de Ohm*, y esta nos daría de la siguiente manera:

$$VR(t) = R.\frac{dq}{dt} \longrightarrow VR(t) = R.q'(t) \textbf{ (9)}$$

Como ya se obtuvo la caída de voltaje de la resistencia *VR,* se procede a calcular la caída de voltaje en el Inductor *VL,* donde se multiplica la inductancia *L* por la derivada de la corriente respecto al tiempo, el cual se expresa de la siguiente manera:

$$VL = L.\frac{di}{dt} \longrightarrow VL = L.q'(t) \textbf{ (10)}$$

Se procede a escribir la ecuación, en términos de la carga (q), entonces se deriva la ecuación de la derivada de la corriente, en otras palabras, la segunda derivada, a partir de calcular la carga ya es más fácil hallar el valor de la corriente, por ende, este se expresa de la siguiente manera:

$$VL = Lq''(t) \quad \textbf{(11)}$$

Por último, para el condensador (C), se tiene que el voltaje en el condensador *Vc,* según *la Ley de Ohm,* se puede expresar de la siguiente manera:

$$Vc(t) = \frac{q(t)}{C} \longrightarrow Vc(t) = \frac{1}{t}q(t) \quad \textbf{(12)}$$

Como ya se tiene el voltaje de cada elemento en términos de carga, la ecuación *(2)* se puede sustituir para dar expresión del circuito modelado de la **fig. #** en forma de ecuación diferencial, quedando de la siguiente manera:

$$Rq'(t) + Lq''(t) + \frac{1}{C}q(t) = E(t) \textbf{ (13)}$$

Cumpliendo el objetivo principal del inciso el cual es *ecuaciones diferenciales ordinarias,* se procede a ordenar la ecuación diferencial de acuerdo al orden de la derivada, obteniendo una ecuación diferencial de segundo orden, cuyos coeficientes son constantes, en general no es homogénea, expresada de la siguiente manera:

$$Lq''(t) + Rq'(t) + \frac{1}{C}q(t) = E(t) \quad \textbf{(14)}$$

Se puede retirar la *(t)* de las ecuaciones si desea ya que implícitamente se sabe que la carga depende del tiempo, entonces la ecuación diferencial ordinaria de segundo orden de un circuito *RLC* en serie, quedaría así:

$$Lq'' + Rq' + \frac{1}{C}q = E(t) \textbf{ (14)}$$

El siguiente ejercicio, realizado en el lenguaje de programación Python, pertenece a la fuente (Curiosidata, s.f.). En el cual se aprecia la salida de la corriente y el voltaje del circuito

RLC, en este algoritmo se emplean las librerías numpy y matplotlib y se da uso del framework Kivy. Al modificar el ejercicio circuito RLC, se le agrega la función input (), para agregar números enteros y flotantes para y modificar su salida donde el usuario pueda experimentar con las gráficas y pueda llegar a su propio análisis. Por consiguiente, se toman los siguientes valores para realizar esta demostración:

Inductancia (L) = 3 H, Resistencia (R) = 4 ohm ,Capacitancia (C) = 0.02 F

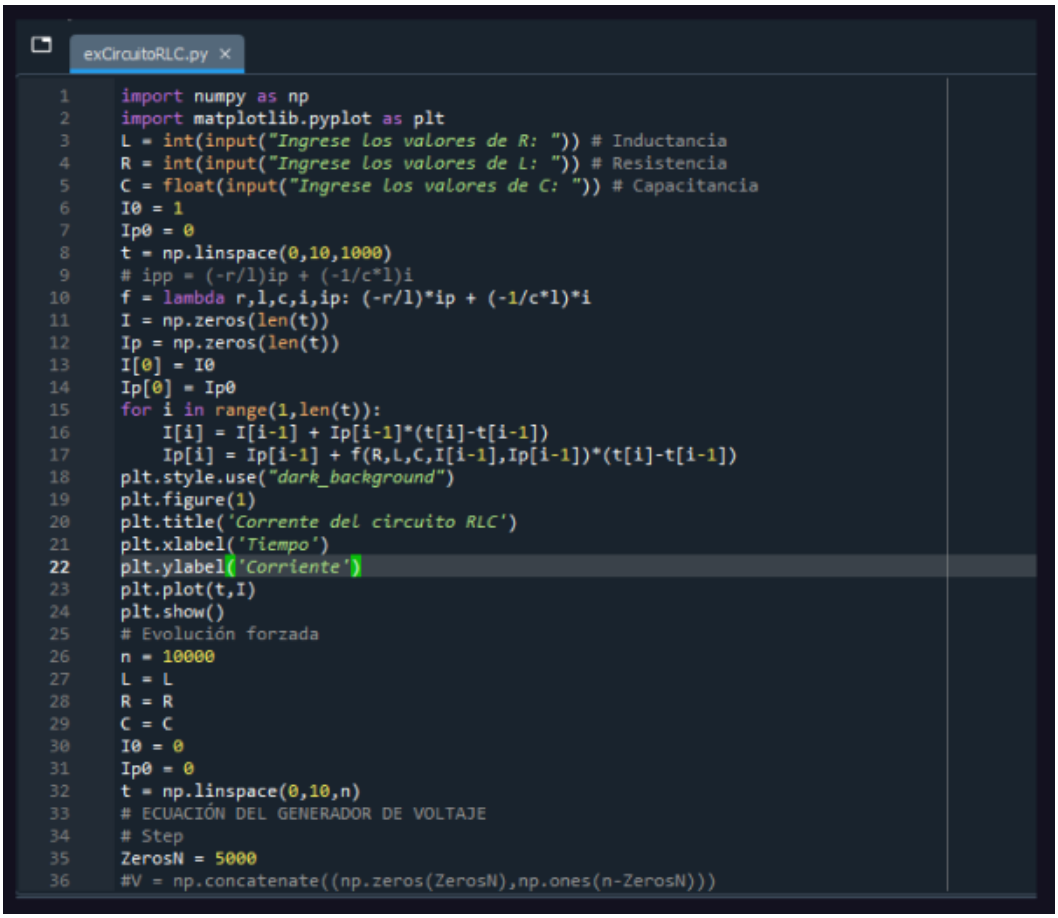


Figura. 5-1 (c) circuito RLC en serie. Fuente: autor

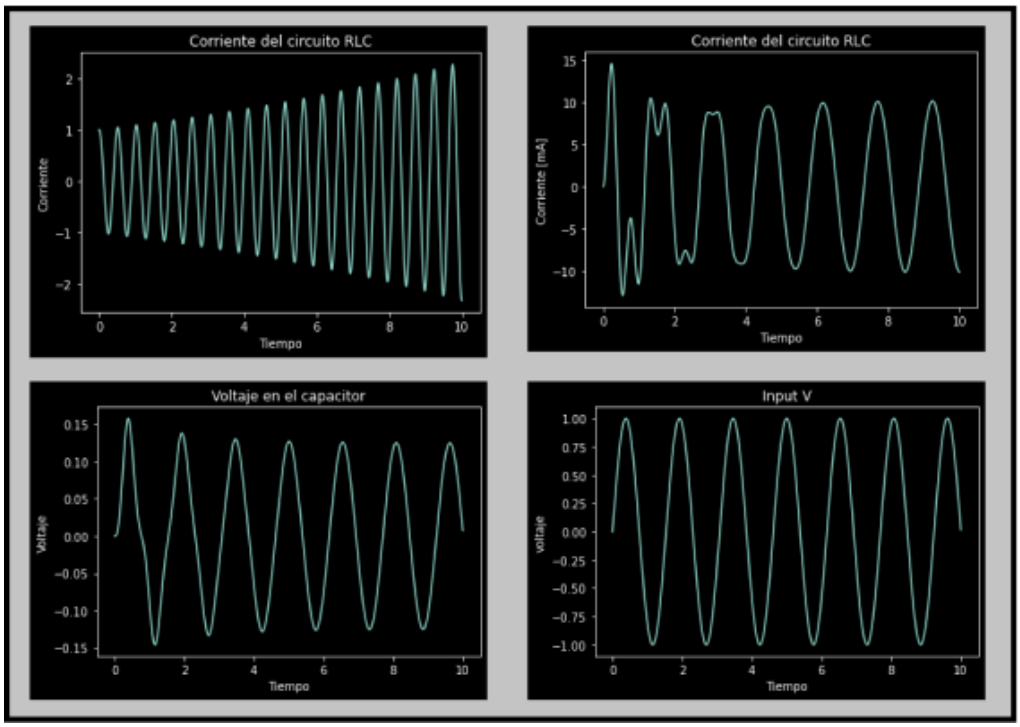


Figura. 5-2 (d) circuito RL en serie. Fuente: autor

6. Resultados esperados

A continuación, se presenta la siguiente encuesta digital que corresponde a la respectiva información recolectada a los estudiantes del área de ingeniería electrónica en la asignatura de señales, sistemas y control dinámico de la cual se hizo una observación a la tendencia de las respuestas obtenidas para poder brindar una conclusión acertada mediante un detallado análisis. https://docs.google.com/forms/d/e/1FAIpQLSdmhaYTcCwaO_W8NeWoTA9zkc__29uuiho5Q65A94M3siOKmw/viewform

Al finalizar el proceso de investigación de este trabajo integral de grado, se obtuvieron los siguientes resultados:

- Ambiente de simulación de ejercicio sobre señales, sistemas y control dinámico en dispositivos móviles.
- Anexo de los ejercicios implementados, algunos realizados de forma convencional y pasados en el lenguaje de programación Python, para finalmente incluir en el Framework Kivy.
- Anexo del Mock Up App Móvil con Figma de cada Screen visto en la aplicación móvil.
- Anexo del Manual de usuario.

6.1 Requerimientos mínimos de hardware y software

Los requerimientos mínimos en relación al hardware que se necesitarán para la implementación y buen funcionamiento del software ya sea en el computador personal o en el dispositivo móvil.

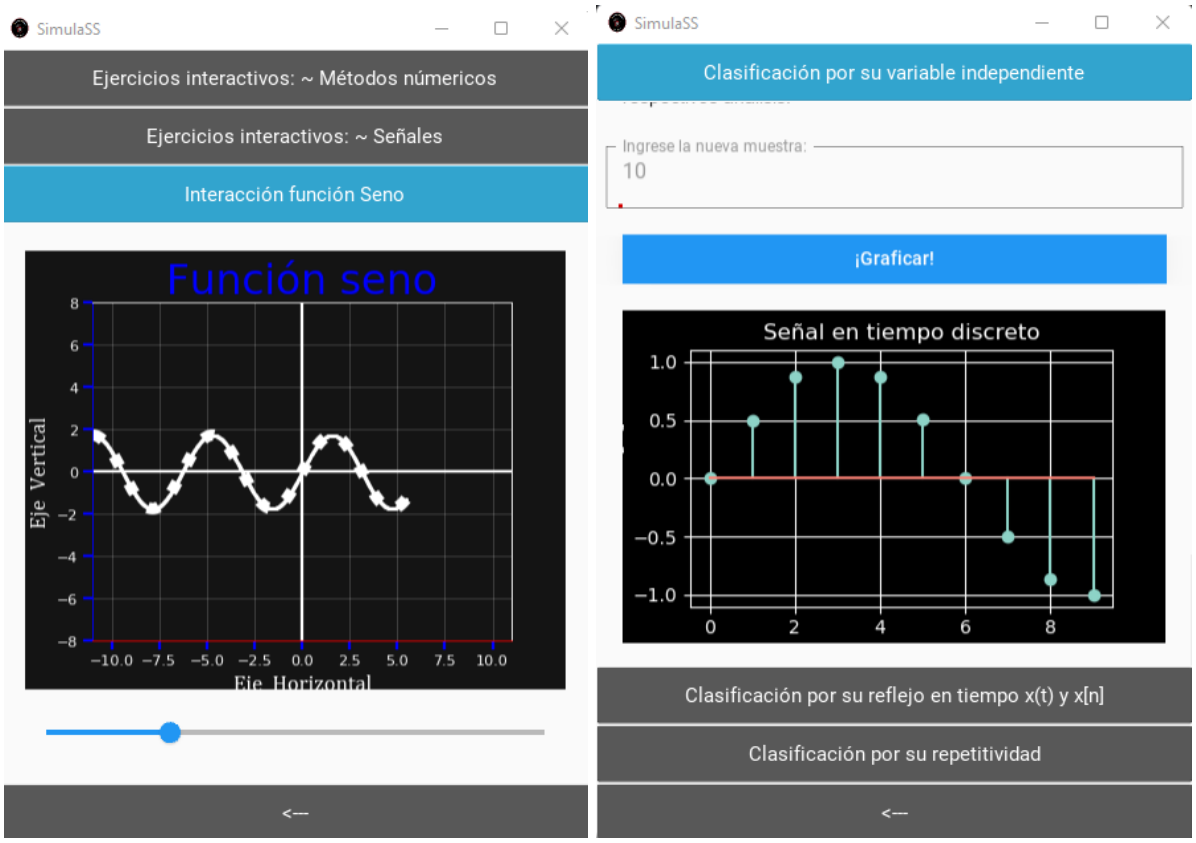
Tabla 6-1

Requerimientos mínimos de hardware y software

Computador personal (PC)	Dispositivo móvil
Monitor	Sistema Operativo Android
1 Gb disponible de espacio en el disco duro	1 Gb disponible en memoria interna
Memoria RAM mínima de 1 Gb	Memoria RAM mínima de 1 Gb
Tarjeta gráfica AMD RADEON Graphics	Procesador Snapdragon, Media Tek, Unisoc, Octa-core y Quad-core

6.2 Validación de la APK

El usuario experimenta con ejercicios interactivos, en el cual puede avriar cambiando los parametrps de ingreso al sistema y simular para ver su resultado, asegurando confianza y calidad de la aplicación al indagar y viajar por todas los screen que este ofrece.



Conclusiones

El buen uso de los dispositivos móviles, optimiza el desarrollo de cualquier actividad productiva de las personas. SimulaSS es una herramienta para afianzar conocimientos previos en la cual pretende cambiar el ámbito educativo.

Con el desarrollo de esta aplicación móvil, se puede demostrar la capacidad que tiene el lenguaje de programación Python, trabajando junto a sus librerías de ciencia de datos y desarrollo de software.

Es necesario recalcar el manejo del framework kivy para el desarrollo de la aplicación móvil, debido a que no es muy utilizado y/o conocido, por aficionados o programadores.

Anexos

A. Anexo: Implementación Framework Kivy

Figura. Anexo A: Implementación Framework Kivy. Fuente: autor

B. Anexo: Mock up de la App Móvil con Figma

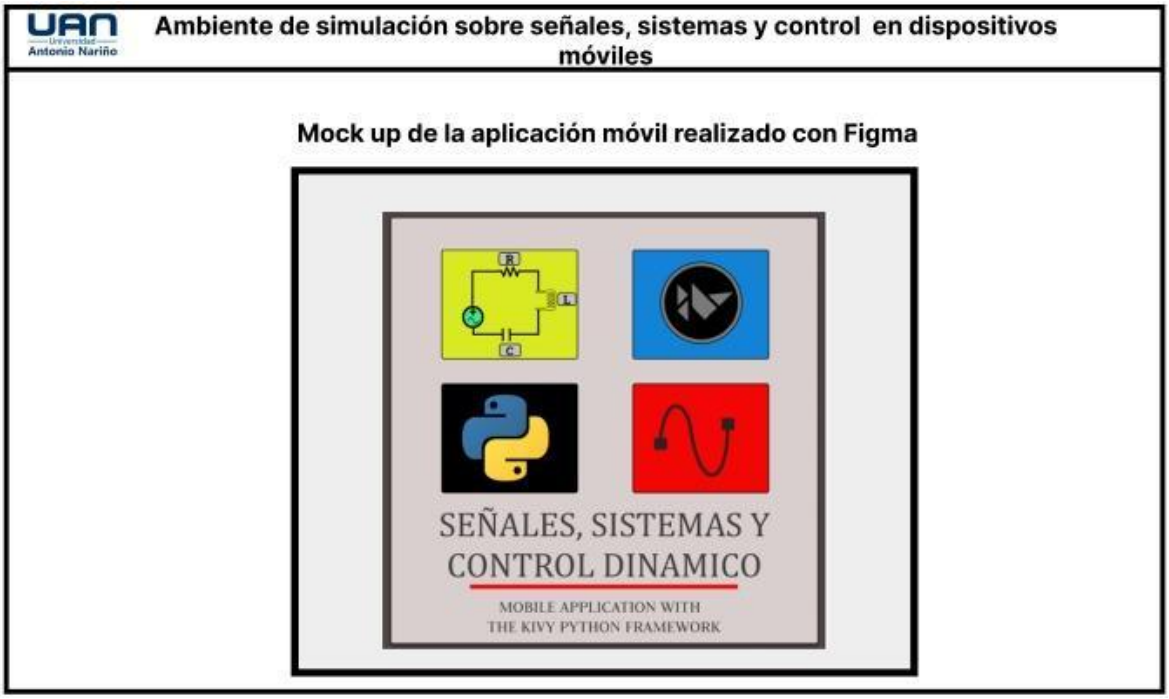


Figura. Anexo B: Mock Up app móvil Fuente: autor

C. Anexo: Manual de usuario de la aplicación móvil SimulaSS

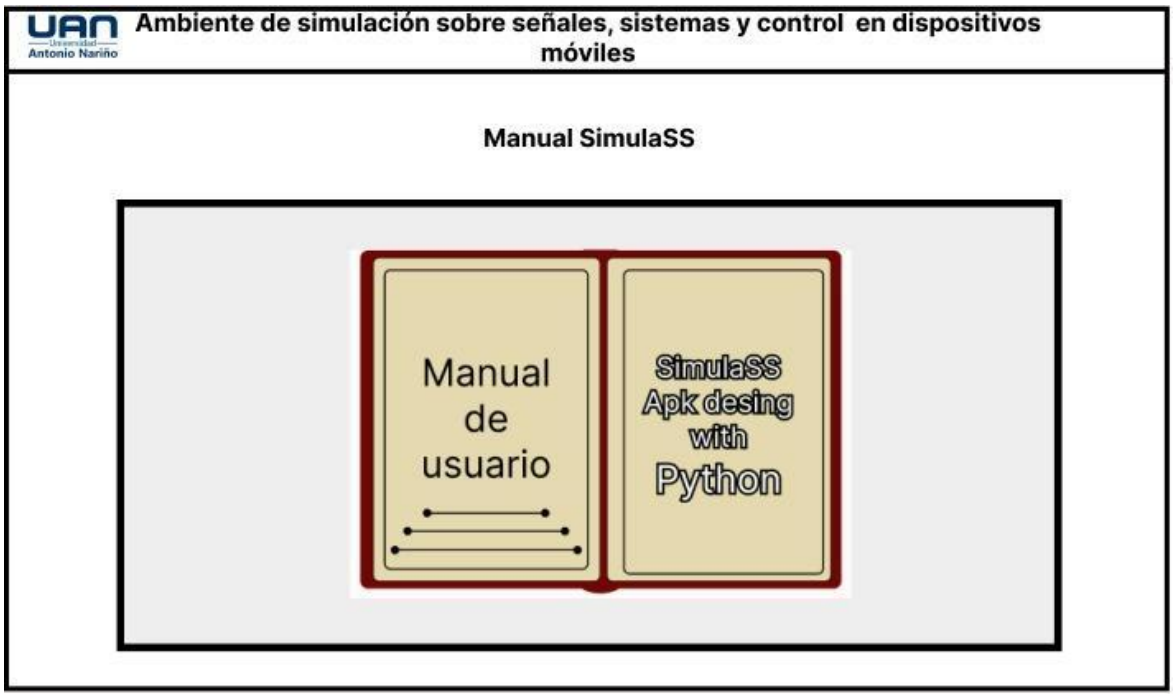


Figura. Anexo C: Manual SimulaSS Fuente: autor

Referencias Bibliográficas

- [1] P. Fabo, M. Skovajsa and L. Pepucha, "PSE simulation environment for the real-time simulation and control of mechatronic systems," 2016 17th International Conference on Mechatronics - Mechatronika (ME), 2016, pp. 1-5.
- [2] Aplicaciones educativas para Android - Sitio oficial de Lihuen. Lihuen.info.unlp.edu.ar https://n9.cl/kkepj
- [3] Hans-Petter Halvorsen. “Mass-Spring-Damper System with Python” https://n9.cl/a50lt
- [4] Codici-Video-Youtube/Circuito RLC at master · Curiosidata/Codici-Video-Youtube. GitHub - https://n9.cl/9ogta
- [5] Raul E. Lopez Briega, “Matemáticas, análisis de datos y Python” 2015 https://n9.cl/xqrlo
- [6] Pablo Antonio Riveraz Hondeoy, “Circuitos RL y RC eléctricos” 2020 – SCRIB, unidad lll parte ll. https://n9.cl/yf5ql
- [7] What is Convolution in Signals and Systems?, Tutorialspoint.com https://n9.cl/iv0rt
- [8] Laplace Transforms – Dynamics and Control 2021, Apmonitor.com https://apmonitor.com/pdc/index.php/Main/LaplaceTransforms
- [9] Sergio Giraldo, Lazo Abierto y Lazo Cerrado - [Sistemas de Control] – Control AutomáticoEducación//https://controlautomaticoeducacion.com/control-realimentado/lazo-abierto-y-lazo-cerrado/

- [10] Python Control System Library, PyPi - https://pypi.org/project/control/?msclkid=cfb21c5dcfa511ecb81d8163bca5e60a
- [11] Welcome to Python.org, Python.org - https://www.python.org/about/
- [12] Christopher Tozzi, Most popular programming Languages: What's Not in 2022 ITPro Today: IT News, How-Tos, Trends, Case Studies, Career Tips, More. https://www.itprotoday.com/programming-languages/most-popular-programming-languages-what-s-hot-what-s-not-2022
- [13] Damián A. Spyder, un potente entorno de desarrollo interactivo para Python. Ubunlog https://ubunlog.com/spyder-entorno-desarrollo-python/
- [14] FIGMA: diseña webs y apps de forma online y colaborativa. - Euskadi+innova-Spri.eus https://n9.cl/113u8
- [15] Kivy: Cross-platform Python Framework for NUI, Kivy.org https://kivy.org/#home
- [16] SymPy - Quick Guide, Tutorialspoint.com -https://www.tutorialspoint.com/sympy/sympy_quick_guide.htm
- [17] Matplotlib Tutorial, W3schools.com - https://www.w3schools.com/python/matplotlib_intro.asp
- [18] Screen Manager — Kivy 2.1.0 documentation, Kivy.org https://kivy.org/doc/stable/api-kivy.uix.screenmanager.html
- [19] Accordion — Kivy 2.1.0 documentation, Kivy.org - https://kivy.org/doc/stable/api-kivy.uix.accordion.html

- [20] RedHat, El concepto de IDE 2019

  https://www.redhat.com/es/topics/middleware/what-is-ide

- [21] venv — Creación de entornos virtuales — documentación de Python - 3.8.13.

  Docs.python.org 2022- https://n9.cl/p96ze

La idea de la aplicación móvil surge en el momento de cursar las asignaturas señales, sistemas y teoría de control, notando la necesidad de complementar el proceso de formación tradiciónal integrado (complementado) a las aplicaciones tecnológicas.

Hablo de

Hice una encuesta que en términos generales con tantos estudiantes me dijeron que estaban de acuerdo con la apk. En los anexos esta la descriocion de la encuenta.

Hablar en tercera persona

De la respuesta

Hice una encuesta con los compañeros de la asignatura que son 12, de todos el 80% responden que si es necesario el software para ayudar en la materia. Se realiza con 12 porque son los compañeros con las que se cuenta en la asignatura.

Con base en la encuenta que realice a los compañeros

Con base en los resultados y la confirmación de la necesidad ee la aplicación, empecé a desarrollar la idea, me fije unos objetivos, busque las metodologías e iniciativas para constuirlos y asi mismo llegar a los resultados planteados

Una diapositiva con los objetivos

Diapositivas claves

Resaltar siempre los logros

Aplicaicon enfocada a esas materias que da la universidad, no se compara con otras, con el pensum nuestros, pensada y diseñada para este pensum

1er diapositiva ~ Presentación

Buenos días, me presento, mi nombre es Andres Felipe Jimenez vega, sustentando mi trabajo integral de grado el cual se basa en un **ambiente de simulación sobre señales, sistemas y control en dispositivos móviles**, como director del proyecto se encuentra presente el doctor Sergio Orjuela.

2da diapositiva ~ introducción

El proyecto consiste en el desarrollo de una aplicación móvil y mediante la implementación de la iniciativa CDIO donde esta tiene como centro fundamental el ciclo de desarrollo de la aplicación, identificando metodologías y herramientas de desarrollo de software, como lo es la

metodología scrum y la herramienta visual studio code y otros editores de texto y grafico que fueron necesarios para su optimo desarrollo

3ra diapositiva ~ planteamiento del problema

La idea de desarrollar una aplicación móvil con carácter científico surge en el momento de estar cursando las asignaturas de señales, sistemas y control, y en esto noto de mi parte la necesidad de afianzar y complementar un poco más el proceso de formación tradicional que vienen impartiendo los profesores implementando alternativas diferentes de aprendizaje que ya existen como lo es el mobile-learning, dando uso apropiado de herramientas tecnologicas y modernas importantes que todos tenemos como lo es el celular, desarrollando una aplicación offline para mas facilidad siendo esta una ventaja para afianzar su conocimiento en cualquier ocasión si necesidad de internet. Esta alternativa representa un cambio en el ámbito de la enseñanza actual ya que cuenta con

simuladores interactivos con capacidad de potenciar la comprensión en las asignaturas mencionadas.

SIGUIENTE DIAPOSITIVA à

4ta diapositiva ~ Encuesta

Debido a esto, realice una encuesta a los compañeros que también cursaban esas asignaturas, donde el 100% responde de manera satisfactoria la necesidad de desarrollar una alternativa de aprendizaje como lo es el uso de las tecnologías modernas y así mismo generar herramientas dinámicas para fortalecer el conocimiento. Gracias a la positiva respuesta de los compañeros, me lleva a formular la siguiente pregunta de investigación: SIGUIENTE DIAPOSITIVA: à

5ta diapositiva ~ pregunta de investigación

¿Cómo se pueden hacer simulaciones de ejercicios de señales, sistemas y control en dispositivos móviles?

6ta diapositiva ~ Iniciativa CDIO

Con base en los resultados y la confirmación de la necesidad de desarrollar la aplicación móvil, respondo a la pregunta de investigación desarrollando la idea, buscando las metodologías e iniciativas para construirlos y así mismo llegar a los resultados planteados. Es por eso que por medio de la iniciativa CDIO, la cual tiene como objetivo el ciclo del desarrollo de la aplicación, el cual separo en fases, la fase de concebir y diseñar y la fase de implementar y operar. Por estos motivos me fije unos objetivos.

7ma diapositiva ~ Objetivos

Objetivo General

- Desarrollar una aplicación para dispositivos móviles, haciendo uso de herramientas de programación científica, para simular ejercicios de señales, sistemas y control.

Objetivos Específicos

- Establecer una ruta didáctica diseñando una interfaz amigable para la comprensión de cada uno de los pasos para la simulación de cada ejercicio.

- Desarrollar algoritmos para la operación de métodos numéricos seleccionados de libros contemplados en la base de datos de la universidad Antonio Nariño.
- Diseñar prácticas de laboratorio para la utilización de la aplicación desarrollada acorde con los resultados de aprendizaje del programa
- Crear una guía para la implementación de simulaciones usando las librerías NumPy, Matplotlib, Scipy, Sympy y el framework kivy en Python.

8va diapositiva ~ Fase de análisis y diseño

En esta fase se realiza la planificación y diseño de la aplicación, comenzando por un proceso de concebir cual es la solución al problema identificando la información necesaria, explorando las herramientas necesarias para iniciar el proceso de diseño y finalmente buscando la

metodología de desarrollo de software a trabajar la cual es la metodología Scrum. SIGUIENTE DIAPOSITIVA:

9na diapositiva ~ metodología scrum

Se escoge esta metodología de desarrollo por su uso en el desarrollo de software ya que se adapta a las necesidades que surgen para desarrollar la aplicación y además cuenta con roles principales de bastante importancia para el ciclo de vida del proyecto como lo son:

El ScrumMaster: Es la persona encargada de dirigir el proyecto. En este caso es el director de la tesis.

ProductOwner: Nicho al cual va dirigida la aplicación. Este trabajo integral de grado, va dirigido a estudiantes de la universidad Antonio Nariño los cuales cursan las asignaturas: señales, sistemas y control. (DUEÑO DEL PRODUCTO)

Development team: Son los encargados de recibir los parámetros que dirige el ScrumMaster. En este caso, es el desarrollador de la aplicación móvil, mi persona por suuesto. (EQUIPO DE DESARROLLO)

10 diapositiva ~ Mock up

Continuando con el diseño de la aplicación móvil, cada interfaz grafica fue diseñada en el editor de graficos figma, haciendo un diseño lo mas realista posible.

11 diapositiva ~ fase de implementación

Continuando con la fase de implementación que brinda la iniciativa CDIO,

Por lo general, es una buena práctica configurar un nuevo entorno para nuevos proyectos como:

Por ejemplo, ML flow requiere una versión inferior de Numpy y cuando intenta instalar ML flow en el directorio base, entra en conflicto con las bibliotecas preinstaladas y dificulta la administración de diferentes versiones.

Ayuda a aislar códigos personalizados y facilita la implementación de su aplicación en cualquier plataforma.

Estamos listos para instalar las librerías requeridas. Como estamos usando python, pip es una excelente manera de instalar y administrar paquetes de python. Para instalar Kivy y sus dependencias, escriba el siguiente comando uno por uno:

https://towardsdatascience.com/building-android-apps-with-python-part-1-603820bebde8

Ayuda a mantener las diferentes versiones de diferentes bibliotecas