DISEÑO Y DESARROLLO DE UNA PLATAFORMA COMPUTACIONAL SOBRE DISPOSITIVOS MÓVILES PARA EMULACIÓN DE SISTEMAS DINÁMICOS

LEANDRO FLÓREZ ARISTIZÁBAL

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
PROGRAMA MAESTRÍA EN INGENIERÍA
SANTIAGO DE CALI
2015

DISEÑO Y DESARROLLO DE UNA PLATAFORMA COMPUTACIONAL SOBRE DISPOSITIVOS MÓVILES PARA EMULACIÓN DE SISTEMAS DINÁMICOS

LEANDRO FLÓREZ ARISTIZÁBAL

Proyecto de grado para optar al título de Magister en Ingeniería con énfasis en informática

Director
Diego Fernando Almario Álvarez
Magister en Automática

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
PROGRAMA MAESTRÍA EN INGENIERÍA
SANTIAGO DE CALI
2015

Nota de aceptación:

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar al título de Magister en Ingeniería

Jimmy Tombe

Jurado

Fabian Andres Gonzalez

Jurado

Santiago de Cali, Agosto 11 de 2015

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

RESUMEN

Este trabajo muestra el diseño y desarrollo de una plataforma computacional sobre dispositivos móviles que servirá de herramienta para el aprendizaje de conceptos de control automático y sistemas dinámicos. Para este propósito, se desarrolló una aplicación móvil para dispositivos con sistema operativo Android que permite simular un control en lazo abierto y cerrado de un sistema masa-resorte-amortiguador. En este sistema, el controlador y la planta trabajan juntos en el mismo dispositivo, lo que se denominó modo embebido. Un modo de trabajo adicional se implementó, que consiste en simular el sistema separando el controlador y la planta para que trabajen de forma independiente en diferentes dispositivos, este forma de trabajo se llamó modo externo.

Como método numérico para la solución de ecuaciones diferenciales del controlador y la planta, se usó un Runge-Kutta de 4to orden en cualquiera de los dos modos de trabajo implementado. Los resultados de la solución de las ecuaciones diferenciales fueron usados para visualizar las variables relevantes del sistema en el dominio del tiempo (posición de la masa, acción de control, error y referencia) e igualmente para crear la animación del sistema.

La aplicación permite parametrizar la planta y el controlador de forma independiente. Además, la aplicación permite al usuario generar perturbaciones l sistema controlado usando el sensor de proximidad del dispositivo.

PALABRAS CLAVE: M-learning. Sistemas dinámicos. Bluetooth. Android. Dispositivos móviles. Control en lazo abierto. Control en lazo cerrado. Runge-Kutta.

INTRODUCCIÓN

Los sistemas dinámicos, son un área de estudio aplicable a diferentes campos del conocimiento, entre ellos la economía, la administración, la biología y la ingeniería. Este tipo de sistemas describen un comportamiento en el cual se presenta un resultado o efecto a causa de eventos presentes y pasados. Desde un punto de vista ingenieril, se podría definir un sistema dinámico como aquel sistema físico cuyo estado evoluciona con el tiempo y su salida dependerá de estados actuales y previos además de sus entradas.

A nivel de ingeniería es importante representar matemáticamente los sistemas dinámicos que se desean estudiar, para tal fin existen representaciones en ecuaciones diferenciales que permiten determinar características importantes del comportamiento del sistema dado que al obtener la solución de las mismas, se puede conocer el funcionamiento del sistema físico en cualquier instante de tiempo y ante diferentes condiciones.

Algunos ejemplos de sistemas dinámicos generalmente estudiados son aquellos que pueden ser representados por tres elementos mecánicos conocidos como masa, resorte y amortiguador. Analógicamente, estos sistemas también pueden ser representados por su equivalente eléctrico, compuesto por condensador, bobina y resistencia. Otros sistemas dinámicos que pueden ser estudiados son la transferencia de líquido entre dos tanques en serie que se encuentren a diferentes alturas, el comportamiento de un ascensor o un avión durante el despegue entre otros.

Este tipo de sistemas son la base para el estudio de la ingeniería de control y su enseñanza en ocasiones puede volverse compleja debido a la falta de herramientas que permitan contextualizar al estudiante en situaciones donde el modelado que describe el comportamiento de estos sistemas se realiza de forma matemática.

Existen diferentes herramientas informáticas que permiten analizar y facilitan el modelado de sistemas dinámicos, el problema es que su uso en ocasiones se ve restringido por diferentes factores como el hardware sobre el cual deben ser ejecutadas, su portabilidad, pago de licencias y usabilidad.

Hoy en día la mayoría de los objetos de aprendizaje se implementan en plataformas tradicionales como computadores de escritorio y portátiles. Lo anterior

se debe comenzar a evaluar pues se han estado posicionando dispositivos móviles como los teléfonos inteligentes (*smartphones*) y las tabletas (*tablets*) que debido a sus altas prestaciones técnicas y la penetración de mercado se pueden considerar como dispositivos en los cuales es necesaria la implementación de herramientas que ayuden al proceso de aprendizaje sin limitantes de tiempo o espacio.

En este contexto, el propósito de este proyecto, es diseñar e implementar una plataforma computacional para dispositivos móviles que permita la emulación de sistemas dinámicos y la interacción con ellos. Para lograrlo, se deberá:

- Seleccionar los tipos de sistemas dinámicos más adecuados a ser implementados en dispositivos móviles de acuerdo a sus características de visualización y posibilidades de interacción con el usuario.

- Proponer una arquitectura para una plataforma que permita la parametrización, simulación y animación de los sistemas dinámicos seleccionados usando dispositivos móviles.

- Desarrollar un prototipo de la plataforma utilizando la arquitectura propuesta mediante parametrización local o inalámbrica de una dinámica o un sistema específico.

- Validar el funcionamiento de la plataforma con potenciales usuarios.

La razón de desarrollar una herramienta móvil es debido a que la computación móvil ha evolucionado de forma acelerada y en beneficio de todos permitiendo tener, literalmente, al alcance de las manos lo necesario para realizar tareas que en el pasado solo podían hacerse en una estación de trabajo fija y con recursos limitados en cuanto a conexiones de red, interfaces de comunicación, almacenamiento y procesamiento.

La inclusión de dispositivos móviles y herramientas virtuales en el proceso de enseñanza se ha vuelto imperioso al ser estos de uso cotidiano para los alumnos y deben ser usados y aprovechados como un recurso para fortalecer los procesos de enseñanza en todas las áreas.

Con el desarrollo de esta plataforma móvil la cual ha sido llamada DSC (Dynamic Systems Control), se ofrecerá a los estudiantes la oportunidad de emular e

interactuar con sistemas y estudiar sus dinámicas de acuerdo a unos parámetros establecidos por medio de sus *smartphones* o *tablets* haciendo uso de interfaces inalámbricas para ejercer control de forma remota. De esta forma, se complementan las bases teóricas y se cuenta con un laboratorio portable capaz de ser usado en cualquier momento y lugar.

Este documento se encuentra organizado de la siguiente forma:

El capítulo 1 da una introducción teórica a la metodología de aprendizaje móvil o M-Learning, los sistemas dinámicos y el sistema operativo Android. El capítulo 2 da a conocer el diseño de la plataforma mientras el capítulo 3 muestra la implementación del prototipo. El capítulo 4 expone las perspectivas futuras de la plataforma y finalmente el capítulo 5 concluye el documento.

1. MARCO TEÓRICO

Este proyecto integra diferentes áreas de la ingeniería y herramientas matemáticas, así como metodologías de enseñanza-aprendizaje para lograr resultados confiables de acuerdo a las necesidades tanto de docentes como estudiantes para reducir el tiempo en la curva de aprendizaje del estudio de sistemas dinámicos, siendo el aprendizaje móvil (m-learning) la metodología sobre la cual se centra el objeto de aprendizaje desarrollado. La investigación teórica abarca temáticas como dispositivos móviles y sus respectivos sistemas operativos, sistemas de control y modelos matemáticos.

1.1 M-LEARNING

El aprendizaje móvil o M-Learning (Mobile Learning en inglés), es una metodología de aprendizaje basada en el uso de dispositivos móviles como herramienta para la adquisición de nuevos saberes y competencias la cual ha sido posible gracias a los avances tecnológicos a nivel de teléfonos inteligentes (Smartphones) y tabletas (Tablets), elementos que con el transcurrir de los días se han vuelto más asequibles e indispensables en la vida de algunas personas a nivel laboral, educativo o personal.

El aprendizaje móvil a través de tecnología inalámbrica, permite tener acceso a recursos y materiales sin importar el lugar o momento en el que se encuentre[1]. En su libro, el autor menciona que el resultado de esta disponibilidad es la posibilidad que tienen los aprendices de tener control de su proceso de aprendizaje sin restricciones de tiempo o espacio.

Investigadores del Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CISESE) y el Instituto de Investigación y Desarrollo Educativo de la Universidad Autónoma de Baja California (IIDE-UABC)[2], encuentran relevantes cuatro elementos en el uso y desarrollo de aplicaciones para Smartphones en un contexto de aprendizaje móvil (ver Figura 1), estos son Condición de proximidad personal, Movilidad, Conectividad y Espontaneidad.

[1] ALLY, Mohamed. Mobile Learning .Transforming the Delivery of Education and Training. AU Press, Athabasca University. Canadá.2009 p.5

[2] SERRANO SANTOYO Arturo y ORGANISTA SANDOVAL Javier. Challenges and Opportunities to Support Learning with Mobile Devices. MexIHC' 2010.p.6

Figura 1. Características del Smartphone involucradas en el aprendizaje

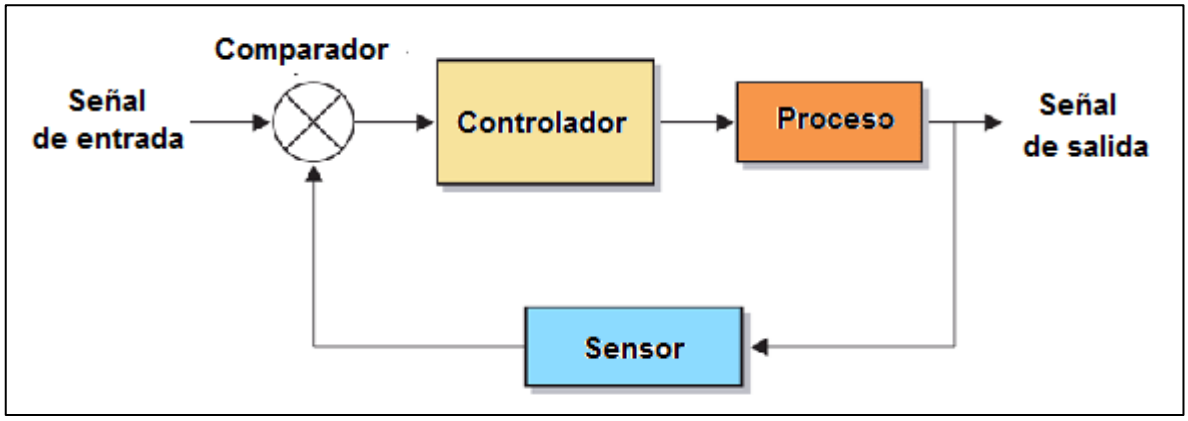
Fuente: SERRANO SANTOYO Arturo y ORGANISTA SANDOVAL Javier. Challenges and Opportunities to Support Learning with Mobile Devices. MexIHC' 2010.p9

Haciendo uso de esta metodología, es posible ofrecer herramientas que faciliten el aprendizaje de diferentes áreas de la ingeniería, entre ellas los sistemas dinámicos, pues aunque se cuenta con software especializado y laboratorios que acercan al estudiante a la realidad del comportamiento de dichos sistemas, en ocasiones las limitaciones de tiempo, espacio y disponibilidad dificultan el acceso a estas herramientas.

1.2 SISTEMAS DINÁMICOS

Durante los procesos de enseñanza del control es necesario pasar por el modelamiento matemático del sistema de tal manera que se pueda observar su comportamiento, previo a la fase del diseño del sistema de control. El esquema de control básico estándar puede ser representado por un esquema como el de la Figura 2.

Figura 2. Bucla típica de control realimentado



Fuente: Grupo PRISA DIGITAL. Tipos de Sistemas de Control. 2011.

Este sistema incluye el proceso (sistema dinámico) a controlar (en este caso incluído el actuador), el contolador y el sensor (el cual en algunos casos se incluye como parte del proceso). El objetivo del sistema como tal es alcanzar unas especificaciones deseadas de funcionamiento de la señal de salida (variable controlada) que no pueden ser logradas por el proceso de manera autónoma. Para lograr dicho objetivo se debe diseñar un controlador mediante procesos ya establecidos, el cual finalmente interactuará con el proceso de la manera que se muestra en la figura anterior para lograr el cumplimiento de lo especificado.

Durante el desarrollo del proceso de diseño del controlador, se interactúa repetidamente con ecuaciones diferenciales teniendo en cuenta que tanto el controlador como el proceso pueden ser representados por una de ellas. Parte de la interaccion corresponde al proceso de simulación del proceso a controlar con el objetivo de determinar su estado actual y que tan alejado se encuentra de las especificaciones de diseño planteadas.

Los procesos a controlar pueden ser de diferente dinámica lo cual se relaciona con el tipo de representacion en ecuaciones diferenciales que lo modele. Un sistema dinámico es aquel cuyo estado cambia con el tiempo[3] y la dinámica de muchos de estos ya sean mecánicos, eléctricos, económicos o biológicos solo por mencionar algunos, se describe en términos de ecuaciones diferenciales las cuales son obtenidas a partir de leyes físicas que gobiernan un sistema determinado, como

[3] ARROWSMITH, D.K. Y PLACE, C.M. An Introduction to Dynamical Systems. Cambridge University Press. 1994.p.6

las leyes de Newton para sistemas mecánicos y las leyes de Kirchhoff para los eléctricos[4].
Los sistemas dinámicos se pueden clasificar entre otras características de acuerdo al grado de las ecuaciones que los representan o en función de su linealidad de las mismas bien sea como sistemas lineales o sistemas no lineales. Acorde con lo planteado en los alcances, este trabajo se centrará en los sistemas lineales, no obstante la estructura quedará abierta para poder integrar posteriormente sistemas no lineales a la misma.

1.2.1 Sistemas Lineales. Los sistemas lineales como su nombre lo indica, son modelados a partir de ecuaciones lineales y el principio de superposición es aplicable a ellos.

A manera de ejemplo se ilustran algunas de las estructuras típicas en la dinámicas de procesos:

- **Sistemas de primer orden:**

$$\tau \frac{dy}{dt} + y(t) = ku(t) \quad Ec.1$$

- **Sistemas de segundo Orden:**

$$\frac{d^2y}{dt^2} + 2\delta W_n \frac{dy}{dt} + W_n^2 y(t) = W_n^2 u(t) \quad Ec.2$$

- **Sistemas con retardo:**

$$\tau \frac{dy}{dt} + y(t) = ku(t - t_o) \quad Ec.3$$

De manera semejante se opera con los controladores; son ecuaciones diferenciales diseñadas a partir de las especificaciones deseadas y finalmente operarán en conjunto (en lazo cerrado) con la ecuación que representa al proceso.

Estos son algunos de los tipos de dinámicas que se pueden encontrar en los procesos y que podrían ser objeto de implementación bajo la plataforma

[4] OGATA, Katsuhiko. Ingeniería de Control Moderna 4 Ed. Madrid.:Pearson Educación, S.A. 2003.p.8

planteada. El sistema dinámico sobre el cuál se realizará el prototipo de la plataforma DSC es el Masa-Resorte-Amortiguador, un sistema típico en la enseñanza de sistemas dinámicos y que desde el punto de vista de la animación, facilita observar diferentes dinámicas de forma clara.

1.2.2 Modelos Matemáticos. Un modelo matemático de un sistema dinámico está definido por un conjunto de ecuaciones que representan las dinámicas del sistema. Dichos modelos pueden tomar muchas formas y dependiendo del sistema y las circunstancias en particular, un modelo puede ser más conveniente que otros[5].

Al obtener un modelo matemático, se debe tener en cuenta que un modelo que se aproxime mucho más al comportamiento de un sistema real, será a su vez más complejo, por lo cual debe buscarse una relación entre simplicidad y precisión.

Retomando las ecuaciones 1, 2 y 3, (Ec. 1, Ec.2, Ec. 3) el modelo matemático de los siguientes sistemas vienen dados por:

- **Sistemas de primer orden:**

$$G(s) = \frac{K}{\tau s + 1} \quad Ec.4$$

- **Sistemas de segundo Orden:**

$$\frac{Y(s)}{U(s)} = \frac{KWn^2}{s^2 + 2\delta Wns + Wn^2} \quad Ec.5$$

- **Sistemas con retardo:**

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-\tau_o s}}{\tau s + 1} \quad Ec.6$$

[5] Ibid., p. 53.

1.2.3 Funciones de Transferencia. Son funciones que permiten caracterizar la relación de entrada y salida de sistemas de ecuaciones diferenciales lineales, invariantes en el tiempo. Esta relación debe hacerse a partir de las transformadas de Laplace de la entrada (función impulsora) y la transformada de Laplace de la salida (función de respuesta), en otras palabras, la función de transferencia de un sistema es un modelo matemático que implica un método operacional de expresar la ecuación diferencial que relaciona la variable de salida con la variable de entrada[6].

Bajo el concepto de función de transferencia es posible diseñar como ejemplo un sistema de control el cual servirá como método para conseguir la modificación del comportamiento del sistema.

La función de transferencia de un controlador Proporcional-Integral típico es:

$$Gpi_{(s)} = Kp + \frac{Ki}{s} \qquad Ec.7$$

1.2.4 Sistema Masa-Resorte-Amortiguador. Este es un sistema dinámico de tipo mecánico el cual posee tres tipos de elementos básicos[7]:

- Elementos de inercia
- Elementos de resorte
- Elementos amortiguadores

Inercia es la resistencia de un objeto al cambio de sus estado de movimiento o al cambio en la fuerza requerida para generar un cambio en la aceleración. De acuerdo a esto, la masa (m) es la propiedad física que provee la inercia a los objetos. Por la segunda ley de Newton del movimiento, la fuerza es:

$$F_{(t)} = m.a = m\frac{d^2x_{(t)}}{dt^2} \quad Ec.8$$

Los resortes lineales son elementos mecánicos que pueden ser estirados o comprimidos por una fuerza externa para deformarlo proporcionalmente a la fuerza aplicada. Esta deformación solo es un cambio en términos de

[6] OGATA, Katsuhiko. Dinámica de Sistemas. Prentice-Hall Hispanoamericana, S.A. Naucalpán de Juarez. 1987.p.9

[7] Aziz A. “Mechanical Systems,” in Systems Dynamics & Control. [En línea].En: AIAA Journal, VOl. 3, no. 4. abril. 2007. p. 678-685.[consultado 15 de marzo de 2015]Disponible en Internet: http://arc.aiaa.org/doi/abs/10.2514/3.2947

desplazamiento y es determinado por una constante de elasticidad (k). La ley de Hooke gobierna la fuerza asociada a un resorte:

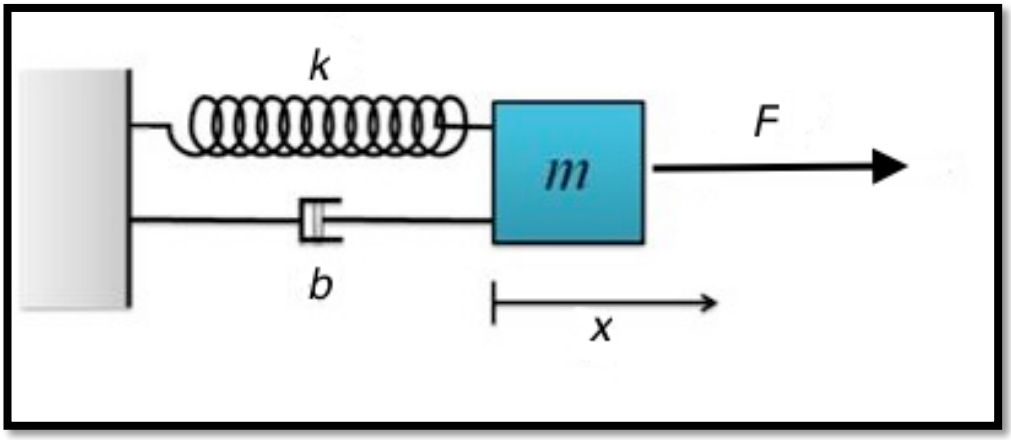
$$F_{(t)} = kx_{(t)} \qquad Ec.9$$

Un amortiguador disipa la energía y su propósito es aplicar una fuerza resistiva determinada por una constante de amortiguamiento (b). La ecuación diferencia de la fuerza es entonces:

$$F_{(t)} = b.v = b\ \frac{dx_{(t)}}{dt} \qquad Ec.10$$

Estos 3 elementos combinados hacen parte de un sistema traslacional denominado masa-resorte-amortiguador como se observa en la siguiente figura:

Figura 3. Sistema Masa-Resorte-Amortiguador



Fuente: Introduction to Dynamic Simuation. [En línea].En: NATIONAL INSTRUMENTS. 2015[consultado 16 de marzo de 2015] Disponible en internet: http://www.ni.com/tutorial/11606/en/#reviews

Por la segunda ley de Newton, la ecuación diferencial que representa el comportamiento de este sistema es:

$$f(t) = m\ \frac{d^2x_{(t)}}{dt} + b\ \frac{dx_{(t)}}{dt} + kx_{(t)} \qquad Ec.11$$

La relación entre la respuesta del sistema modelado y la señal de entrada o excitación de dicho sistema es de vital importancia en el marco del estudio de los

sistemas dinámicos y usualmente se trabaja en el dominio complejo, al cual se llega a través de la transformada de Laplace considerando condiciones iniciales igual a cero. El resultado de esta operación permite encontrar la denominada función de transferencia.

Para un sistema masa-resorte-amortiguador del cual se conoce su modelo como ecuación diferencial (*Ec. 11*), usando una representación como función de transferencia se obtiene:

$$Gp_{(s)} = \frac{1}{ms^2 + bs + k} \qquad Ec.12$$

Los modelos formulados como funciones de transferencia permiten predecir el comportamiento del sistema al igual que lo que se logra en el modelo representado por ecuaciones diferenciales, pero la solución de estas últimas como las vistas anteriormente (*Ec. 8, Ec. 9, Ec. 10, Ec. 11*) es mucho más fácil de implementar en sistemas de cómputo gracias al uso de herramientas matemáticas como los métodos numéricos.

1.2.5 Métodos Numéricos. Los métodos numéricos son técnicas que permiten resolver problemas matemáticos por medio de operaciones aritméticas. Estos combinan dos herramientas imprescindibles en la ingeniería, las cuales son las matemáticas y las computadoras convirtiéndose en herramientas muy poderosas capaces de manejar sistemas de ecuaciones grandes, no linealidades y geometrías complicadas, comunes en la práctica de la ingeniería y a menudo imposibles de resolver analíticamente, por lo tanto aumentan la habilidad de quien los estudia para resolver problemas[8].

Las ecuaciones planteadas como representación de algun sistema dinámico deberán ser solucionadas por métodos numéricos como: Método de Euler, de Taylor, de Runge Kutta entre otros, los cuales han sido ampliamente validados y se implementan comunmente en sistemas computacionales.

1.2.6 Runge-Kutta 4to Orden. Conocido también como RK4, este es un método numérico iterativo implementado en muchos sistemas de álgebra computacional, publicado en 1895 por Carl Runge y fue generalizado en 1901 a sistemas de ecuaciones diferenciales ordinarias (EDO) por M. Wilhelm Kutta.

La fórmula de RK4 para estimar la solución de ecuaciones diferenciales es:

[8] CHAPRA, Steven C y CANALE, Raymond P. Métodos numéricos para ingenieros Ed. 5. McGraw Hill. 2007.p.8

$$y_{i+1} = y_i + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4] \quad Ec.13$$

donde,

$$k_1 = h.f(x_i, y_i) \quad Ec.14$$

$$k_2 = h.f\left[x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right] \quad Ec.15$$

$$k_3 = h.f\left[x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right] \quad Ec.16$$

$$k_4 = h.f[x_{\square} + h, y_i + k_3] \quad Ec.17$$

En cada caso (Ec.14 – Ec.17) *h* es el paso en cada iteración.

Implementar este método numérico por medio de software es bastante simple independiente del lenguaje de programación o el dispositivo de cómputo usado, incluyendo dispositivos móviles como smartphones o tablets. Estos dispositivos son gobernados por un sistema operativo y en el desarrollo de la plataforma DSC se decidió hacer sobre plataformas Android.

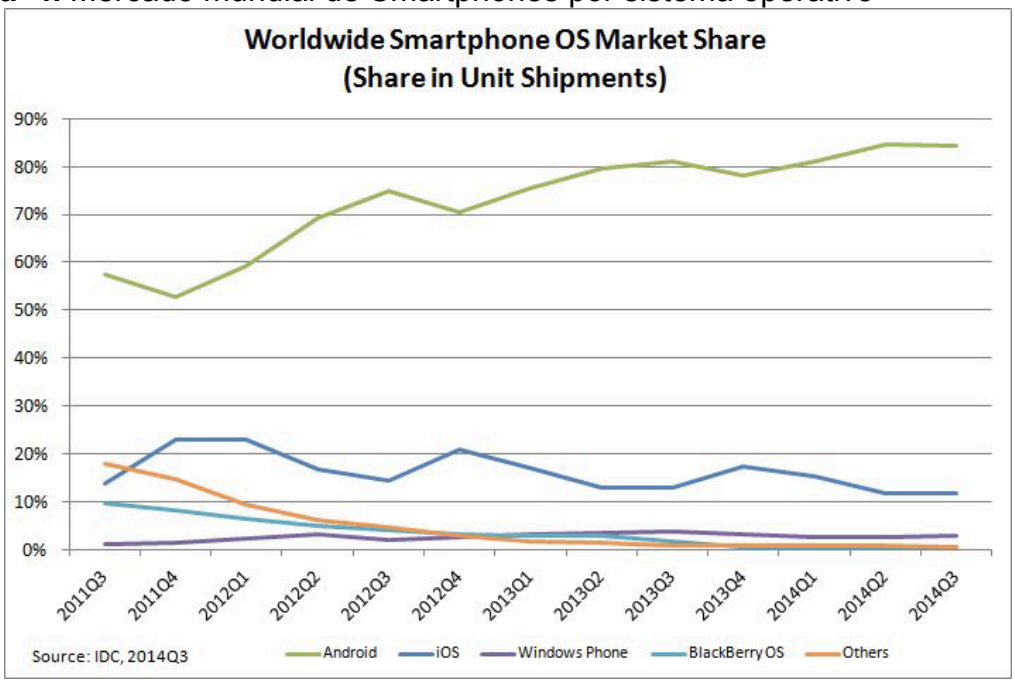
1.3 ANDROID

Android (Google) es un Sistema Operativo para *smartphones*, el cual está en continuo proceso de desarrollo con versiones muy estables hoy en día como *Jelly Bean* (Android 4.1 – 4.3), KitKat (Android 4.4.4) y Lollipop (Android 5.0). Android está basado en el Kernel de Linux y utiliza una máquina virtual diseñada para optimizar los recursos de memoria y hardware en un ambiente móvil. Su código es abierto y puede extenderse para incorporar tecnologías emergentes[9].

Para el tercer trimestre del 2014 (2014Q3), este sistema operativo lidera el mercado con un 84.4% de dispositivos vendidos[10]. Una de las razones de este alto porcentaje son los diferentes fabricantes que producen dispositivos de diversas gamas con este sistema operativo, lo que permite el acceso a un mayor número de usuarios.

[9] Android-Overview. [En línea]. En: OPEN HANDSET ALLIANCE. [consultado 8 de mayo de 2015] 2013. Disponible en Internet:http://www.openhandsetalliance.com/android_overview.html
[10] Smartphone OS Market Share, Q3 2014. [En línea] En: International Data Corporation. 2015. Disponible en Internet: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

Figura 4. Mercado mundial de Smartphones por sistema operativo



Fuente: Smartphone OS Market Share, Q3 2014. [En línea] En: International Data Corporation. 2015. Disponible en Internet: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

1.3.1 Aplicaciones Android. Las aplicaciones para dispositivos Android están clasificadas en 4 categorías[11]

- Aplicaciones nativas que se desarrollan usando el IDE oficial (Entorno de Desarrollo) que anteriormente era Eclipse y actualmente es Android Studio.

- Aplicaciones web móviles genéricas que son sitios web diseñados para dispositivos móviles y pueden ser usadas independiente de la plataforma.

- Aplicaciones web dedicadas que son acondicionadas para una plataforma específica (Android, iOS, Blackberry).

- Aplicaciones híbridas que son una combinación entre una aplicación nativa y web.

Las aplicaciones nativas permiten tener un mayor control en el desarrollo y no posee restricciones a nivel de acceso a hardware, por lo que su uso es necesario

[11] GOK, Nizamettin y KHANNA, Nitin. Building Hybrid Android Apps. O'Reilly. USA. 2013.p.65

en aplicaciones con funcionalidades muy específicas y que requieran sacar el mayor provecho a los recursos del dispositivo.

- **Bloques Principales.** Una aplicación Android posee cuatro componentes y cada uno es un punto diferente por el cual el sistema puede entrar a la aplicación. Estos componentes son:

  - Actividades (Activities): Representa una pantalla con una interfaz de usuario.

  - Servicios (Services): Es un componente que se ejecuta de fondo en operaciones de larga ejecución o en trabajo con procesos remotos.

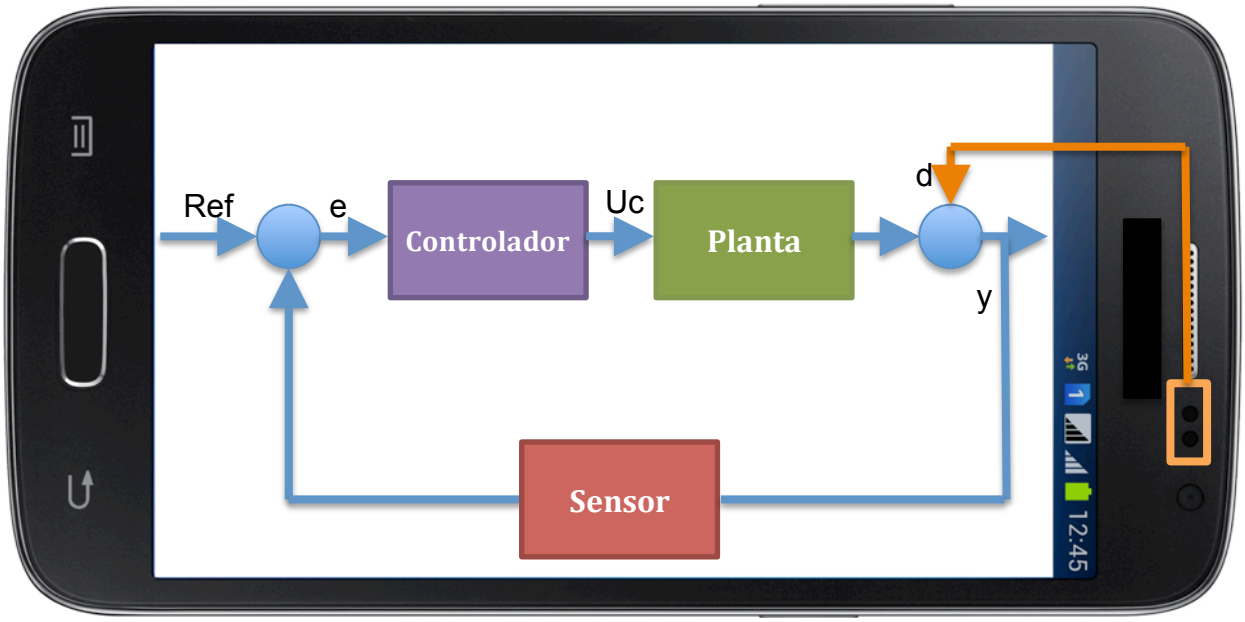
  - Proveedores de contenido (Content Providers): Gestiona datos compartidos entre diferentes aplicaciones.

  - Receptores de difusión (Broadcast Receivers): Componente que responde a anuncios difundidos por el sistema.

2. DISEÑO DE LA PLATAFORMA DSC

La plataforma se diseñó con el propósito de ofrecer a los estudiantes de la Universidad Autónoma de Occidente una herramienta en la cual pudieran emular el funcionamiento de sistemas dinámicos desde sus dispositivos móviles. Hay que recordar que un sistema controlado consta de unos elementos comunicados entre sí formando lo que se conoce como la bucla típica de control y el propósito es embeberla en un mismo dispositivo como se puede observar en la siguiente figura.

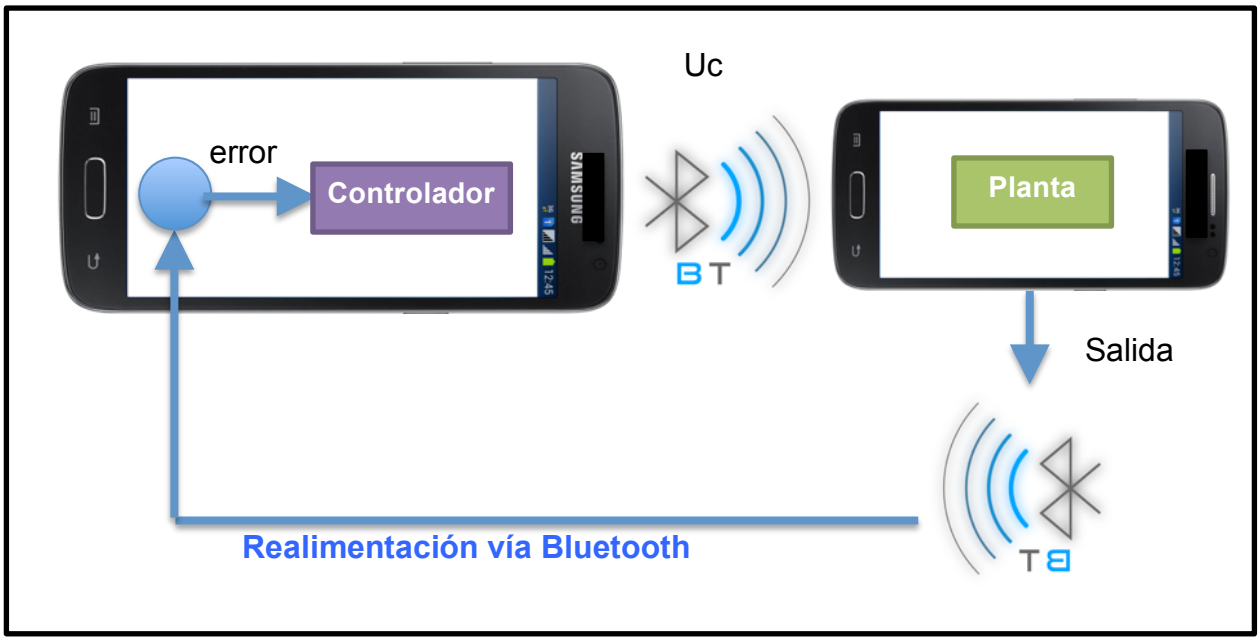
Figura 5. Bucla típica de control implementada en un dispositivo móvil



Fuente: elaboración propia

Por otra parte, se requirió la implementación del sistema en lazo cerrado distribuyendo el control y la planta en diferentes dispositivos y de esta forma lograr un control a distancia por medio de una interfaz Bluetooth.

Figura 6. Interfaz inalámbrica Bluetooth para control externo



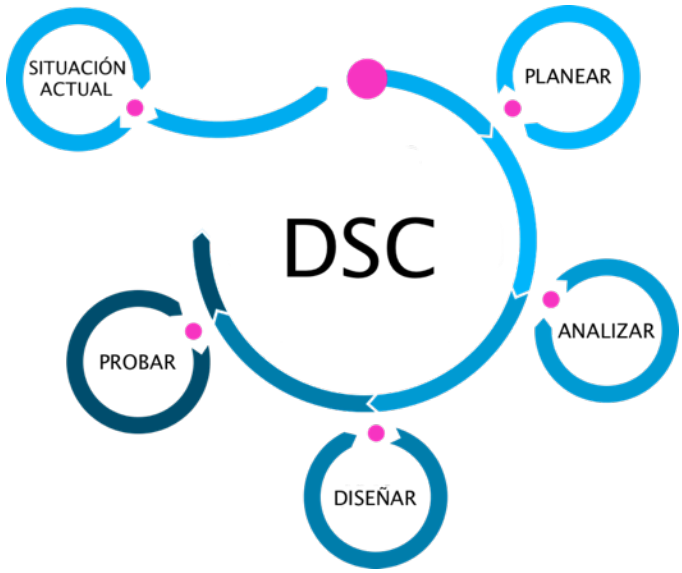
Fuente: elaboración propia

2.1 METODOLOGÍA

La metodología usada en el desarrollo de este proyecto es Diseño Centrado en el Usuario, debido a que se desarrolló una plataforma orientada a dispositivos móviles para usuarios muy específicos de la Universidad Autónoma de Occidente.

2.1.1 Diseño Centrado en el Usuario. El proceso del Diseño Centrado en el Usuario está compuesto de varios métodos y tareas iterativas en las cuales debe estar involucrado el usuario. Cuatro fases se definieron para el desarrollo de la plataforma, dentro de las cuales se incluyen la planeación, el análisis, el diseño y las pruebas. De forma general, el proceso se describe de la siguiente manera.

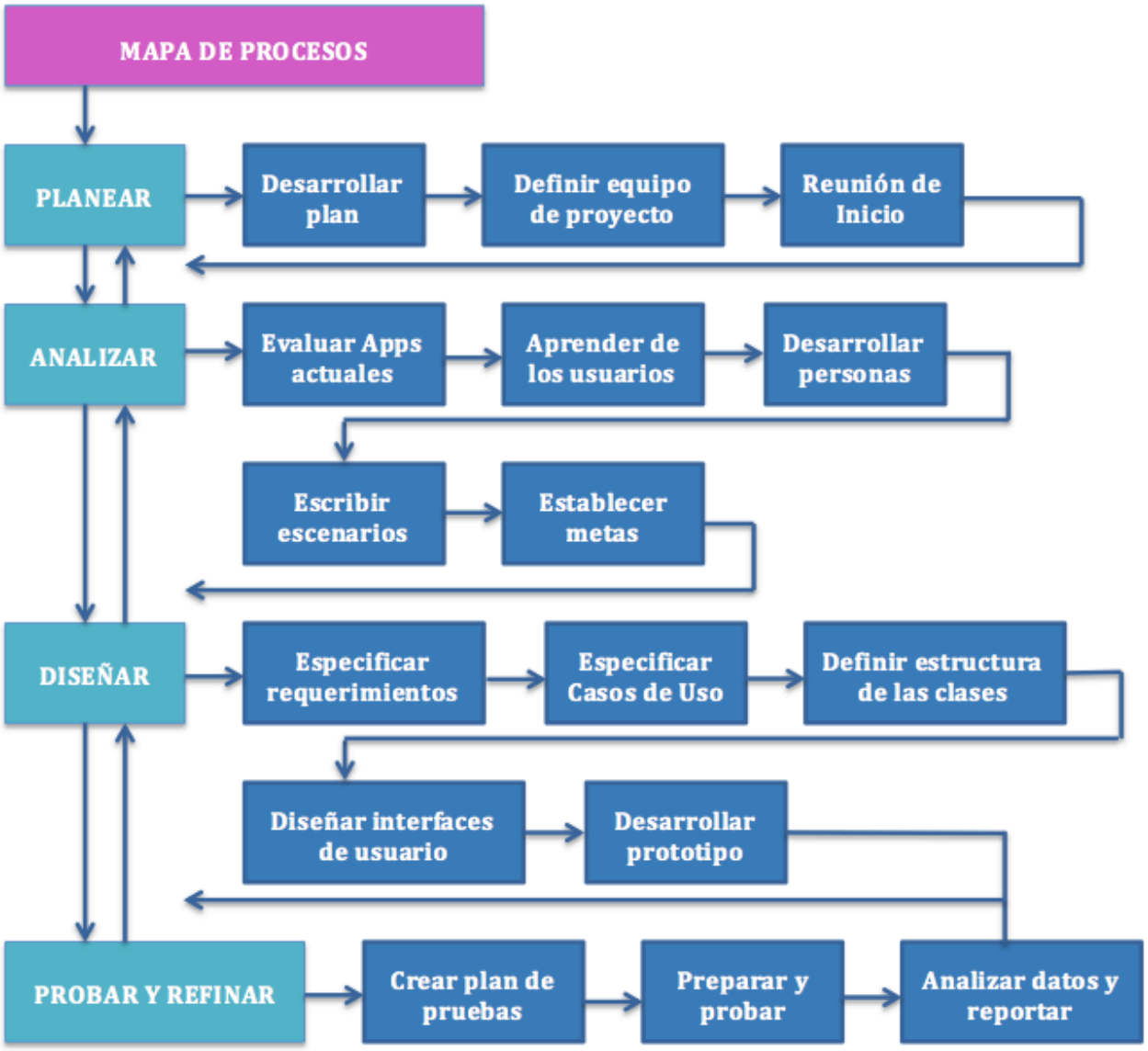
Figura 7. Descripción del Diseño Centrado en el Usuario



Fuente: KIM, Daniel. What is User Centered Design. [En línea]. daylightdesign 2015.[consultado 7 de marzo de 2015]Disponible en Internet: http://daylightdesign.com/team/

2.1.2 Mapa de Procesos. A continuación se muestra cada uno de los pasos requeridos en cada una de las cuatro fases:

Figura 8. Mapa de procesos del desarrollo de la plataforma DSC



Fuente: elaboración propia

2.1.3 Planeación. En esta fase se desarrolló el plan de trabajo, se definió el equipo de proyecto y se realizó una reunión de inicio en la cual se establecieron los puntos de partida y se dio vía libre al desarrollo de la plataforma.

2.1.4 Análisis. Inicialmente se evaluaron las actuales aplicaciones móviles existentes en diferentes tiendas como el AppStore y Google Play Store para conocer el estado actual de este tipo de herramientas disponibles a los usuarios, encontrado que hasta ese momento sólo se disponía de una aplicación móvil para dispositivos con sistema operativo iOS llamada iDynamic desarrollada por NgM Software.

Seguidamente, se procedió a conocer un poco mejor a los usuarios finales, en este caso los estudiantes de los cursos de Sistemas Dinámicos y Control Automático de la Universidad Autónoma de Occidente por medio de una encuesta (ver Anexo A) que involucró 54 estudiantes y buscaba extraer información relevante para conocer cuál sería la plataforma objetivo.

Con los resultados de la encuesta, se definió el sistema operativo objetivo y las diferentes versiones de Android que deberían soportar la aplicación móvil. Adicionalmente, se desarrolló una "persona" que representa a los posibles usuarios del sistema, con ella se escribieron los escenarios de uso y se definieron las metas a cumplir por parte de la plataforma.

2.1.5 Diseño. El diseño de la plataforma inició con la especificación de requerimientos y los casos de uso. Ya que gran parte del sistema está basado en el software para plataformas móviles desarrollado, este se estructuró de forma modular para permitir un escalamiento mucho más sencillo, por esta razón se definió la estructura de las clases y las vistas de la aplicación, estas últimas, tuvieron diversos cambios durante su desarrollo para mejorar la experiencia de usuario.

Como resultado de este proceso, se obtuvo un prototipo funcional de la plataforma.

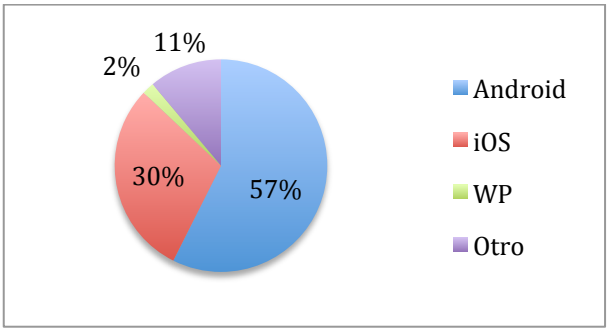
2.1.6 Pruebas. Ya con el prototipo funcionando, se creó un plan de pruebas el cual involucró el diseño de diferentes tareas y objetivos con el fin de encontrar mejoras a nivel de usabilidad. Dichas pruebas se llevaron a cabo en las instalaciones de la Universidad Autónoma de Occidente.

2.2 APRENDIENDO DE LOS USUARIOS

La encuesta empleada, buscaba encontrar información principalmente sobre los diferentes dispositivos que los usuarios poseen, el uso que se les da, principal

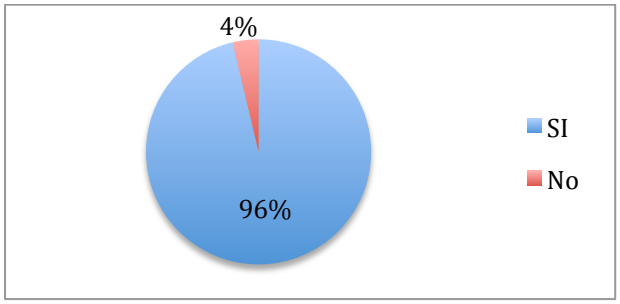
sistema operativo móvil y su aporte en el proceso de formación. Los principales resultados arrojados fueron:

Gráfica 1. Sistema operativo de los dispositivos móviles que poseen



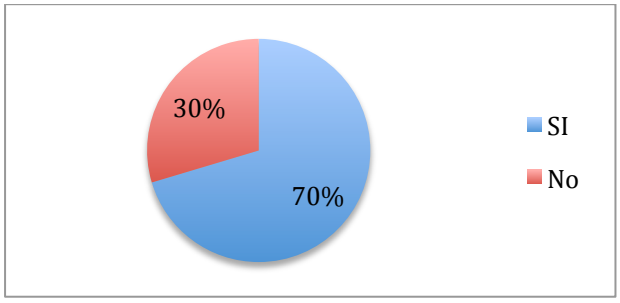
Con este resultado, fue claro que la plataforma sobre la cual se desarrollaría la aplicación sería Android. Debido a que la aplicación a desarrollar requiere de un gran procesamiento gráfico y matemático, se hizo necesario hacerlo en un entorno nativo para aprovechar al máximo las características de hardware y software del dispositivo.

Gráfica 2. Uso de software de simulación para el proceso de aprendizaje



Otro punto a resaltar, es la importancia del uso de software de simulación en los cursos de Sistemas Dinámicos ya que este ayuda a comprender mejor el funcionamiento y comportamiento de estos. Hasta el momento, el uso de este tipo de programas se hace sobre plataformas fijas como computadores de escritorio y el 96% de los encuestados hace uso de ellos.

Gráfica 3. Uso de dispositivos móviles para complementar el proceso de aprendizaje



Hoy en día, son cada vez más las aplicaciones móviles orientadas a ambientes educativos y los estudiantes son consientes de ello, por tal motivo el 70% de los encuestados las descargan y usan en los diferentes cursos de su carrera universitaria, pero hasta la fecha en la que se realizó la encuesta, no habían aplicaciones móviles para Android destinadas a complementar el proceso de aprendizaje de los cursos de Sistemas Dinámicos.

Entre otros resultados de la encuesta se obtuvo:

- El 96% consideran interesante la inclusión de dispositivos móviles como herramientas de trabajo en clase y fuera de ella.

- El 40% invierte entre 3 y 5 horas en el uso de su Smartphone y/o Tablet, mientras el 30% lo hace entre 1 y 2 horas, el 26% entre 6 y 9 horas y solo el 4% por más de 10 horas.

- El 60% conoce software de simulación diferente al ofrecido por la universidad.

- El 70% considera que la universidad cuenta con suficientes recursos de hardware y software para continuar el proceso de aprendizaje en horarios distintos a clase, mientras el 30% considera lo contrario.

- El 54% de los encuestados ha indagado en las diferentes tiendas de aplicaciones por algunas que les permitan complementar su proceso de aprendizaje.

2.3 PERSONA

Con la encuesta realizada, se procedió a implementar una 'Persona' que identificara a los estudiantes de la Universidad Autónoma de Occidente que posiblemente harían uso de la plataforma a diseñar.

2.3.1 Kevin el estudiante

Figura 9. Persona (Kevin)

Fuente: New Frontiers in Education. [En línea]. University of Rochester. 2012.[consultado 6 de marzo de 2015] Disponible en internet: http://www.rochester.edu/online-learning/symposium/index.html

Kevin es un estudiante de Ingeniería Electrónica de la Universidad Autónoma de Occidente y actualmente cursa séptimo semestre. Cuando no está en clase, Kevin pasa el tiempo con sus compañeros dentro de la universidad en espacios como la cafetería o las zonas verdes para relajarse un poco, pero en época de exámenes o proyectos, Kevin debe aprovechar su tiempo para prepararse estudiando en los laboratorios de cómputo de la universidad y si no hay disponibilidad de equipos, debe hacerlo desde su laptop con algunas limitaciones.

Durante su carrera ha aprendido el uso de diferentes programas de computadora para cada una de las asignaturas que ha cursado, desde procesadores de texto hasta programas especializados para el desarrollo de software o la simulación de sistemas, pero se ha dado cuenta que este tipo de programas en su mayoría son

licenciados e instalarlos en su computadora implicaría hacer un gasto o evadir la licencia descargándolo ilegalmente de Internet.

Desde que Kevin adquirió su smartphone, ha encontrado en este dispositivo una herramienta adicional para algunas de sus clases, pues las aplicaciones que encuentra en la tienda virtual, le permiten hacer diferentes tareas desde el mismo aparato, como la calculadora científica que descargó para su curso de matemáticas o el traductor para sus clases de inglés. Ahora Kevin no solo usa su Smartphone para estar al día con sus redes sociales, sino que saca el mayor provecho buscando herramientas que lo hagan más productivo.

Aunque la cantidad de aplicaciones que puede encontrar es bastante extensa, Kevin no ha podido encontrar aún una aplicación móvil que le ayude a prepararse para su curso de Sistemas Dinámicos, solo cuenta con el software de simulación y los equipos de laboratorio que se usan en la universidad, pero su disponibilidad no concuerda con los horarios de Kevin y esto se ha vuelto para él en un gran problema.

2.4 ESPECIFICACIÓN DE REQUERIMIENTOS

A continuación se dará una descripción de los requerimientos para el software DSC desarrollado para la Universidad Autónoma de Occidente. Se explicarán las restricciones del sistema, interfaz e interacciones con otros dispositivos. Estos requerimientos se presentaron a los docentes de los cursos de Sistemas Dinámicos y Control Automático para su aprobación y como referencia para desarrollar la primera versión del sistema.

2.4.1 Restricciones. Las limitaciones que se deben tener en cuenta en el diseño y el desarrollo del sistema son las descritas a continuación:

- La aplicación debe funcionar sobre dispositivos móviles con plataforma Android.

- La aplicación debe soportar versiones de Android a partir del API 14 (ICE CREAM SANDWICH) hasta la versión más reciente. Entre los dispositivos soportados deben encontrarse teléfonos inteligentes y tabletas.

- Los dispositivos móviles deberán contar con conexión Bluetooth

2.4.2 Requisitos Funcionales. Esta sección contiene todos los requerimientos a nivel funcional del sistema.

- **RF01:** Descarga de la aplicación móvil. (Restricción)
  La aplicación debe estar disponible para que el usuario tenga la posibilidad de descargarla desde Google Play Store de forma gratuita.

- **RF02:** Actualización de versiones de la aplicación móvil. (Restricción)
  El usuario debe revisar e instalar las nuevas versiones o actualizaciones de la aplicación, las cuales se harán desde el mismo dispositivo móvil de la misma forma que se descargó la App.

- **RF03:** Visualización de video demostrativo.
  La aplicación debe permitir visualizar un video demostrativo sobre el uso de la plataforma.

- **RF04:** Selección de sistemas a simular.
  La aplicación debe permitir al usuario escoger el sistema que desea simular en lazo abierto o cerrado, ya sea genérico de primer orden, genérico de segundo orden, nivel de un tanque, temperatura en una habitación, velocidad de un motor DC, masa-resorte-amortiguador, posición de un motor DC, nivel en tanques acoplados, péndulo invertido.

Los siguientes requisitos (RF05-RF40) son parejas de requisitos en los que se involucra parametrización y simulación de sistemas tanto en lazo abierto como en lazo cerrado. Por tal motivo, se explicará en qué consiste cada pareja de forma general ya que aplica para todos la misma descripción pero diferenciados entre sí por las variables propias de cada sistema.

Parametrización en lazo abierto: La aplicación debe permitir al usuario parametrizar un sistema en lazo abierto. Los datos a parametrizar dependerán de cada sistema.

Parametrización en lazo cerrado: La aplicación debe permitir al usuario parametrizar un sistema en lazo cerrado. Los datos a parametrizar dependerán de cada sistema junto a los datos propios del controlador.

Simulación en lazo abierto: La aplicación debe permitir visualizar una animación del sistema en lazo abierto y una gráfica que describa el comportamiento de dicho sistema. Las señales a visualizar en la gráfica serán la señal de referencia y la salida del sistema simulado.

Simulación en lazo cerrado: La aplicación debe permitir visualizar una animación de un sistema en lazo cerrado y una gráfica que describa el comportamiento de

dicho sistema. Las señales a visualizar en la gráfica serán la señal de referencia, la salida del sistema simulado, la salida del controlador y la señal de error.

- **RF05:** Parametrización de un sistema genérico de primer orden en lazo abierto.
- **RF06:** Simulación de un sistema genérico de primer orden en lazo abierto.
- **RF07**: Parametrización de un sistema genérico de primer orden en lazo cerrado.
- **RF08:** Simulación de un sistema genérico de primer orden en lazo cerrado.
- **RF09:** Parametrización de un sistema genérico de segundo orden en lazo abierto.
- **RF10:** Simulación de un sistema genérico de segundo orden en lazo abierto.
- **RF11:** Parametrización de un sistema genérico de segundo orden en lazo cerrado.
- **RF12:** Simulación de un sistema genérico de segundo orden en lazo cerrado.
- **RF13:** Parametrización de un sistema de nivel de un tanque en lazo abierto.
- **RF14:** Simulación de un sistema de nivel de un tanque en lazo abierto.
- **RF15:** Parametrización de un sistema de nivel de un tanque en lazo cerrado.
- **RF16:** Simulación de un sistema de nivel de un tanque en lazo cerrado.
- **RF17:** Parametrización de un sistema de temperatura en una habitación en lazo abierto.
- **RF18:** Simulación de un sistema de temperatura en una habitación en lazo abierto.
- **RF19:** Parametrización de un sistema de temperatura en una habitación en lazo cerrado.
- **RF20:** Simulación de un sistema de temperatura en una habitación en lazo cerrado.
- **RF21:** Parametrización de un sistema de velocidad de un motor de DC en lazo abierto.

- **RF22:** Simulación de un sistema de velocidad de un motor de DC en lazo abierto.
- **RF23:** Parametrización de un sistema de velocidad de un motor de DC en lazo cerrado.
- **RF24:** Simulación de un sistema de velocidad de un motor de DC en lazo cerrado.
- **RF25:** Parametrización de un sistema de amortiguación de un auto en lazo abierto.
- **RF26:** Simulación de un sistema de amortiguación de un auto en lazo abierto.
- **RF27:** Parametrización de un sistema de amortiguación de un auto en lazo cerrado.
- **RF28:** Simulación de un sistema de amortiguación de un auto en lazo cerrado.
- **RF29:** Parametrización de un sistema de posición de un motor DC en lazo abierto.
- **RF30:** Simulación de un sistema de posición de un motor DC en lazo abierto.
- **RF31:** Parametrización de un sistema de posición de un motor DC en lazo cerrado.
- **RF32:** Simulación de un sistema de posición de un motor DC en lazo cerrado.
- **RF33:** Parametrización de un sistema de nivel en tanques acoplados en lazo abierto.
- **RF34:** Simulación de un sistema de nivel en tanques acoplados en lazo abierto.
- **RF35:** Parametrización de un sistema de nivel en tanques acoplados en lazo cerrado.
- **RF36**: Simulación de un sistema de nivel en tanques acoplados en lazo cerrado.
- **RF37:** Parametrización de un sistema de péndulo invertido en lazo abierto.
- **RF38:** Simulación de un sistema de péndulo invertido en lazo abierto.
- **RF39:** Parametrización de un sistema de péndulo invertido en lazo cerrado.

- **RF40:** Simulación de un sistema de péndulo invertido en lazo cerrado.

- **RF41:** Conmutar animación y gráfica en el control de sistemas**.**
  El sistema debe permitir conmutar la animación y gráfica en el control de todos los sistemas. La simulación del sistema deberá iniciar dando más protagonismo a la animación sobre la gráfica.

- **RF42:** Implementar control externo de sistemas.
  El sistema debe permitir separar el controlador de la planta, actuando cada uno en un dispositivo móvil diferente y comunicándose entre sí por medio de una conexión Bluetooth.

- **RF43:** Activar Bluetooth del dispositivo.
  El sistema debe permitir activar el Bluetooth del dispositivo si este se encuentra apagado.

- **RF44:** Búsqueda de dispositivos cercanos para conexión Bluetooth en el control externo de sistemas.
  El sistema debe permitir a la aplicación instalada en el dispositivo que inicia el control externo, asumir el rol de controlador y buscar un dispositivo cercano vía Bluetooth para conectarse y que asuma el rol de planta.

- **RF45:** Parametrización del sistema en el Control Externo de Sistemas.
  El sistema debe permitir al usuario parametrizar el controlador y la planta de forma individual en cada uno de los dispositivos donde estos se encuentran.

- **RF46:** Simulación del controlador en el Control Externo de Sistemas.
  El sistema debe permitir visualizar una gráfica que describa el comportamiento del controlador en el dispositivo que asumió este rol.

- **RF47:** Simulación de la planta en el Control Externo de Sistemas.
  El sistema debe permitir visualizar una gráfica que describa el comportamiento de la planta al mismo tiempo que ejecutar una animación dependiente de dicho comportamiento en el dispositivo que asumió este rol.

- **RF48:** Perturbar el sistema por medio de sensores.
  El sistema debe permitir el uso de sensores del dispositivo móvil para perturbar los diferentes sistemas.

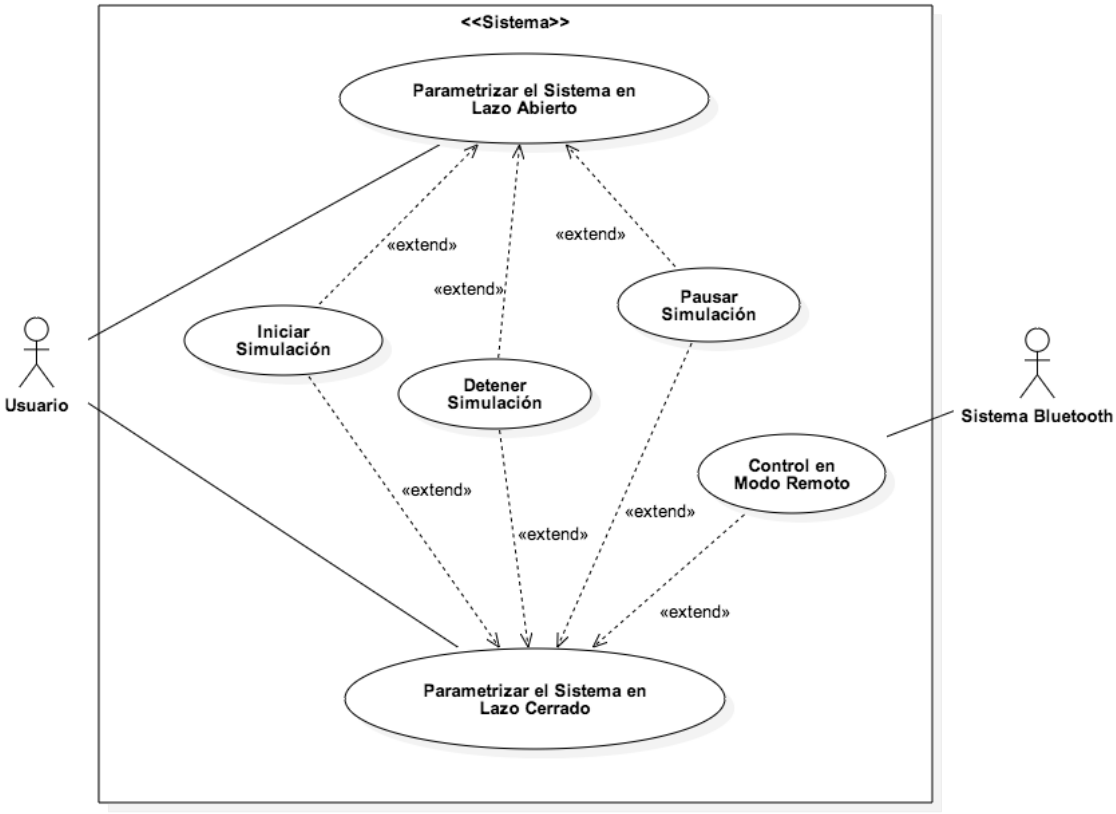
- **RF49:** Acceso a información personal del Usuario en el Dispositivo Móvil. (Restricción)

El sistema no debe permitir el acceso a la información personal del usuario almacenada en el dispositivo móvil, tales como imágenes, videos, archivos personales.

2.5 CASOS DE USO

La implementación de casos de uso da una perspectiva un poco más clara de cómo el usuario interactúa con las diferentes opciones que ofrece la plataforma, la siguiente figura muestra un diagrama general de casos de uso que aplica para cada uno de los sistemas dinámicos que conforman la plataforma.

Figura 10. Diagrama general de Casos de Uso



Fuente: elaboración propia

En el diagrama se visualizan dos actores, el usuario y el sistema Bluetooth del dispositivo, donde este último solo actúa cuando se desea trabajar con el control a distancia de la planta.

El usuario por su parte, actúa parametrizando los sistemas dinámicos tanto en lazo abierto como en lazo cerrado, y una vez ingresados los parámetros, puede iniciar una simulación, pausarla o detenerla, así como decidir cuándo se hará control en modo remoto solo si se ha escogido un sistema dinámico en lazo cerrado.

Los casos de uso CU01 hasta CU18 hacen referencia a la parametrización de sistemas dinámicos, por lo tanto se explicará en qué consiste de forma general ya que aplica de igual forma para todos los sistemas.

- Parametrizar un sistema en lazo abierto: El estudiante realiza la parametrización de un sistema en lazo abierto cuyos parámetros dependerán del sistema dinámico escogido.
    - Flujo de eventos:
        - El caso de uso inicia cuando el estudiante selecciona la opción lazo abierto de un Sistema Dinámico.
        - La aplicación muestra la pantalla para ingresar los parámetros.
        - El estudiante ingresa los parámetros de la planta.
    - Post-Condiciones: El estudiante podrá iniciar la simulación (CU19)

- Parametrizar un sistema en lazo cerrado: El estudiante realiza la parametrización de un sistema en lazo cerrado cuyos parámetros dependerán del sistema dinámico escogido junto a los parámetros del controlador.
    - Flujo de eventos:
        - El caso de uso inicia cuando el estudiante selecciona la opción lazo cerrado de un Sistema Dinámico.
        - La aplicación muestra la pantalla para ingresar los parámetros
        - El estudiante ingresa los parámetros de la planta y el controlador.
    - Post-Condiciones: El estudiante podrá iniciar la simulación (CU19)

- **CU01:** Parametrizar Sistema Genérico de primer orden en lazo abierto.

- **CU02:** Parametrizar Sistema Genérico de primer orden en lazo cerrado.

- **CU03:** Parametrizar Sistema Genérico de segundo orden en lazo abierto.

- **CU04:** Parametrizar Sistema Genérico de segundo orden en lazo cerrado.

- **CU05:** Parametrizar Sistema de Nivel de un tanque en lazo abierto.

- **CU06:** Parametrizar Sistema de Nivel de un tanque en lazo cerrado.

- **CU07:** Parametrizar Sistema de temperatura en una habitación en lazo abierto.

- **CU08:** Parametrizar Sistema de temperatura en una habitación en lazo cerrado.
- **CU09:** Parametrizar Sistema de Velocidad de un Motor DC en lazo abierto.
- **CU10:** Parametrizar Sistema de Velocidad de un Motor DC en lazo cerrado.
- **CU11:** Parametrizar Sistema de Amortiguación de un Auto en lazo abierto.
- **CU12:** Parametrizar Sistema de Amortiguación de un Auto en lazo cerrado.
- **CU13:** Parametrizar Sistema de Posición de un Motor de DC en lazo abierto.
- **CU14:** Parametrizar Sistema de Posición de un Motor de DC en lazo cerrado.
- **CU15:** Parametrizar Sistema de Nivel en Tanques Acoplados en lazo abierto.
- **CU16:** Parametrizar Sistema de Nivel en Tanques Acoplados en lazo cerrado.
- **CU17:** Parametrizar Sistema de Péndulo Invertido en lazo abierto.
- **CU18:** Parametrizar Sistema de Péndulo Invertido en lazo abierto.
- **CU19:** Iniciar Simulación de Sistema.

  El estudiante inicia la simulación de un sistema una vez los parámetros de este hayan sido introducidos.
  - Pre-Condiciones: El estudiante debe haber ingresado todos los parámetros del sistema seleccionado.
  - Flujo de eventos:
    - El caso de uso inicia cuando el estudiante ha terminado de parametrizar el sistema escogido y pulsa el botón de inicio de simulación
    - La aplicación muestra una animación del sistema seleccionado de acuerdo a los valores entregados por el método numérico aplicado al sistema escogido, al tiempo que grafica este comportamiento
  - Post-Condiciones: El estudiante podrá pausar o detener la simulación (CU20) / (CU21)

- **CU20:** Pausar Simulación de Sistema: El estudiante pausa la simulación de un sistema.
  - Pre-Condiciones: El estudiante debe haber iniciado la simulación de algún sistema. (CU19).
  - Flujo de eventos:

    - El caso de uso inicia cuando el estudiante ha iniciado la simulación del sistema y pulsa el botón de Pausado.
    - La aplicación pausa la animación del sistema seleccionado y la gráfica de este de forma temporal hasta que sea reanudada o detenida.
  - Post-Condiciones: El estudiante podrá reanudar o detener la simulación. (CU19).

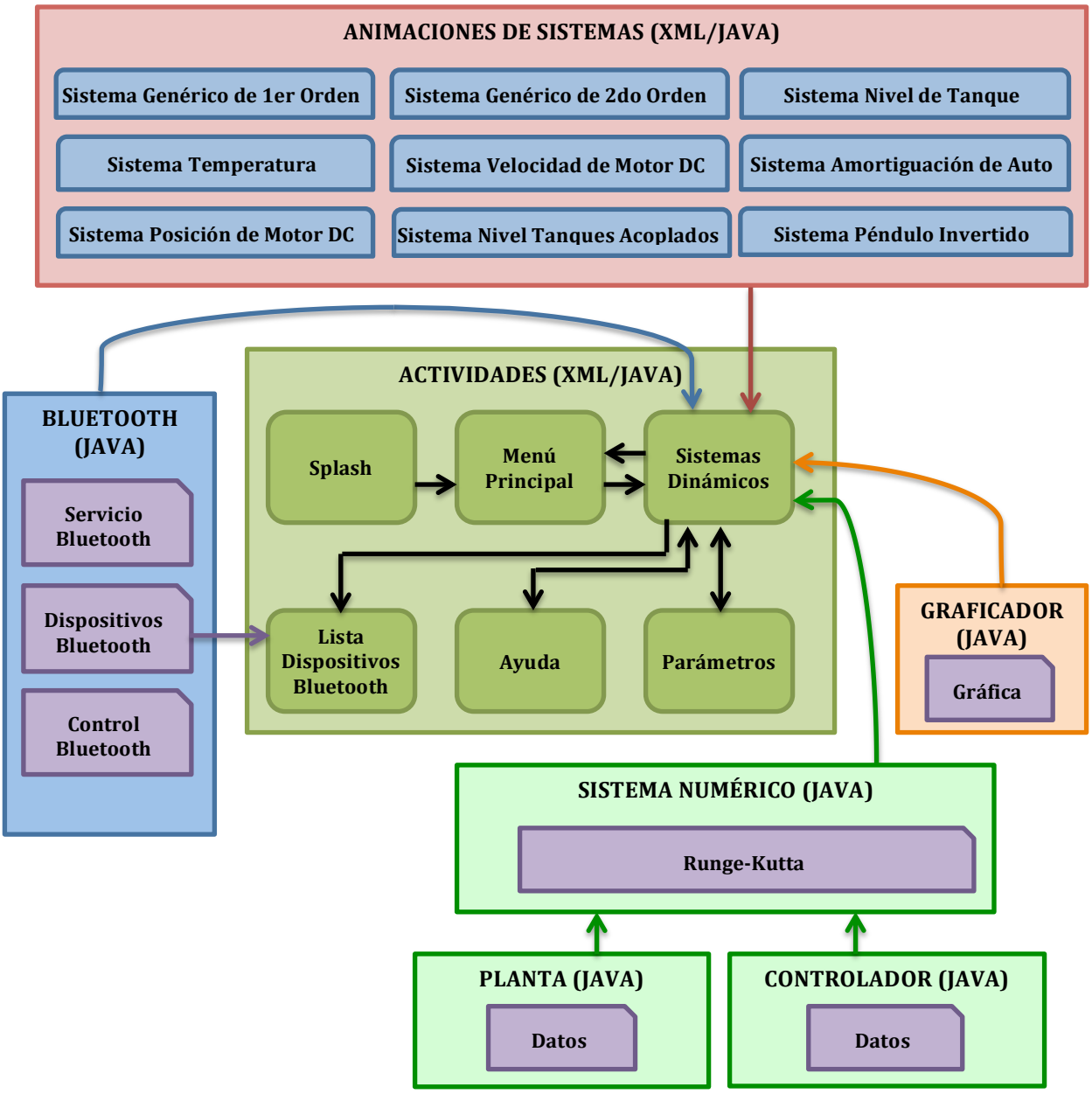
- **CU21:** Detener Simulación de Sistema: El estudiante detiene la simulación de un sistema.
  - Pre-Condiciones: El estudiante debe haber iniciado/pausado la simulación de algún sistema. (CU19/CU20).
  - Flujo de eventos:
    - El caso de uso inicia cuando el estudiante ha iniciado o pausado la simulación del sistema y pulsa el botón de Detener.
    - La aplicación detiene la animación del sistema seleccionado y elimina la gráfica de este, reiniciando todos los valores del sistema a sus condiciones iniciales.
  - Post-Condiciones: El estudiante podrá iniciar nuevamente la simulación. (CU19).

- **CU22:** Control Remoto de Sistema: El estudiante controla el sistema desde un dispositivo remoto vía Bluetooth.
  - Pre-Condiciones: El estudiante debe haber escogido la opción “lazo cerrado” de cualquier sistema.
  - Flujo de eventos:
    - El caso de uso inicia cuando el estudiante ha escogido un sistema en lazo cerrado y pulsa el botón de control remoto Bluetooth.
    - La aplicación activa el sistema Bluetooth y lo pone en modo de búsqueda de otros dispositivos.
    - El sistema Bluetooth busca otros dispositivos Bluetooth que se encuentren en modo visible.
    - El estudiante selecciona el dispositivo con el cual desea implementar el sistema en lazo cerrado.
    - El sistema Bluetooth se conecta al otro dispositivo.
    - La aplicación toma el rol de controlador y el otro dispositivo el rol de planta.
  - Post-Condiciones: El estudiante podrá iniciar la simulación. (CU19).

2.6 DIAGRAMA DE BLOQUES DEL SOFTWARE

El diseño de la plataforma se dividió en módulos para facilitar su futura ampliación con sistemas dinámicos más complejos.

Figura 11. Diagrama de bloques de la plataforma DSC

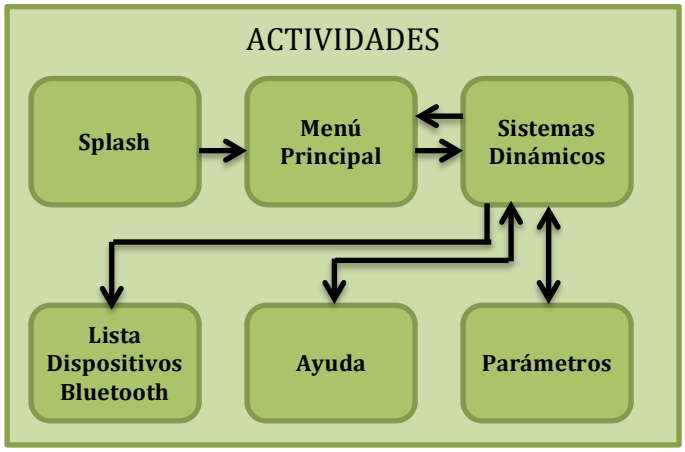


Fuente: elaboración propia

En el diagrama de bloques se puede observar los diferentes módulos que componen la aplicación móvil y la interacción entre ellos. El bloque central (ACTIVIDADES) representa todas las pantallas de la aplicación desarrolladas en lenguaje XML para la interfaz gráfica y JAVA para la funcionalidad.

Inicia con un *Splash* y seguidamente mostrando el menú principal. En el menú principal se escoge uno de los sistemas dinámicos y es aquí donde se centra el enfoque principal del software. Desde esta pantalla el usuario puede dirigirse a la parametrización del sistema, ver la ayuda o buscar dispositivos Bluetooth para el control remoto.

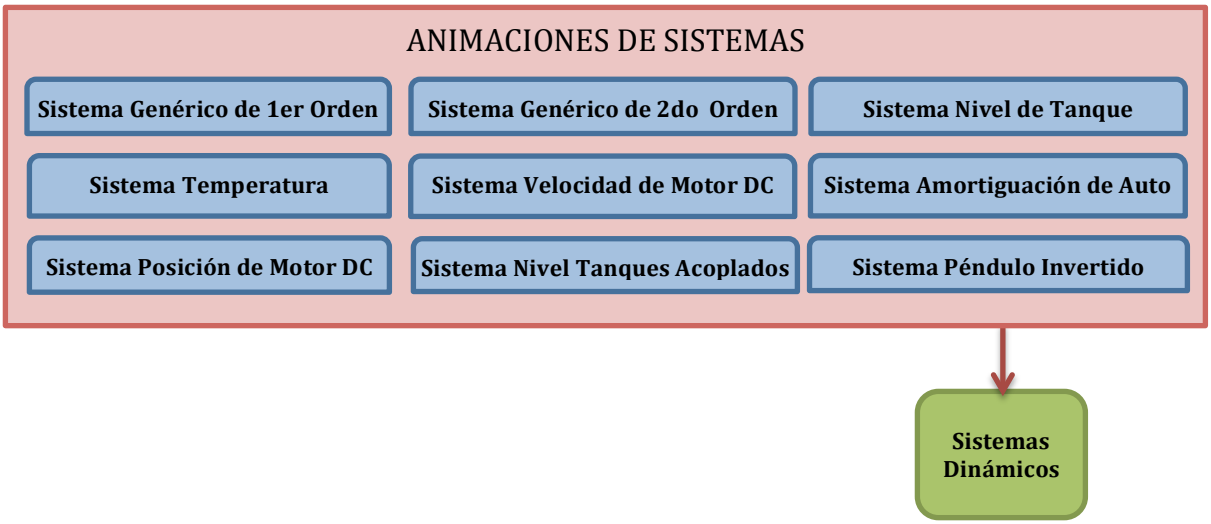
Figura 12. Bloque de Actividades



Fuente: elaboración propia

El bloque ANIMACIONES DE SISTEMAS contiene las diferentes animaciones (desarrolladas en lenguaje XML y JAVA) de cada posible sistema para ser cargado en la actividad *Sistemas Dinámicos* del bloque ACTIVIDADES y permitir las diferentes simulaciones.

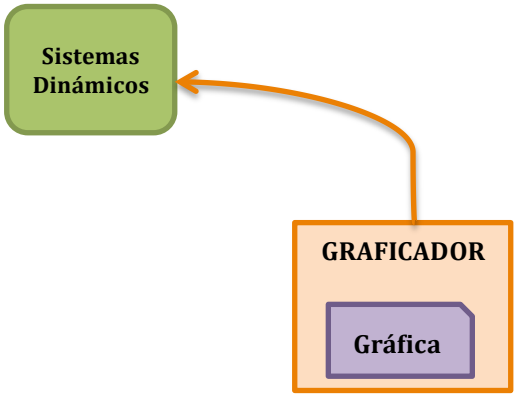
Figura 13. Bloque Animaciones de Sistemas



Fuente: elaboración propia

El bloque GRAFICADOR contiene una clase JAVA que se encarga de configurar y adicionar los datos de la gráfica que muestra las señales correspondientes al comportamiento del sistema entre las que se incluyen la señal de salida del controlador y la planta, el error y la referencia. Esta información es graficada en la actividad *Sistemas Dinámicos* del bloque ACTIVIDADES.

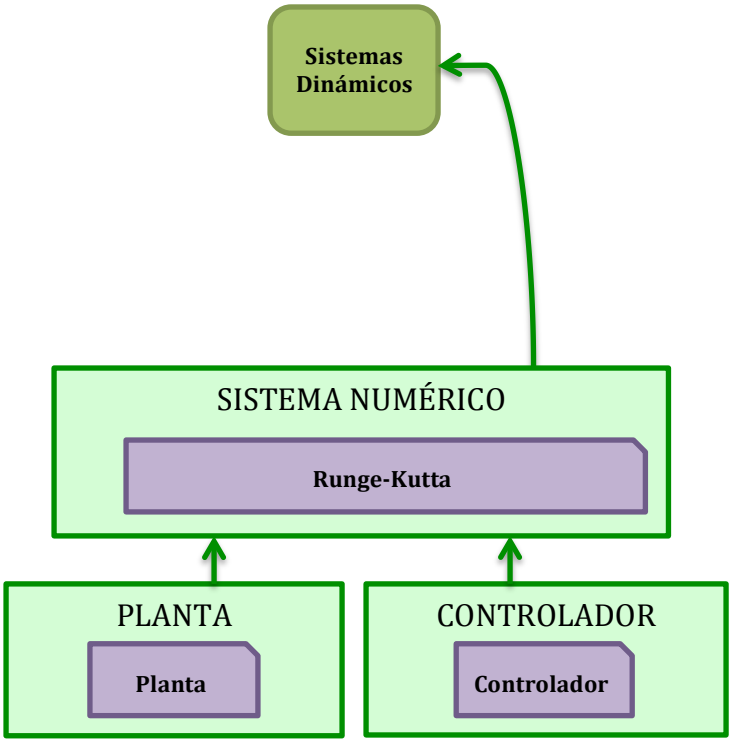
Figura 14. Bloque Graficador



Fuente: elaboración propia

El bloque SISTEMA NUMÉRICO contiene una clase encargada de ejecutar el RUNGE-KUTTA de 4to orden al controlador y la planta del sistema cuyos datos provienen de dos bloques llamados igualmente CONTROLADOR y PLANTA. Esta información es recibida por la actividad *Sistemas Dinámicos* quien la usa para el cálculo de otras variables y generar la simulación del sistema.

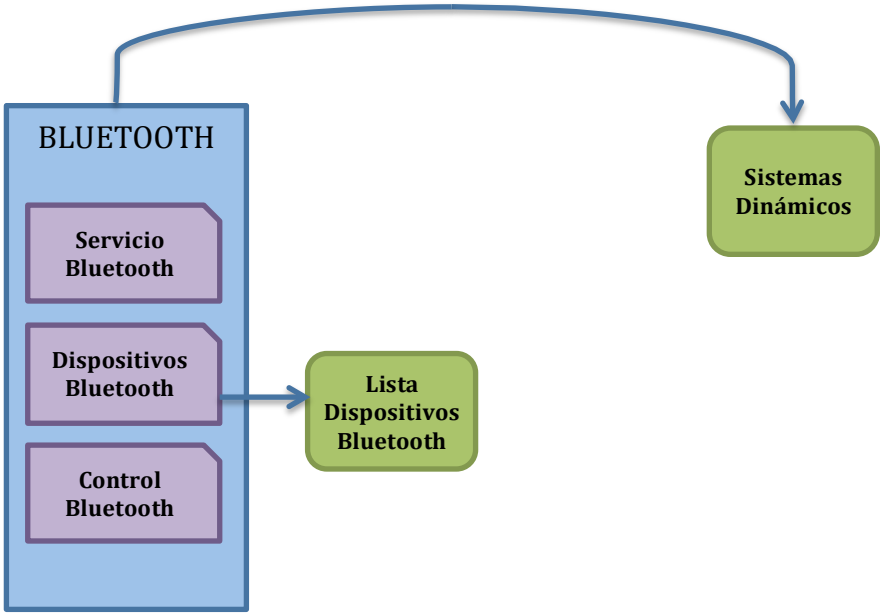
Figura 15. Bloque Sistema Numérico-Planta-Controlador



Fuente: elaboración propia

Finalmente el bloque BLUETOOTH contiene 3 clases JAVA encargadas de gestionar la búsqueda, conexión y comunicación entre dispositivos para el control remoto por medio de este protocolo inalámbrico. La búsqueda de dispositivos hace parte de la actividad *Lista de dispositivos Bluetooth* del bloque ACTIVIDADES y los datos enviados y recibidos entre dispositivos interactúan con la actividad *Sistemas Dinámicos* del bloque ACTIVIDADES.

Figura 16. Bloque Bluetooth

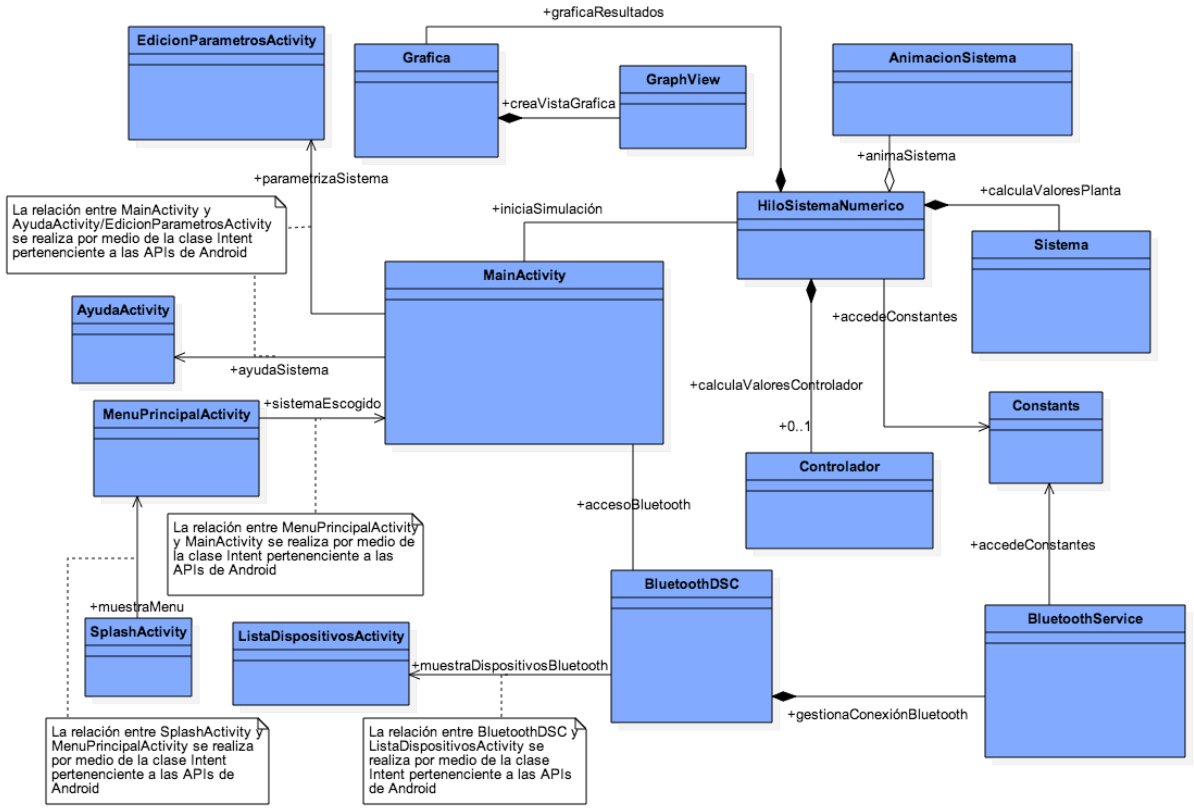


Fuente: elaboración propia

2.7 DIAGRAMA DE CLASES

El diagrama de clases describe la estructura del sistema (software) mostrando las clases que lo componen y la relación que existe entre ellas. A continuación se muestra en la figura 17 el diagrama de clases sin sus atributos ni métodos, por el contrario, se muestra de forma general las diferentes clases relacionadas entre sí y posteriormente se procederá a explicar con más detalle cada una de ellas.

Figura 17. Diagrama de clases del sistema DSC



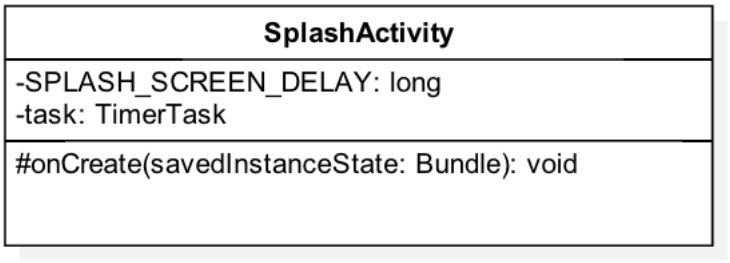
Fuente: elaboración propia

En el diagrama de clases se puede observar que el sistema está compuesto por 15 clases, pero una de estas no es propia, sino que debió incluirse para mostrar la relación entre la clase Gráfica y la vista que permite graficar y visualizar las señales del sistema, en este caso particular se está hablando de la clase GraphView la cual hace parte de la librería que lleva el mismo nombre.

A continuación se explica cada una de estas clases con más detalle.

2.7.1 Clase SplashActivity. Esta clase representa la Actividad que se muestra al iniciar la aplicación por un tiempo de 2 segundos antes de iniciar la Actividad que muestra el menú principal al usuario. Básicamente es la presentación que se da al usuario antes de que pueda hacer uso del software.

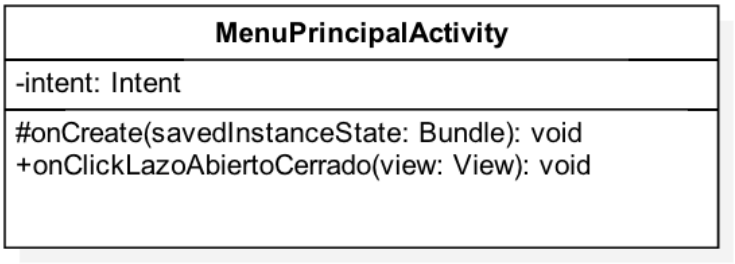
Figura 18. Clase SplashActivity



F

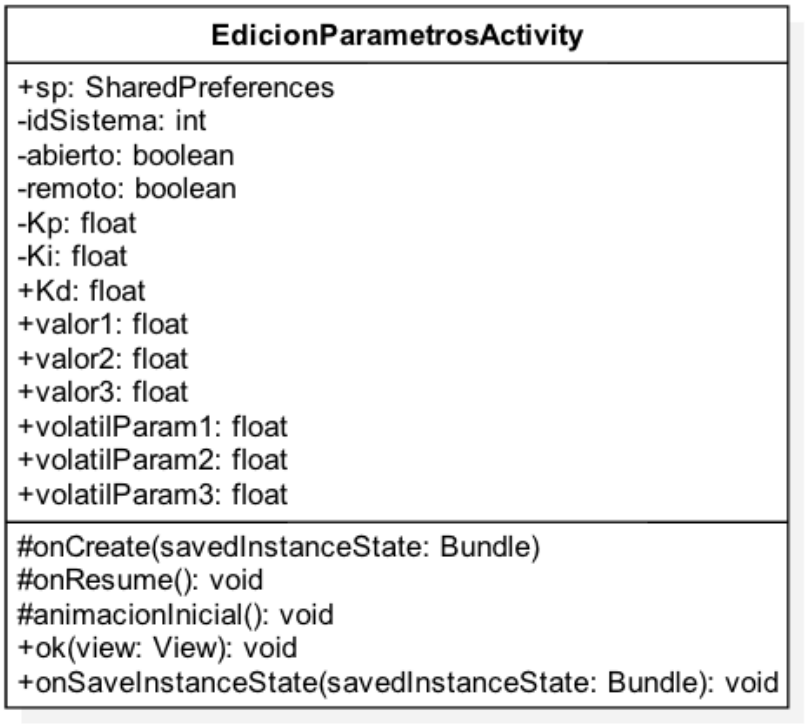
2.7.2 Clase MenuPrincipalActivity. Esta clase representa una Actividad del sistema, en este caso el Menú Principal desde el cual el usuario podrá escoger un sistema y su modo de funcionamiento ya sea en lazo abierto o cerrado.

Figura 19. Clase MenuPrincipalActivity



2.7.3 Clase EdicionParametrosActivity. Esta clase perteneciente a una Actividad, permite que el usuario pueda parametrizar el sistema a simular ingresando los valores de las constantes del sistema y los valores del controlador en caso de ser un sistema en lazo cerrado.

Figura 20. Clase EdicionParametrosActivity



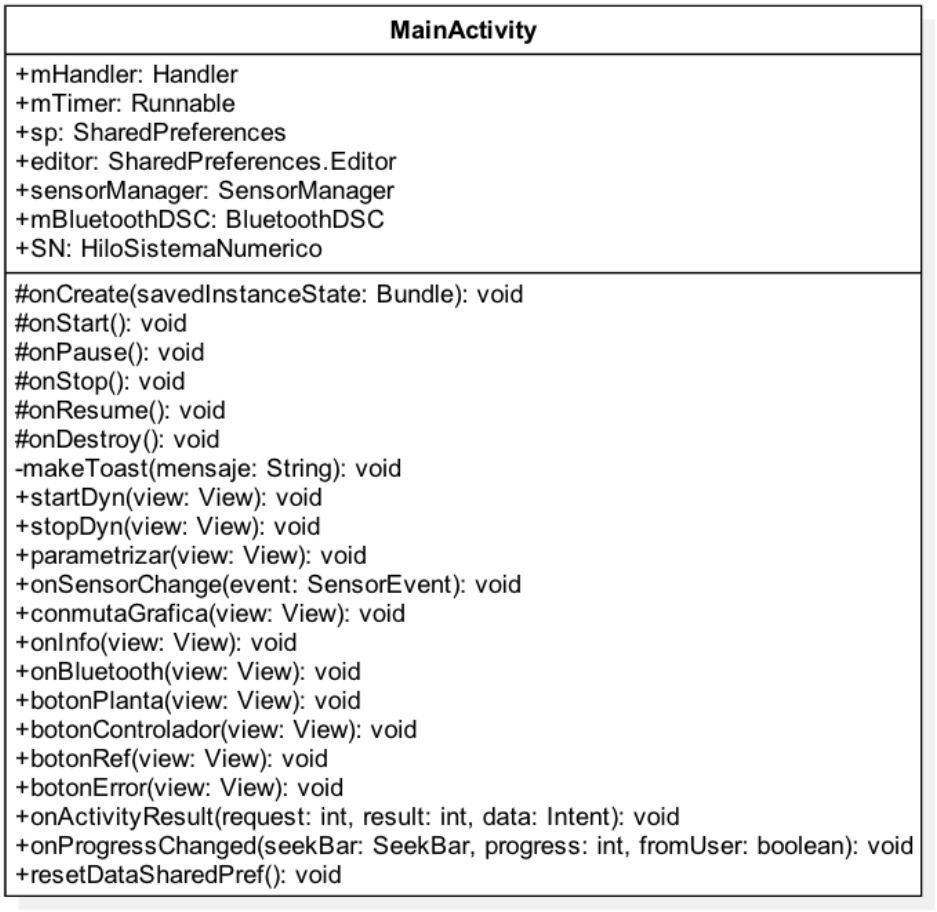
2.7.4 Clase MainActivity. Esta clase hace referencia a la Actividad principal del sistema, en ella se muestra el sistema a simular, el acceso a control externo por Bluetooth, los controles para iniciar/detener la animación y la gráfica de comportamiento del sistema. Esta clase tiene conexión con otras clases que hacen posible la simulación de todo el sistema.

La figura 21 no muestra el total de atributos de la clase MainActivity debido a la gran cantidad que contiene, por esta razón a continuación se mencionan los atributos restantes:

- **public float tao, Kp** , **Ki** , **Kd**
- **private float val1**, **val2**, **val3**, **val4**, **val5**
- **float offset**
- **public boolean proximidad**
- **public boolean botonPlanta**, **botonControlador**, **botonError**, **botonRef**
- **public** TextView **txtVwSetPoint, txtVwK**, **txtVwZita**, **txtVwWn**
- **public** ImageView **bPlanta**, **bError**, **bControlador**, **bRef, btn_start**, **btn_stop**
- **public** ImageView **btn_BT**
- **public** List<Sensor> **sensores**
- **private static final int *REQUEST_CONNECT_DEVICE_INSECURE***
- **public** SeekBar **skBrSetPoint**
- **double maxRangeProximitySensor**

- **private int idSistema**
- **public boolean remoto**, **inicio=true**, **abierto**
- **private static final** String ***TAG***
- **private static final boolean *D***
- **private static final int *REQUEST_ENABLE_BT,***
  REQUEST_CONNECT_DEVICE_SECURE

Figura 21. Clase MainActivity

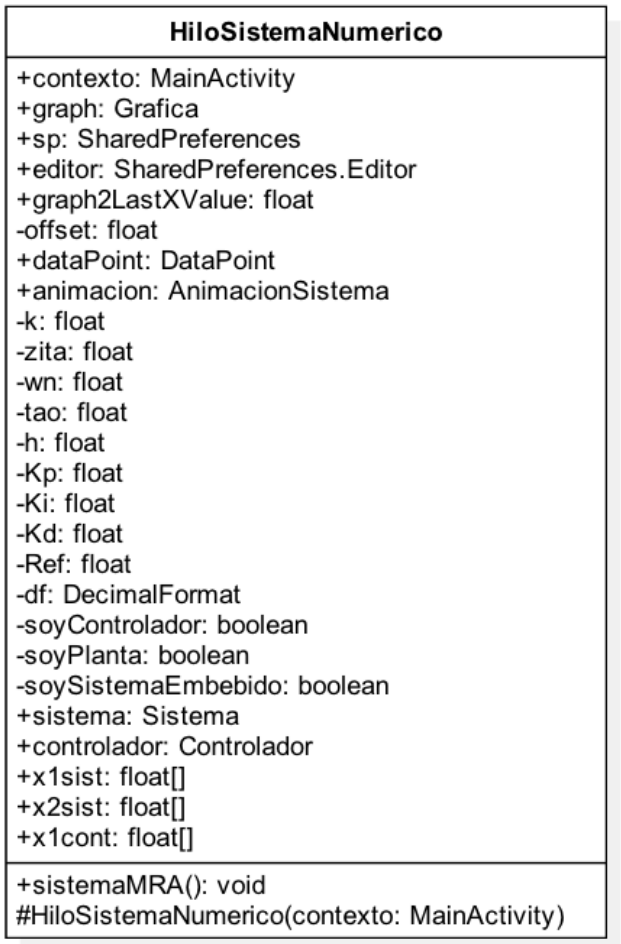


2.7.5 Clase HiloSistemaNumerico. Esta clase se encarga de ejecutar el hilo encargado de hacer el procesamiento matemático, la animación y graficar las señales del sistema. La razón de hacer esta gran cantidad de procesos en un hilo fue con el propósito de no bloquear el hilo principal encargado de manejar la interfaz de usuario.

La figura 22 no muestra el total de atributos de la clase HiloSistemaNumerico debido a la gran cantidad que contiene, por esta razón a continuación se mencionan los atributos restantes:

- **int i, j**
- **float ciAs**, **ciBs**, **ciAc, max, min, tmax, Uc, Error, x2B, salidaPlanta**
- **float**[] **t**
- **boolean first, limit**
- DecimalFormat **df**

Figura 22. Clase HiloSistemaNumerico

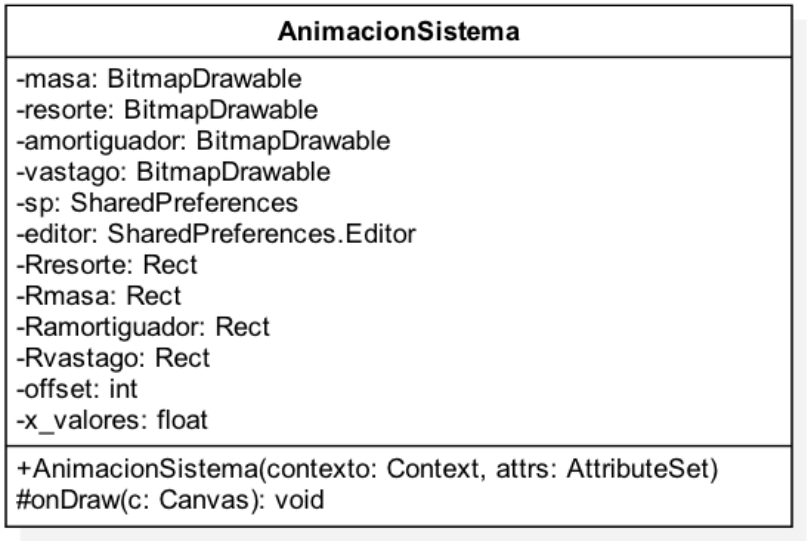


2.7.6 Clase AnimacionSistema. La clase AnimacionSistema se encarga de redibujar la animación a partir de los valores calculados en la clase HiloSistemaNumerico. Para el prototipo desarrollado solo se cuenta con esta clase, pero para la plataforma completa deberá existir una clase de este tipo por cada sistema que se vaya a implementar.

La figura 23 no muestra el total de atributos de la clase AnimacionSistema debido a la gran cantidad que contiene, por esta razón a continuación se mencionan los atributos restantes:

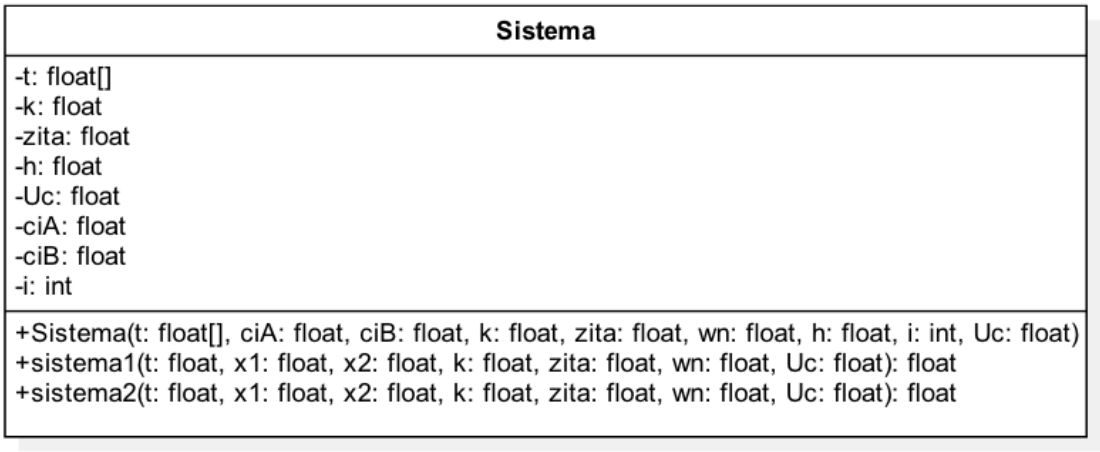
- **int getHeightMedio**
- **double amort_0_35**, **heightAmortMedio**, **heightVastagoMedio**,**heightResorteMedio**
- **double heightAmort_3_8**, **heightVastago_3_8**, **heightResorte_3_8**, **heightMasa_3_8**
- **int iniXResorte**, **finXResorte**, **iniYResorte**, **finYResorte**
- **int iniXCamisa**, **finXCamisa**, **iniYCamisa**, **finYCamisa**
- **int iniXVastago**, **finXVastago**, **iniYVastago**, **finYVastago**
- **int iniXMasa**, **finXMasa**, **iniYMasa**, **finYMasa**

Figura 23. Clase AnimacionSistema



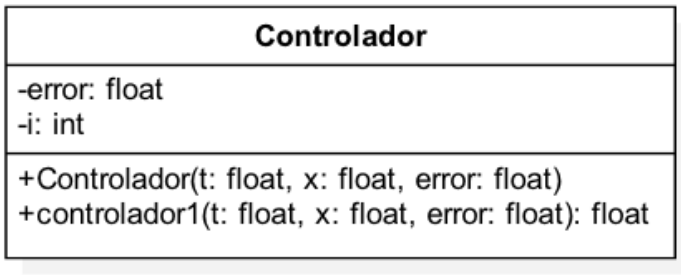
2.7.7 Clase Sistema. Esta clase cumple la función de entregar a la clase HiloSistemaNumerico los valores correspondientes a la planta para que pueda aplicarse el método numérico y así calcular la salida de esta.

Figura 24. Clase Sistema



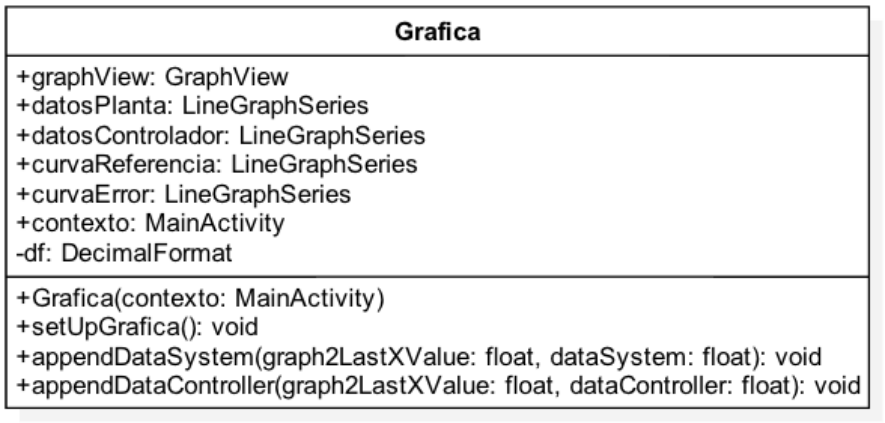
2.7.8 Clase Controlador. Esta clase cumple la función de entregar a la clase HiloSistemaNumerico los valores correspondientes al controlador para que pueda aplicarse el método numérico y así calcular la salida de este.

Figura 25. Clase Controlador



2.7.9 Clase Grafica. La clase Grafica se encarga de manejar lo relacionado con la vista correspondiente a la gráfica que muestra las diferentes señales del sistema como por ejemplo la señal de referencia, la salida del controlador, la salida de la planta y la señal de error.

Figura 26. Clase Grafica

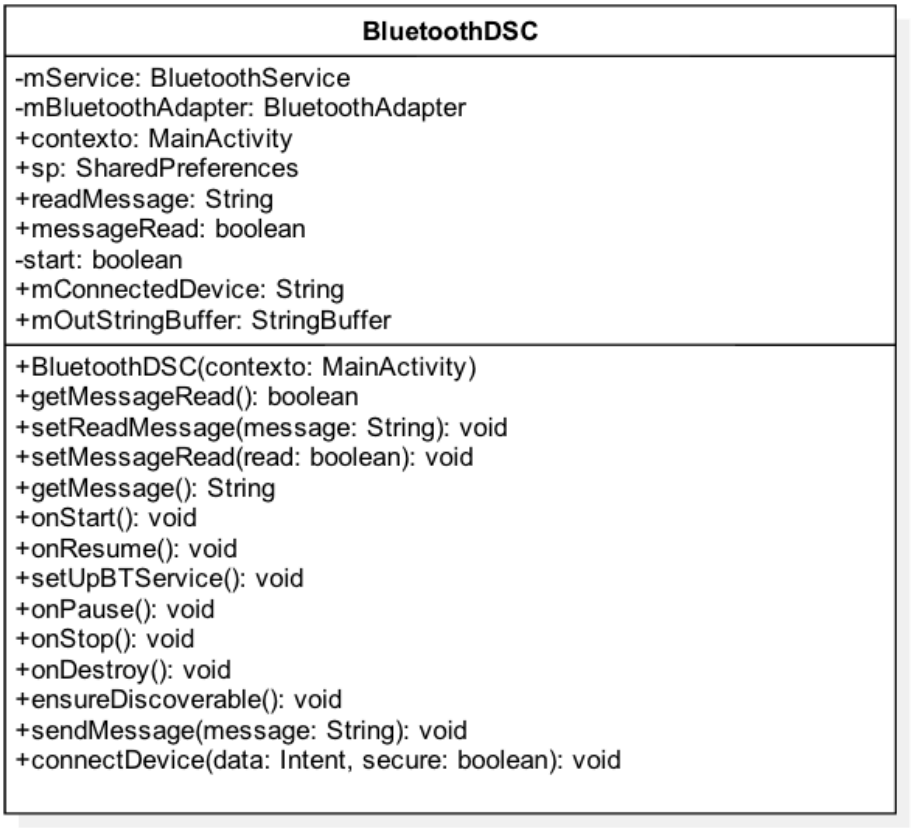


2.7.10 Clase BluetoothDSC. BluetoothDSC es una clase que permite la conexión con otros dispositivos Bluetooth para la opción de control externo del sistema a simular. Esta clase hace uso de otras dos clases que se encargan de tareas individuales para lograr el objetivo de mostrar dispositivos Bluetooth y conectarse a ellos.

La figura 27 no muestra el total de atributos de la clase BluetoothDSC debido a la gran cantidad que contiene, por esta razón a continuación se mencionan los atributos restantes:

- **public static final int *MESSAGE_STATE_CHANGE***
- **public static final int *MESSAGE_READ***
- **public static final int *MESSAGE_WRITE***
- **public static final int *MESSAGE_DEVICE_NAME, MESSAGE_TOAST***
- **public static final** String ***DEVICE_NAME***
- **public static final** String ***TOAST***
- **private static final int *REQUEST_ENABLE_BT***

Figura 27. Clase BluetoothDSC

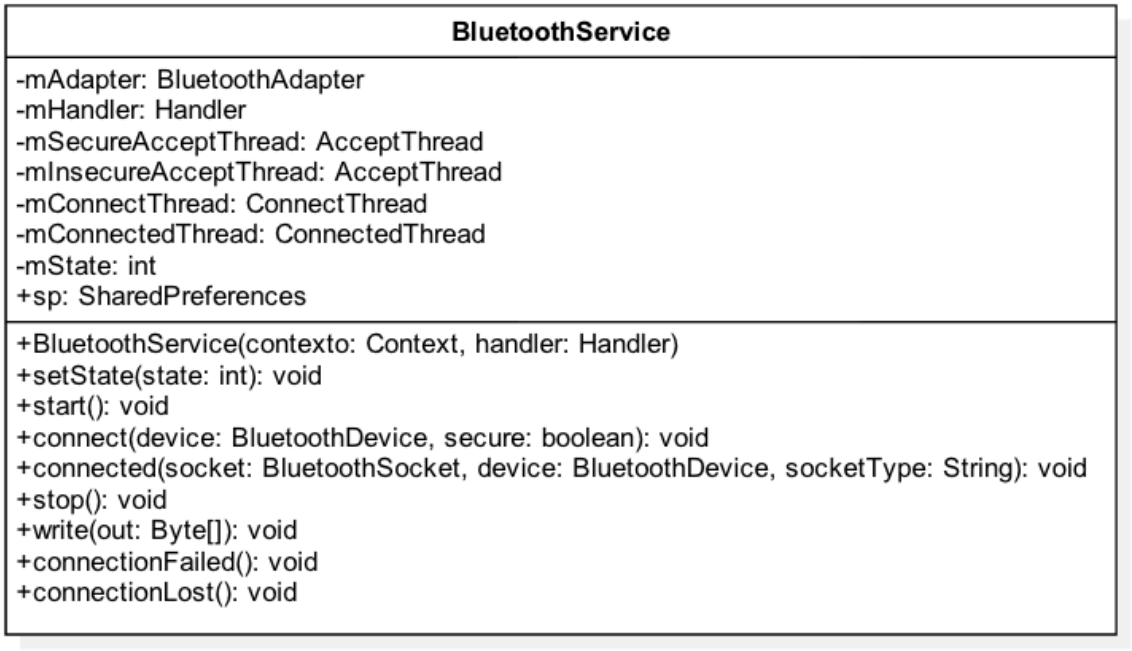


2.7.11 Clase BluetoothService. Esta clase se encarga de gestionar la conexión Bluetooth con otros dispositivos. Contiene un hilo que escucha conexiones entrantes, un hilo para conectar con algún dispositivo y un hilo para la transmisión de datos una vez se establezca una conexión.

La figura 28 no muestra el total de atributos de la clase BluetoothService debido a la gran cantidad que contiene, por esta razón a continuación se mencionan los atributos restantes:

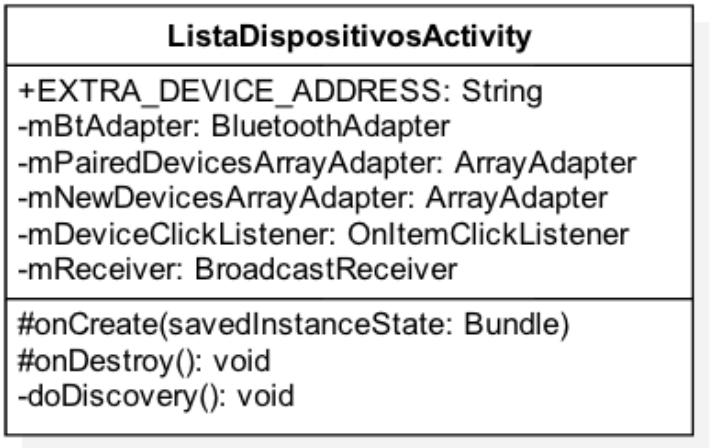
- **private static final** String ***NAME_SECURE***
- **private static final** String ***NAME_INSECURE***
- **private static final** UUID ***MY_UUID_SECURE***
- **private static final** UUID ***MY_UUID_INSECURE***
- **public static final int** ***STATE_NONE***
- **public static final int** ***STATE_LISTEN***
- **public static final int** ***STATE_CONNECTING***
- **public static final int** ***STATE_CONNECTED***

Figura 28. Clase BluetoothService



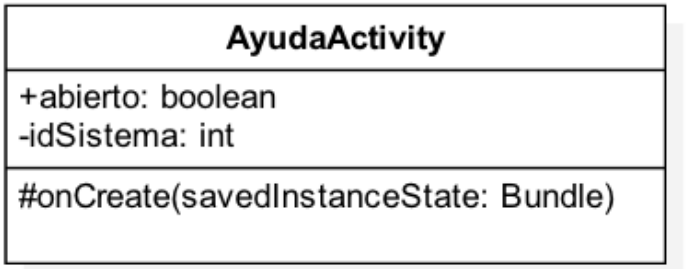
2.7.12 Clase ListaDispositivosActivity. Esta clase representa una actividad con apariencia de diálogo encargada de mostrar los dispositivos Bluetooth a los cuales la aplicación se puede conectar. Este listado es tanto de dispositivos previamente vinculados, como nuevos dispositivos encontrados en una búsqueda reciente.

Figura 29. Clase ListaDispositivosActivity



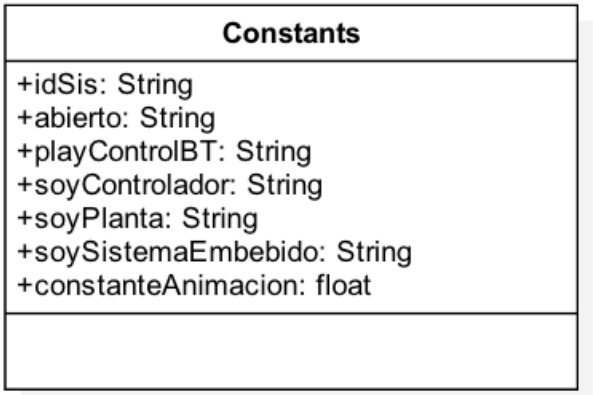
2.7.13 Clase AyudaActivity. Esta Actividad muestra ayuda referente al sistema seleccionado, esta información no tiene relación al uso de la aplicación, por el contrario, su propósito es educativo al mostrarle al usuario información teórica sobre el sistema que desea simular.

Figura 30. Clase AyudaActivity



2.7.14 Clase Constants. Esta clase provee constantes estáticas para ser usadas por cualquier otra clase con el propósito de tener centralizados algunos valores que pueden ser usados desde cualquier lugar de la aplicación y así evitar posibles errores en la codificación.

Figura 31. Clase Constants

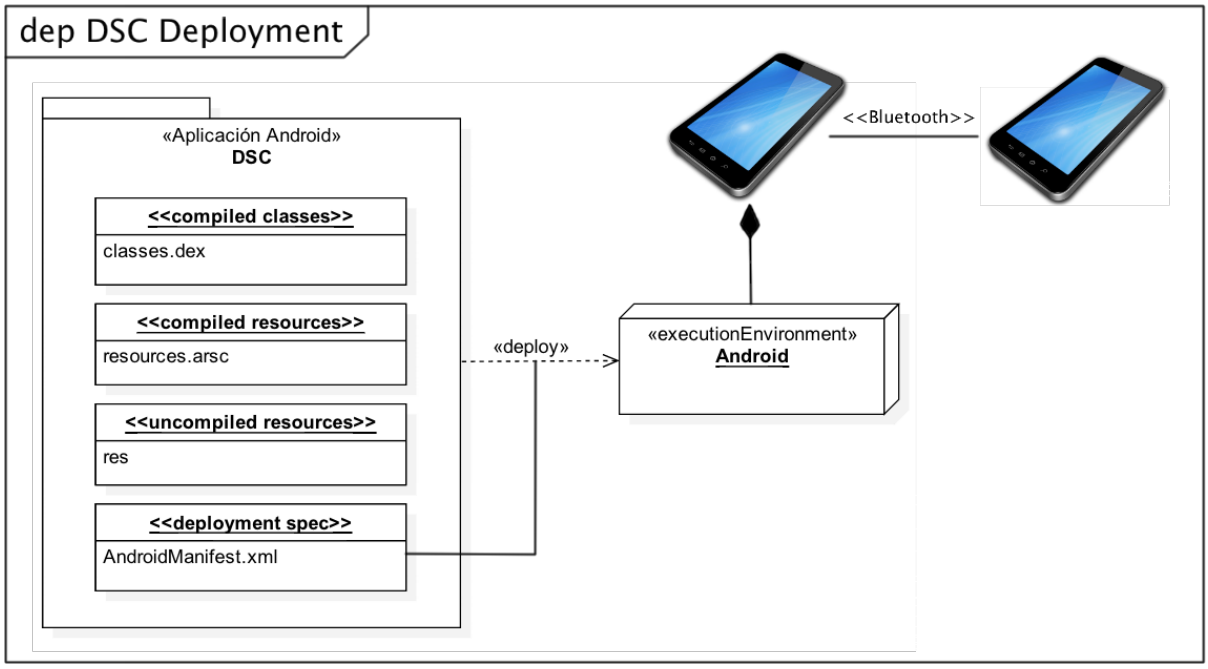


2.8 DIAGRAMA DE DESPLIEGUE

Las herramientas del Kit de Desarrollo de Software (SDK por sus siglas en inglés) de Android compilan y empaquetan el código junto con los demás recursos que no son compilados como las imágenes en un ejecutable con extensión .apk el cual será desplegado en dispositivos con sistema operativo Android. Uno de los

archivos que hace parte de este archivo ejecutable es el AndroidManifest.xml el cual contiene información acerca de la aplicación y sus requerimientos a nivel de acceso a hardware del dispositivo como el uso del módulo Bluetooth e igualmente aquí se encuentran registradas las actividades que componen las interfaces de usuario. El diagrama de despliegue de la figura 32 ilustra estos componentes.

Figura 32. Diagrama de despliegue



3. DESARROLLO DEL PROTOTIPO

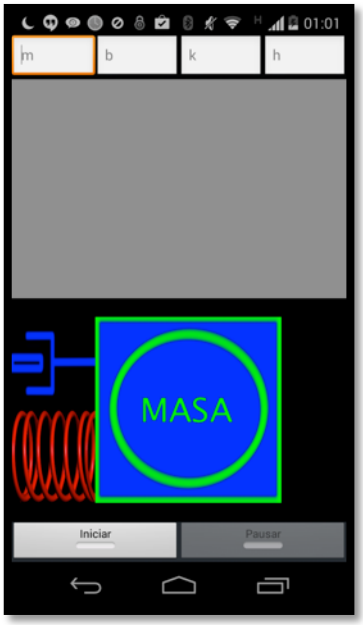
El prototipo se basó en la implementación de un sistema masa-resorte-amortiguador. Su desarrollo incluyó el diseño de interfaces de usuario y la solución algorítmica para embeber los diferentes elementos de un sistema controlado dentro de un único dispositivo. Posteriormente, se adecuó el software para la implementación del controlador a distancia por medio de comunicación Bluetooth e igualmente la adición de perturbaciones por medio de sensores.

3.1 INTERFACES DE USUARIO

El software desarrollado para la plataforma tuvo algunos cambios durante el proceso de implementación, algunos por cuestiones de estética y otros por mejoras en la usabilidad.

El diseño inicial, cumplía su propósito de simular el comportamiento de un sistema masa-resorte-amortiguador a partir de unos parámetros introducidos por el usuario los cuales fueron la masa (m), el coeficiente de amortiguamiento (b), la constante de elasticidad (k) y el paso (h).

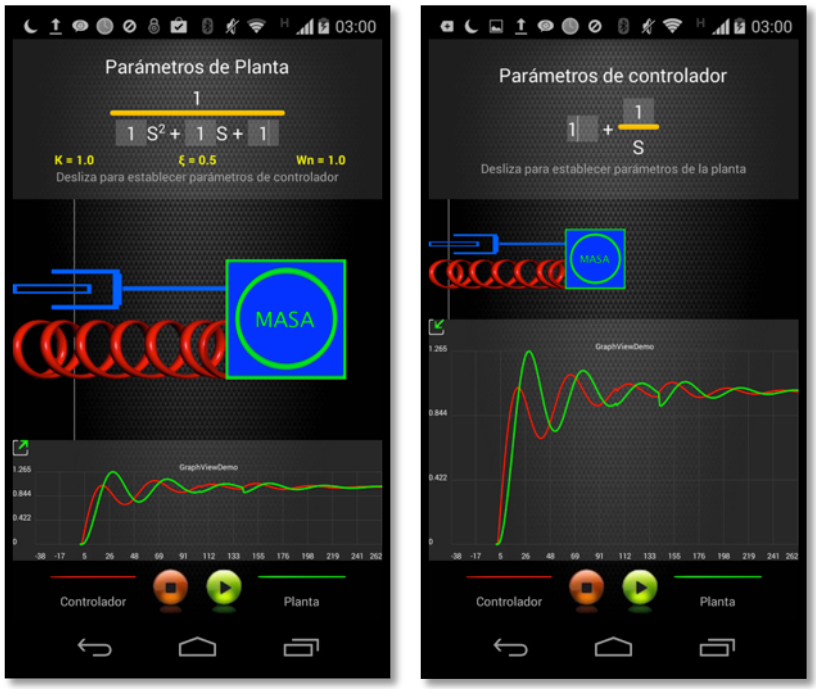
Figura 33. Interfaz de usuario inicial



El diseño que precedió el mostrado en la figura anterior, contó con los elementos que se solicitaron en la especificación de requerimientos, pero tenía algunos problemas de usabilidad que fueron detectados en pruebas no formales con posibles usuarios del sistema.

Debido a que la plataforma debe funcionar sobre dispositivos con pantallas relativamente pequeñas como los Smartphones, se debió hacer un buen uso del espacio que ocuparía cada elemento en la pantalla.

Figura 34. Primera interfaz de usuario funcional



Este diseño se dividió en 4 bloques de forma vertical, el primero de ellos permite la parametrización del sistema (planta y controlador), el segundo muestra la animación del sistema, el tercero muestra una gráfica con el comportamiento del sistema y finalmente un bloque para el control del sistema que permite iniciar, pausar y detener la simulación.

Finalmente, se implementó un último diseño que mejoró la parte visual de la aplicación y su usabilidad.

Figura 35. Interfaz de usuario implementada

(a) (b)

La figura 35(a) muestra una actividad que permite la parametrización del sistema (planta y controlador), la figura 35(b) muestra el sistema masa-resorte-amortiguador representado por un sistema de amortiguamiento de un auto, un elemento para establecer la referencia (set-point) y la gráfica de comportamiento del sistema dinámico.

3.1.1 Librería GraphView. La implementación de la gráfica de comportamiento del sistema fue posible gracias al uso de la librería para Android GraphView, la cual permite de forma programática crear diferente tipo de diagramas[12]. Debido a su facilidad de uso y fácil personalización, esta librería permitió graficar 4 diferentes señales (referencia, salida de controlador, salida de planta y error) sin afectar el desempeño de la aplicación.

En Android, se usa por defecto el patrón arquitectural de software conocido como MVC (Modelo-Vista-Controlador), el cual permite separar las vistas (Interfaces Gráficas de Usuario) del controlador (Clases Java que implementan la parte funcional de la aplicación. De esta forma, se fue trabajando paralelamente en la

[12] GEHRING, Jonas. Librería GraphView. [En línea]. android-graphview 2013.[consultado 7 de marzo de 2015] Disponible en Internet: http://www.android-graphview.org/

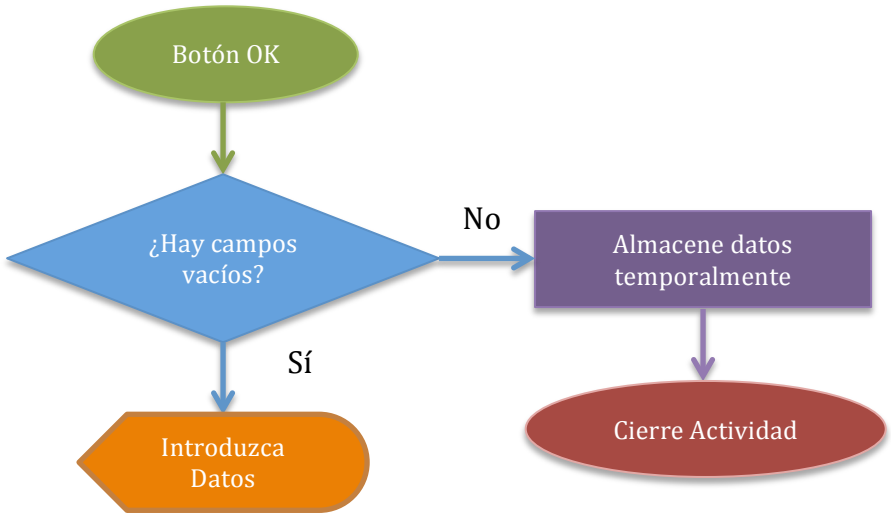
implementación del código fuente que se encargaría de dar funcionalidad a las interfaces gráficas.

3.2 IMPLEMENTACIÓN DEL SISTEMA

El sistema masa-resorte-amortiguador puede ser representado por medio de un modelo matemático basado en ecuaciones diferenciales, esto permite que su implementación por medio de software sea posible. Para la solución de las ecuaciones diferenciales, se empleó el método numérico Runge-Kutta de 4to orden ya que este entrega una aproximación numérica de la solución que para los propósitos de la plataforma es lo suficientemente adecuada.
De acuerdo al diagrama de bloques y de clases, el sistema se dividió en siete paquetes para permitir la integración de nuevos sistemas en un futuro de forma más sencilla.

Inicialmente, se implementó el código de la Actividad de parametrización que posee el diseño mostrado en la figura 35(a). En la clase propia de esta Actividad, una vez se pulsa el botón OK, se recogen y se almacenan temporalmente los datos ingresados por el usuario en los campos: masa, coeficiente de amortiguamiento, constante de elasticidad y en caso de ser en lazo cerrado los parámetros ganancia proporcional (Kp), ganancia integral (Ki) y ganancia derivativa (Kd). Estos datos serán entregados posteriormente a la siguiente Actividad.

Figura 36. Diagrama de flujo de parametrización del sistema

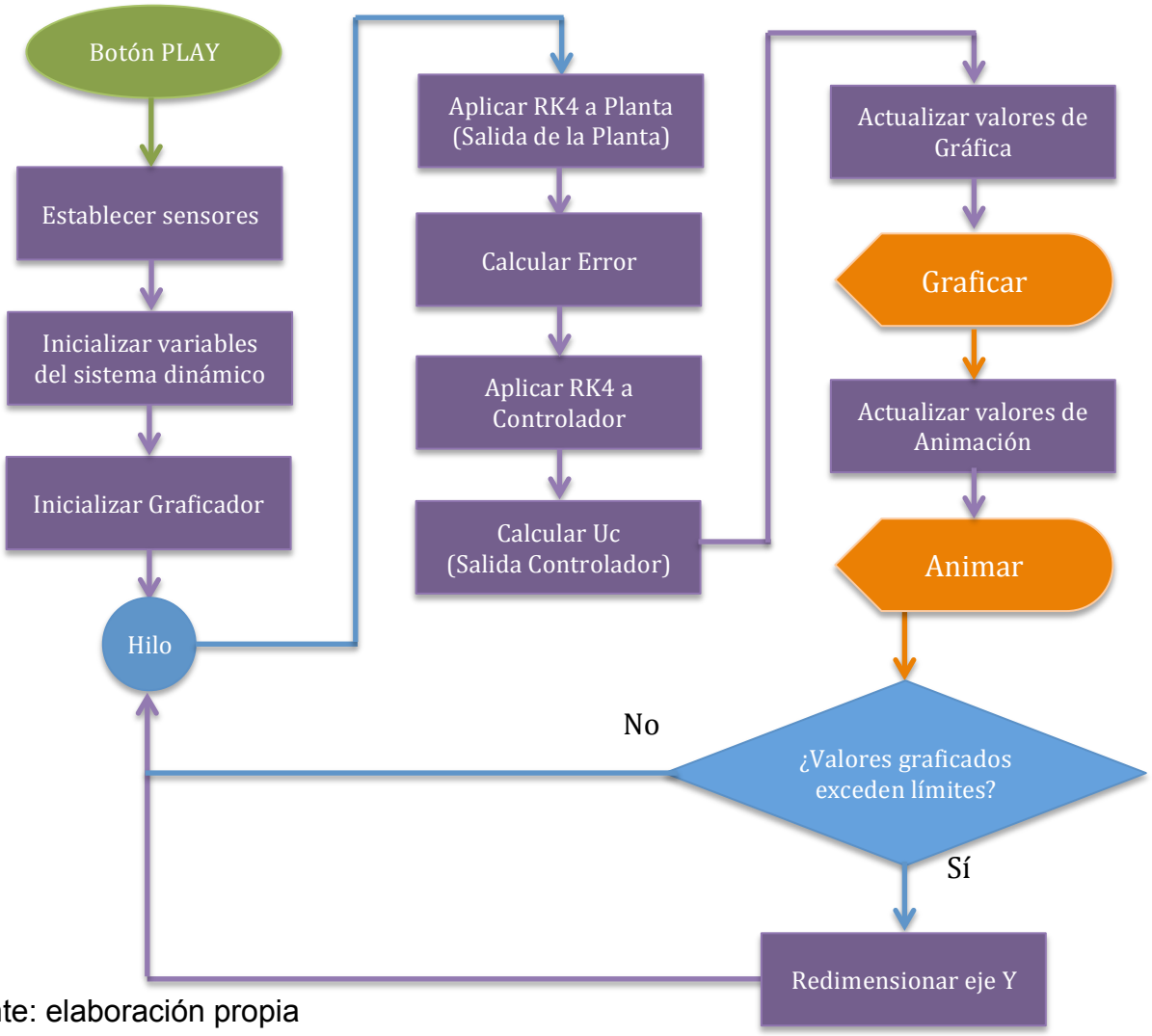


Fuente: elaboración propia

La siguiente Actividad se muestra una vez se parametriza el sistema dinámico. Desde aquí el usuario tiene la opción de establecer la señal de referencia con un valor de 0 a 5 usando el control horizontal y finalmente iniciar la simulación del sistema pulsando el botón “Play” o iniciar el modo de controlador externo pulsando el botón Bluetooth. El control del sistema puede ser embebido (controlador y planta en un mismo dispositivo) o externo (controlador y planta en diferentes dispositivos conectados por Bluetooth).

3.2.1 Modo Embebido. El flujo de la simulación se observa en el siguiente diagrama.

Figura 37. Diagrama de flujo de inicio de simulación



Fuente: elaboración propia

Al iniciar la simulación, la aplicación establece los sensores que se requieran para el sistema seleccionado, en este caso, se usó el sensor de proximidad para inducir disturbios al sistema masa-resorte-amortiguador. Adicionalmente inicializa las variables propias de dicho sistema y el graficador.
La simulación se ejecuta en un hilo para evitar que la aplicación se bloquee o se detenga debido al procesamiento matemático y gráfico que debe realizar. Este hilo se ejecuta de forma continua hasta que el usuario pulse el botón de "PAUSA" o "DETENER"

Dentro del hilo, la aplicación toma los parámetros de la planta y el controlador, para proceder a aplicar el método Runge-Kutta de 4to orden (RK4) a la planta, cuyo resultado es tomado como la salida de esta y será usado para calcular el Error (*Ec. 19*)

$$SalidaPlanta = RK4\ Planta \quad Ec.18$$

$$Error = Referencia - (SalidaPlanta) \quad Ec.19$$

Por cada ciclo, la salida de la planta cambiará debido al RK4.

Con este nuevo valor (Error), se ejecuta el RK4 para el controlador. Con este resultado se puede calcular la salida del controlador (Uc) usando la *Ec. 22.*

$$SalidaControlInteg = RK4\ Controlador \quad Ec.20$$

$$DerError = \frac{(Error - ErrorAnterior)}{Ts} \quad Ec.21$$

$$Uc = Kp * Error + Ki * SalidaControlInteg + Kp * DerError \quad Ec.22$$

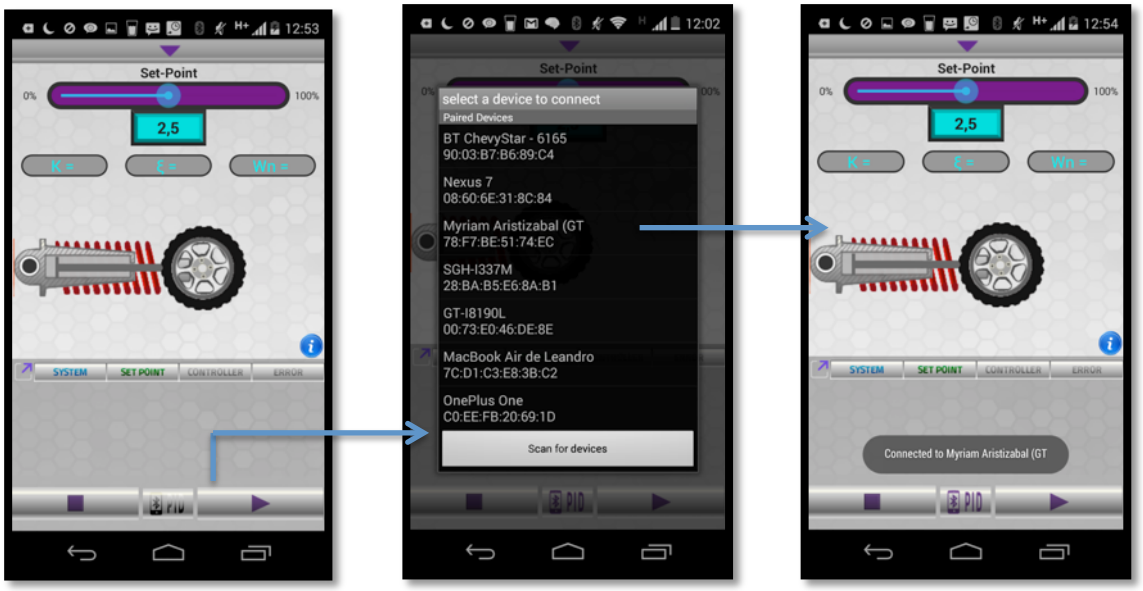
La salida del controlador alimenta la planta y el ciclo se repite de forma continua. La salida de la planta se almacena por la aplicación de forma temporal para animar el sistema usando este valor como el desplazamiento de la masa en el eje X. Este mismo valor al igual que la salida del controlador, son graficados para mostrar las dinámicas del sistema. Adicionalmente, si el usuario mueve su mano sobre el sensor de proximidad, se dispara un evento que usa esta señal como disturbio adicionando un 15% de la referencia a la salida de la planta.

3.2.2 Modo Externo. Esta característica permite dividir el controlador trabajando en un dispositivo (Smartphone/Tablet) y la planta en otro (Smartphone/Tablet). Para lograr esto, se usó la tecnología inalámbrica Bluetooth provista por los dispositivos como un puente para realimentar el sistema.

La operación es la misma al modo embebido, pero en este caso el dispositivo que actúa como controlador no ejecutará ninguna acción perteneciente a los procesos propios de la planta, la cual recibirá la salida del controlador vía Bluetooth y esta a su vez enviará su salida al controlador para el cálculo del Error.

Para este modo, uno de los usuarios deberá asumir el rol de controlador con su dispositivo, para esto, deberá pulsar el botón Bluetooth (figura 35(a)) que al ser pulsado ejecuta acciones relacionadas con el Bluetooth del dispositivo como verificar si este está encendido y de no estarlo solicitar permiso al usuario de encenderlo. Con el Bluetooth activo, se hace una búsqueda de dispositivos que deberán estar visibles para asumir el rol de planta (figura 35(b)). Una vez los dispositivos se han enlazado, el usuario con el dispositivo controlador podrá iniciar la simulación del sistema el cual mostrará la animación y mostrará la gráfica con sus dinámicas en este dispositivo.

Figura 38. Selección de control externo

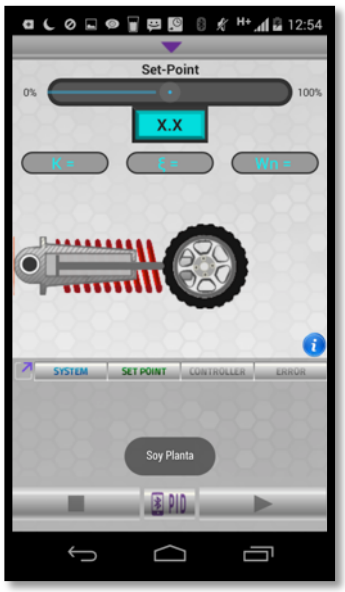


(a) (b) (c)

Por otra parte, el dispositivo al que se realizó la conexión asume el rol de planta y se le desactivará el control del Set-Point ya que este solo podrá ser modificado por

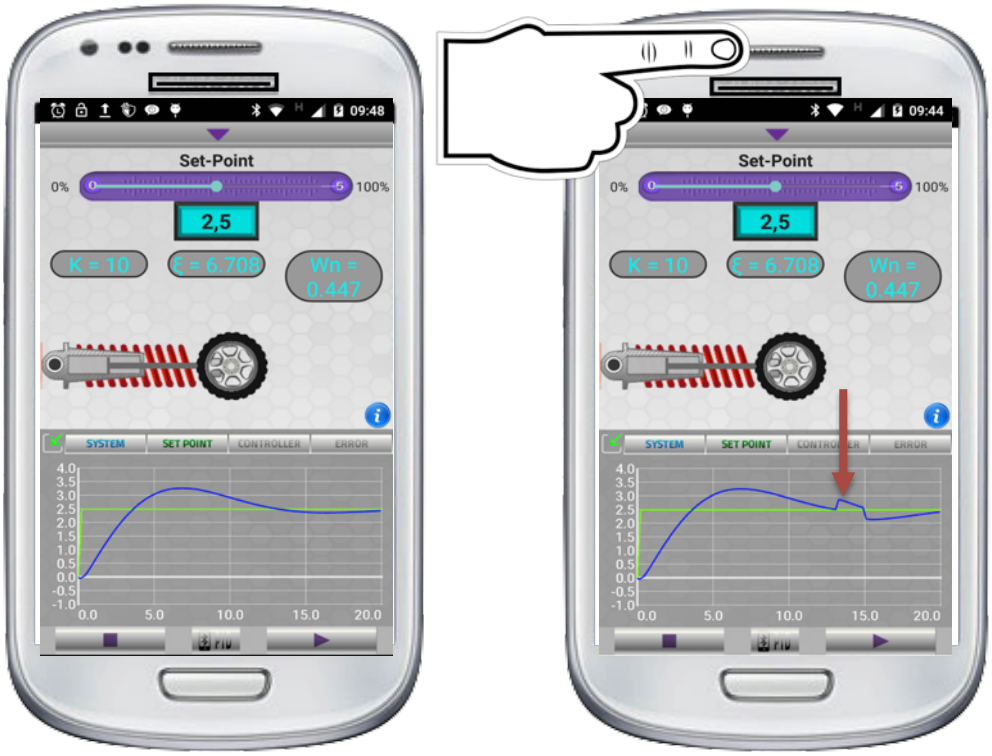
el dispositivo con rol de controlador, al igual que los botones de “Play” y “Stop” (figura 39).

Figura 39. Interfaz de planta remota



Por último, la aplicación se dotó con soporte para usar uno de los sensores del dispositivo (sensor de proximidad), con el propósito de emular una perturbación en el sistema que halará la masa y en este caso el controlador responderá para llevarla de nuevo a su punto de referencia. Si se libera la masa (se deja de usar el sensor), nuevamente el controlador estabilizará el sistema. La figura 40 ilustra dichos comportamientos.

Figura 40. Sensor de proximidad para emular perturbación del sistema



Una vez se contó con la interfaz gráfica adecuada y el sistema piloto funcionando, se decidió implementar las pruebas de usabilidad con el propósito de obtener información relacionada con la facilidad de uso y diseño de la aplicación por parte de posibles usuarios, en este caso estudiantes de la Universidad Autónoma de Occidente.

3.3 PRUEBAS DE USABILIDAD

La aplicación móvil DSC fue desarrollada con el propósito de servir como herramienta de estudio para el aprendizaje de sistemas dinámicos y sistemas de control en la Universidad Autónoma de Occidente. La aplicación fue sometida a pruebas de usabilidad con potenciales usuarios los cuales son estudiantes de la misma universidad.

3.3.1 Plan de pruebas

- **Propósito y Alcances.** El propósito de este estudio fue evaluar la experiencia de usuario de los estudiantes de la Universidad Autónoma de Occidente con la aplicación DSC mientras interactúan con ella. La información entregada por el estudio buscó entregar información como:

  - Observaciones de comportamiento e información relevante de la actual experiencia de usuario.
  - Ideas para mejorar y fortalecer la experiencia determinando inconsistencias y áreas de problema de usabilidad dentro de la interfaz de usuario y áreas de contenido. Fuentes potenciales de error incluyen:
  - Errores de navegación – Falla al encontrar funciones, demasiadas pulsaciones para completar una función, fallas al seguir el flujo recomendado de la pantalla.
  - Presentación de errores – Falla al encontrar y actuar apropiadamente frente a información deseada en la pantalla, errores de selección debido a ambigüedades en la información mostrada.
  - Problemas de uso de control – Uso inapropiado de menús o campos de entrada.
  - Información de base sobre la actual experiencia de usuario que pueda ser usada en comparación con futuras experiencias.

- **Metodología**

Objetivos. Con la aplicación DSC, los usuarios (estudiantes) debieron alcanzar los siguientes logros:

  - Simular un sistema dinámico (masa-resorte-amortiguador) en lazo abierto
  - Simular un sistema dinámico (masa-resorte-amortiguador) en lazo cerrado
  - Encontrar ayuda sobre el uso de la aplicación
  - Simular un sistema dinámico (masa-resorte-amortiguador) en lazo cerrado entre dos dispositivos asumiendo uno el rol de controlador y el otro el rol de planta

Preguntas de Investigación. El estudio recogió datos cualitativos y cuantitativos para responder múltiples preguntas de investigación, incluyendo:

  - **Realización de tareas** – Qué tan bien la aplicación respalda la habilidad de los usuarios para completar tareas específicas?
  - **Navegación e información de la arquitectura** – Cómo la estructura de la aplicación respalda la habilidad de los usuarios de completar sus tareas? Pueden ellos navegar a dónde desean y completar sus tareas rápida y eficientemente? Qué caminos toman?

- **Contenido y terminología –** Los usuarios entienden el contenido y este los ayuda a completar sus tareas?
- **Diseño visual –** Cuál es la impresión de los usuarios sobre el diseño visual?
- **Comunicación e impresiones de la aplicación–** Cuáles son las impresiones en general de los usuarios respecto a la aplicación? La aplicación comunica adecuadamente lo que los usuarios pueden/necesitan hacer con ella?

Diseño del estudio. Se dirigió un estudio de usabilidad en persona dentro de las instalaciones de la Universidad Autónoma de Occidente usando la última versión desarrollada de la aplicación en dispositivos Android propios de los participantes para obtener información sobre el desempeño del usuario con la aplicación y descubrir necesidades no cubiertas. El estudio recogió información tal como tiempo de realización de las tareas, navegación y percepción del contenido, satisfacción general, áreas de inquietud y necesidades sin suplir.

Para el estudio se realizó una entrevista previa a la prueba y otra posterior a esta para recoger información sobre la percepción del participante antes y después de la prueba sobre la aplicación.

La interacción de los participantes con la aplicación fue monitoreada por un facilitador y tres tomadores de nota quienes simultáneamente atendieron las inquietudes de los participantes y registraron los eventos más relevantes durante la realización de cada tarea.

El facilitador solicitó a los participantes que “pensaran en voz alta” mientras usaban la aplicación y dicha información fue registrada para posterior análisis. Sin previo aviso a los participantes, los tomadores de nota registraron el “primer clic/pulsación” de cada tarea.

Tareas. Las tareas a desarrollar por parte de los estudiantes fueron:

- **TAREA 1**

Usted se encuentra en el laboratorio realizando una práctica de sistemas dinámicos y necesita probar el comportamiento de un sistema controlado masa-resorte-amortiguador. Los parámetros del sistema a simular son:

- Ganancia proporcional =
- Ganancia Integral =
- Masa =
- Coeficiente de amortiguamiento =
- Constante de elasticidad =

Criterio de tarea finalizada con éxito: Inicio de la simulación del sistema una vez se hayan ingresado los parámetros

- **TAREA 2**

Después de realizar la tarea anterior, se le pide que perturbe el mismo sistema para verificar la acción de control ante dicha perturbación.

Criterio de tarea finalizada con éxito: Pasar la mano sobre el sensor de proximidad

- **TAREA 3**

Usted se encuentra en el laboratorio realizando una práctica de sistemas dinámicos y necesita probar el comportamiento de un sistema masa-resorte-amortiguador en lazo abierto. Los parámetros del sistema a simular son:

  - Masa =
  - Coeficiente de amortiguamiento =
  - Constante de elasticidad =

Criterio de tarea finalizada con éxito: Inicio de la simulación del sistema una vez se hayan ingresado los parámetros

- **TAREA 4**

Durante una sesión de laboratorio, el docente solicita crear grupos de 2 personas para una práctica en equipo. El propósito de la práctica es que usted y su compañero realicen la implementación de un sistema MRA en el cual el dispositivo de un estudiante hará las veces de controlador (remoto) y el dispositivo del otro estudiante asumirá el rol de planta. Los parámetros del controlador y de la planta deberá introducirlo cada estudiante de forma independiente en su respectivo dispositivo.

Criterio de tarea finalizada con éxito: Inicio de simulación del sistema en ambos dispositivos una vez se hayan ingresado los parámetros en cada uno de ellos.

- **TAREA 5**

Después de simular un sistema masa-resorte-amortiguador, usted requiere analizar el comportamiento con mayor detenimiento, para esto decide ampliar la gráfica y analizar los valores de principio a fin.

Criterio de tarea finalizada con éxito: Uso de gestos para ampliar la gráfica y deslazarse en el tiempo.

Participantes. Se condujo el estudio con 6 estudiantes del curso Sistemas Dinámicos y Control de los programas de Ingeniería Mecatrónica e Ingeniería Electrónica de la Universidad Autónoma de Occidente. Se consideró hacer el estudio con esta audiencia debido a que la aplicación está dirigida principalmente a estudiantes de dicho curso y a la comunidad universitaria en general. Los

estudiantes seleccionados además de poseer un dispositivo con sistema operativo Android, debían dominar el tema a evaluar para conseguir resultados más confiables sobre el uso de la herramienta.

La responsabilidad de los participantes fue intentar completar un conjunto de tareas representativas presentadas de forma eficiente y oportuna, además de proveer realimentación respecto a la usabilidad y aceptación de la interfaz de usuario. Los participantes fueron dirigidos para que provean opiniones honestas sobre el uso de la aplicación.

Preguntas previas a la prueba

- Alguna vez ha usado software para simular/graficar el comportamiento de sistemas dinámicos?
- Alguna vez ha usado aplicaciones móviles para el curso de Sistemas dinámicos/Control?
- Cuáles son sus expectativas sobre una aplicación para simular sistemas dinámicos?
  - Qué características o servicios esperaría encontrar?
  - Qué servicio le parecería particularmente grandioso en una aplicación móvil para simular sistemas dinámicos?
  - Qué características no le gustaría encontrar en una aplicación de estas?

Preguntas posteriores a la prueba

- Has trabajado con la aplicación DSC por 30 minutos, encuentras que te pueda ser de utilidad?
- Consideras que le hace falta algo?
- Si has usado otro tipo de aplicaciones (móviles/escritorio), como podrías comparar esta?
- Tienes algún comentario/observación sobre el diseño de las interfaces gráficas?
- Menciona una o dos cosas que consideras requieren mejorarse?
- Menciona una o dos cosas que te hayan parecido particularmente buenas de la aplicación DSC.
- Si tuvieras que describir esta aplicación a un compañero en una oración o dos, qué le dirías?
- Usarías esta aplicación en el futuro?

3.3.2 Resultados de las pruebas

Los resultados que arrojaron las pruebas de usabilidad permitieron identificar algunos problemas a nivel de diseño y funcionalidad.

- **Preguntas previas a la prueba**
  - El 100% de los participantes identificó que la aplicación tiene relación con el estudio de sistemas dinámicos y sistemas de control tan solo con ver su menú principal.
  - El 100% de los participantes creen que la aplicación está diseñada para estudiantes/personas que deseen aprender sobre sistemas de control.
  - El 100% de los participantes ha usado software para simular/graficar el comportamiento de sistemas dinámicos.
  - Matlab y LabView es el software más usado por los participantes para simular/graficar el comportamiento de sistemas dinámicos. Las características que destacan de dichos programas son su robustez y complejidad que extiende el tiempo en la curva de aprendizaje.
  - Ninguno de los participantes ha usado aplicaciones móviles en el curso de sistemas dinámicos.
  - Dentro de las expectativas que los participantes tuvieron de la aplicación antes de hacer uso de ella se encuentran:
    - El trabajo con diferentes tipos de sistemas.
    - La manipulación de diferentes variables.
    - Sencilla de usar (intuitiva).
    - Robusta
    - Ágil
    - Confiable
  - En cuanto a las características que esperaron encontrar se tiene:
    - La gráfica debe mostrar el momento/tiempo de estabilización del sistema.
    - Comparación entre sistemas.
    - Manipulación de sistemas modificando sus variables en tiempo real.
    - Trabajo con diferentes tipos de sistemas.
    - Sistemas reales e ideales
    - Fácil de usar
  - Los servicios que encontrarían particularmente grandiosos fueron:
    - Que la aplicación entregue la función de transferencia del sistema.
    - Trabajar con muchas variables al tiempo.
    - Simulaciones interactivas
    - Gráficas que sirvan de guía
  - Las características que no esperan encontrar son:
    - Información poco confiable.
    - Que acepte información no válida como valores negativos donde no sea posible.
    - Que contenga bugs.

- **Tarea 1**
  - Prueba del primer clic (tap)
    - El 83% de los participantes escogió la opción correcta (Amortiguación de un auto) como primer paso en la realización de la tarea.
  - El 83% de los participantes completó con dificultad o ayuda la tarea. Dentro de los problemas presentados se encuentran:
    - Un usuario intenta perturbar el sistema con el dedo
    - La aplicación se bloquea
    - La variable zita del sistema no está siendo bien calculada.
    - Un usuario no hizo uso de la ayuda.
    - Un usuario no pudo usar el botón play/stop
    - Un usuario no encontró utilidad en la barra del Set-Point (Referencia)

- **Tarea 2**
  - El 100% de los participantes no pudo completar la tarea. Entre las razones de no haber finalizado la tarea se encuentra:
    - No se hizo uso de la ayuda.

- **Tarea 3**
  - Prueba del primer clic (tap)
    - El 100% de los participantes escogió la opción Lazo Abierto de Amortiguación de un Auto como primer paso en la realización de la tarea.
  - El 100% de los participantes finalizó la tarea sin dificultad.

- **Tarea 4**
  - Prueba del primer clic (tap)
    - El 100% de los participantes escogió la opción Lazo Cerrado de Amortiguación de un Auto como primer paso en la realización de la tarea.
  - El 67% de los participantes finalizó la tarea con éxito. Razones por las que no se terminó la tarea:
    - Para el 33% de los participantes no fue claro el uso del control remoto.
    - La aplicación se cerraba inesperadamente.

- **Tarea 5**
  - El 100% de los participantes finalizó con éxito la tarea. Fue fácil identificar el botón de ampliación e igualmente usar el gesto de “pellizco” para ampliar o reducir la gráfica.

- **Preguntas de cierre/impresiones del usuario**

- El 100% de los participantes encuentra que la aplicación puede ser de utilidad para entender el comportamiento de sistemas básicos los cuales son motivo de estudio en sus cursos.
- Los participantes consideran que a la aplicación le hace falta o requiere mejorarse lo siguiente:
  - No es clara la barra para parametrizar el sistema.
  - El botón de control remoto (Bluetooth) no es claro.
  - No se sabe que pasa con un sistema inestable.
  - Se deben modificar las escalas de la gráfica.
  - Verificar la toma de datos y la simulación
  - Faltan ayudas
  - Ejemplos de simulación.
- Para los participantes esta aplicación se compara con otras (móviles/escritorio):
  - La aplicación DSC es más dinámica, más rápida de usar para explicar un tema y aclararlo.
  - Le falta robustez
  - Es más fácil entender un sistema dinámico.
- Acerca del diseño de las interfaces los participantes opinan:
  - Es muy plano.
  - Agradable. Facilita guiarse en su uso.
  - Colocar unidades en las gráficas.
  - Menú principal ordenado.
  - Hacer categorías de sistemas.
- Dentro de las cosas que los participantes consideran requiere mejorarse:
  - Especificar cual es la referencia.
  - Más velocidad en la simulación.
  - Cambiar el ícono de control remoto ya que está escondido.
  - Colocar la gráfica en una segunda ventana independiente.
- Los participantes consideran particularmente bueno de la aplicación:
  - El tipo de simulación a realizar (Lazo abierto/cerrado) está bien especificado.
  - Permite ver múltiples señales en la gráfica.
  - Se tienen diferentes tipos de sistemas.
  - Las animaciones son buenas.
- Los participantes describirían la aplicación a otros de las siguientes maneras:
  - “Es una aplicación sencilla para darse cuenta a groso modo lo que se verá en el curso de Sistemas dinámicos”.
  - “Aplicación básica del modelo de un sistema dinámico”.
  - “Aplicación excelente para la interpretación de sistemas dinámicos”.
- Finalmente, el 100% de los participantes considera útil la aplicación y definitivamente la usarían y recomendarían.

3.4 PRUEBAS DE FUNCIONALIDAD

El sistema DSC fue sometido a pruebas de funcionamiento que demostraran cuán útil puede ser su uso en entornos de aprendizaje, para lo cual se requiere que los resultados entregados sean lo más confiable posibles. En este caso, su funcionamiento fue comparado con los resultados que entregan Matlab y la herramienta Simulink para el mismo sistema bajo las mismas condiciones y parámetros.

Las pruebas fueron realizadas en dos dispositivos diferentes con las siguientes características:

- **Dispositivo A**
  - Marca: Google
  - Fabricante: Asus
  - Tipo: Tablet
  - Tamaño: 7”
  - Modelo: Nexus 7
  - Versión de Android: 4.2.2 (Jelly Bean)
  - Memoria RAM: 1GB
  - CPU: Cortex-A9 quad core (1.3 GHz)

- **Dispositivo B**
  - Marca: Motorola
  - Fabricante: Motorola
  - Tipo: Smartphone
  - Tamaño: 5”
  - Modelo: XT1068
  - Versión de Android: 5.0.2 (Lollipop)
  - Memoria RAM: 1GB
  - CPU: Qualcomm Snapdragon 400 quad-core (1.2 GHz)

Para el sistema masa-resorte-amortiguador, su función de transferencia es:

$$\frac{X(s)}{F(s)} = \frac{1/m}{S^2 + \frac{b}{m}s + \frac{k}{m}} \quad Ec.23$$

Con base en esta información se procede a hacer las siguientes pruebas en lazo abierto y cerrado.

3.4.1 Prueba funcional del sistema en lazo abierto

- **Caso A**
  - Masa (M) = 1
  - Constante del amortiguador (B) = 0.5
  - Constante del resorte (K) = 1
  - Referencia (set-point) = 3

Se obtiene la siguiente función de transferencia

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 0.5s + 1} \qquad Ec. 24$$

A continuación se muestra la información que entrega Matlab y la aplicación DSC.

Figura 41. Respuesta de Matlab en el caso A

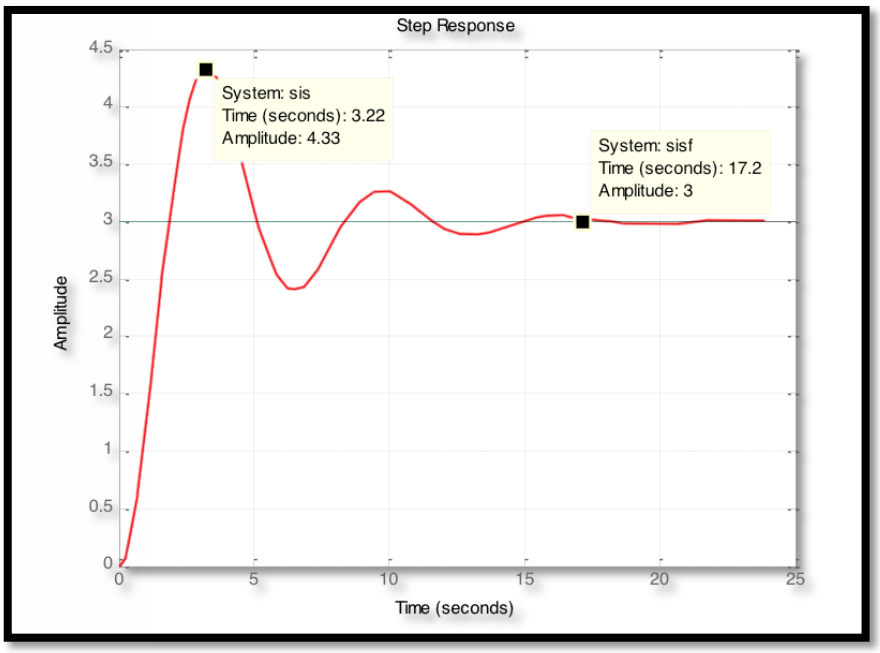
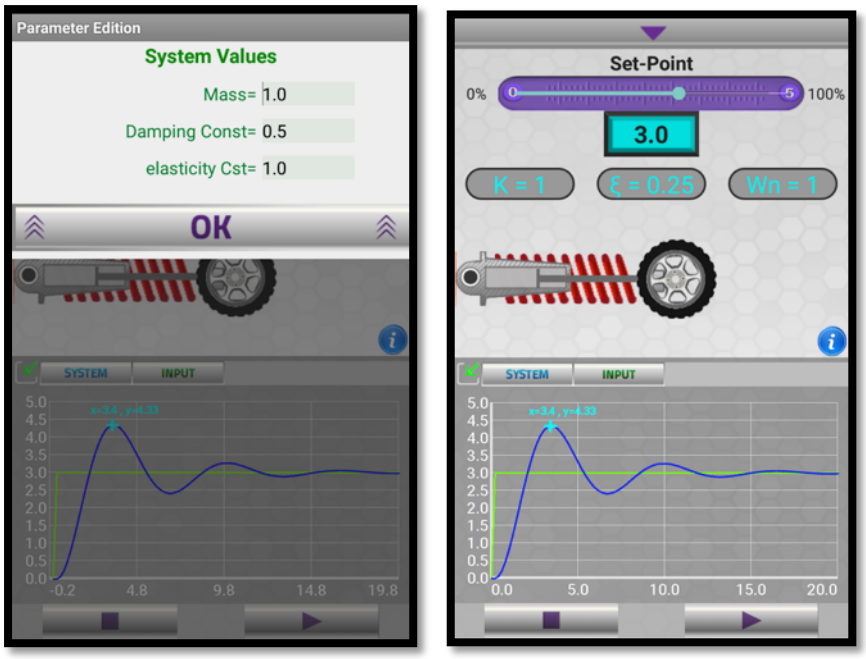
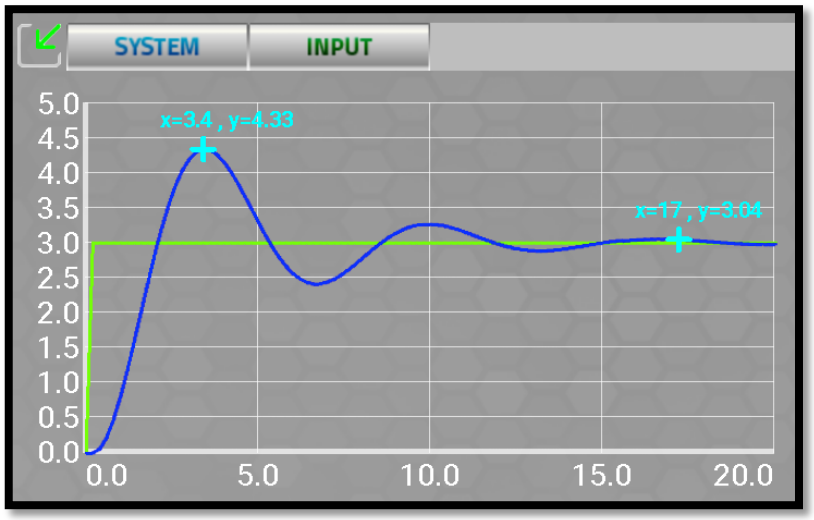


Figura 42. Respuesta de DSC en el caso A



Se puede observar de las figuras 41 y 42 que las respuestas son semejantes en forma y cantidad de oscilaciones.

Figura 43. Máx. Sobrepico y tiempo de estabilización de DSC en el caso A



En la figura 43 se observa que el máximo sobrepico es de 4,33 y el tiempo de estabilización es de aproximadamente 17 segundos.

- **Caso B**
  - Masa (M) = 1
  - Constante del amortiguador (B) = 0.3
  - Constante del resorte (K) = 1
  - Referencia (set-point) = 3

Se obtiene la siguiente función de transferencia

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 0.3s + 1} \qquad Ec.25$$

A continuación se muestra la información que entrega Matlab y la aplicación DSC.

Figura 44. Respuesta de Matlab en el caso B

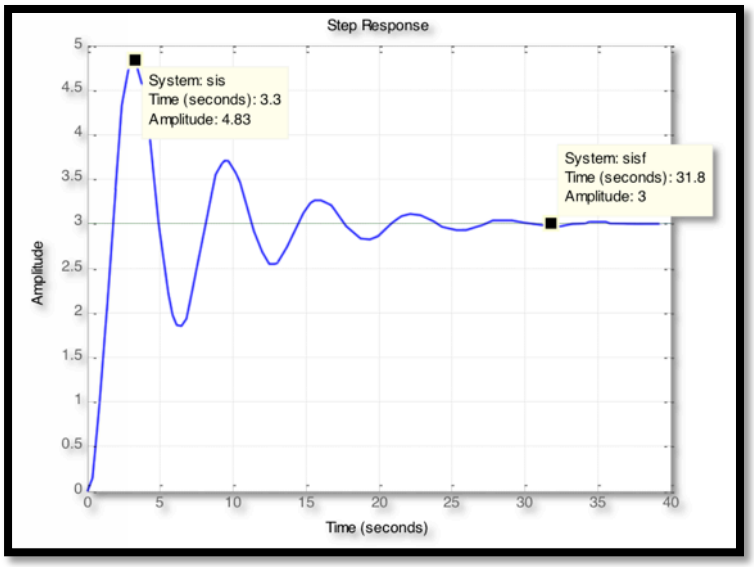
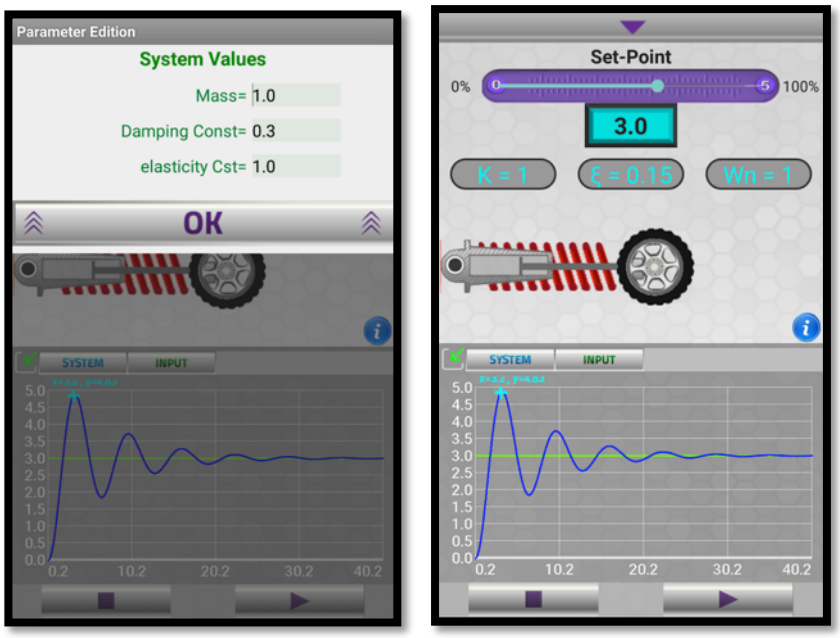
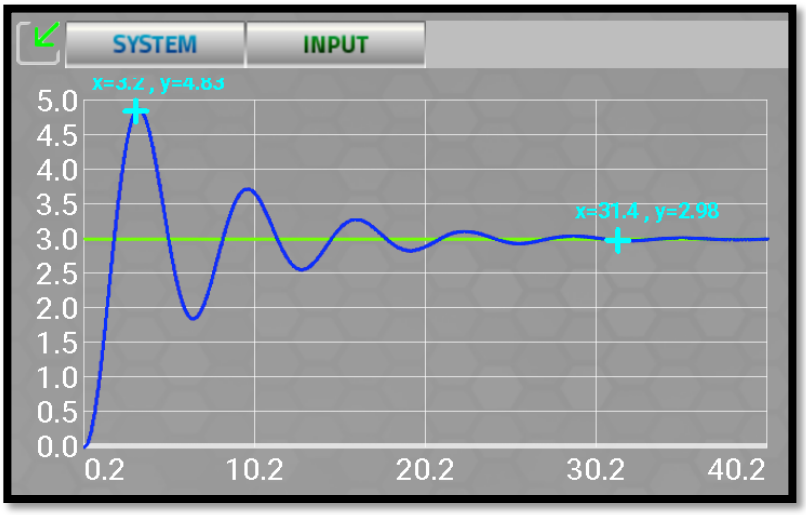


Figura 45. Respuesta de DSC en el caso B



Se puede observar de las figuras 44 y 45 que las respuestas son semejantes en forma y cantidad de oscilaciones.

Figura 46. Máx. Sobrepico y tiempo de estabilización de DSC en el caso B



En la figura 46 se observa que el máximo sobrepico es de 4,83 y el tiempo de estabilización es de aproximadamente 31,4 segundos.

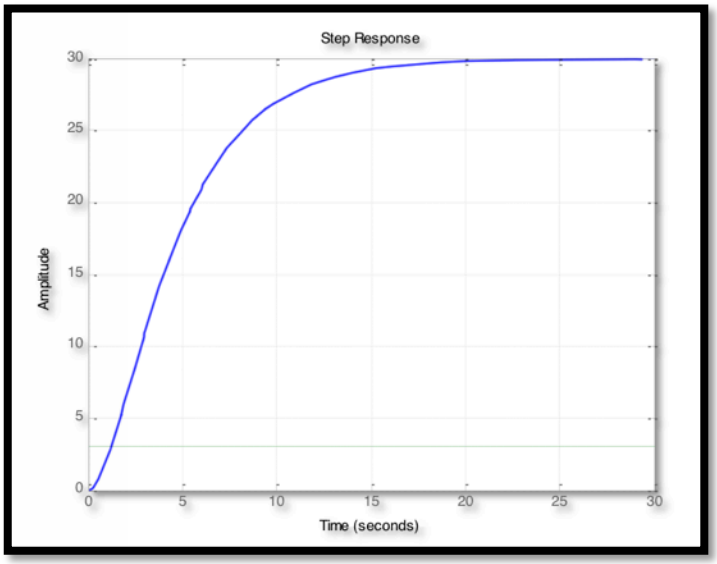
- **Caso C**
  - Masa (M) = 0.5
  - Constante del amortiguador (B) = 0.5
  - Constante del resorte (K) = 1
  - Referencia (set-point) = 3

Se obtiene la siguiente función de transferencia

$$\frac{X(s)}{F(s)} = \frac{2}{s^2 + s + 0.2} \qquad Ec. 26$$

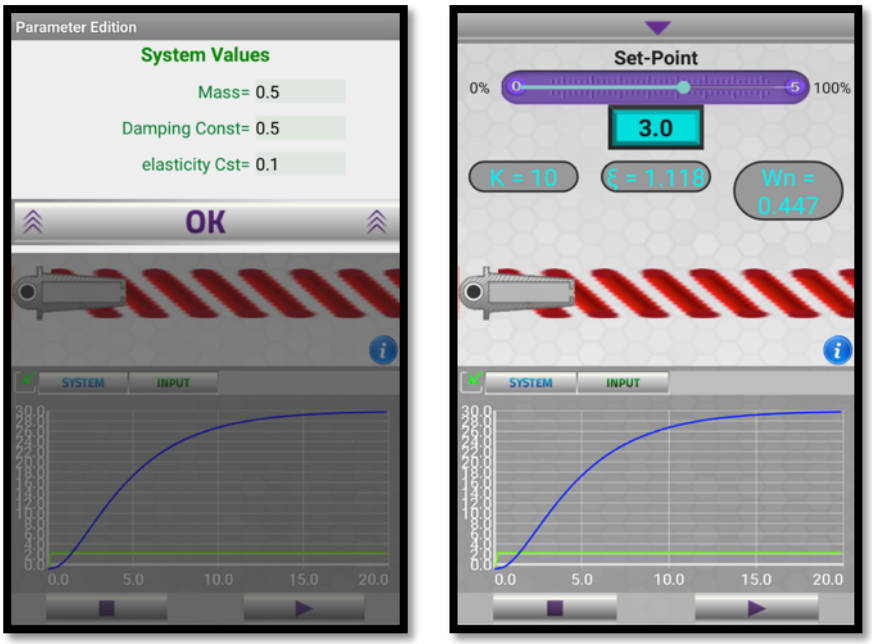
A continuación se muestra la información que entrega Matlab.

Figura 47. Respuesta de Matlab en el caso C



La figura 48 muestra la respuesta de la aplicación DSC.

Figura 48. Respuesta de DSC en el caso C



Se puede observar de las figuras 47 y 48 que las respuestas son semejantes en forma y cantidad de oscilaciones. Igualmente se observa que el tiempo de estabilización es aproximadamente 20 segundos y que la salida alcanza un valor de estabilización de 30 en ambos casos.

- **Caso D**
  - Masa (M) = 1
  - Constante del amortiguador (B) = 0
  - Constante del resorte (K) = 1
  - Referencia (set-point) = 3

Se obtiene la siguiente función de transferencia

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 1} \qquad Ec. 27$$

A continuación se muestra la información que entrega Matlab.

Figura 49. Respuesta de Matlab en el caso D

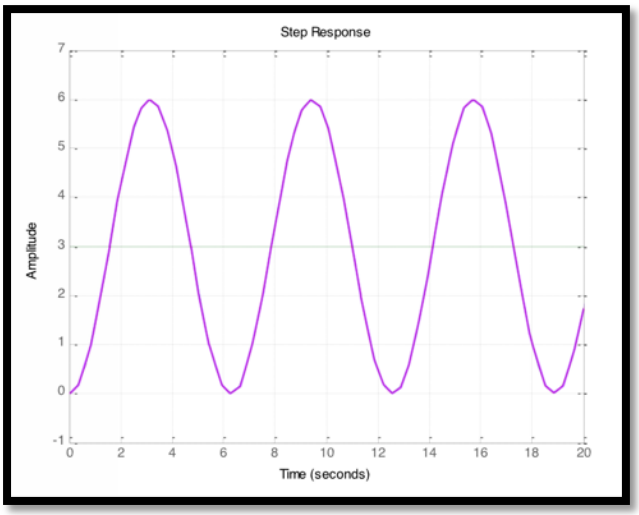
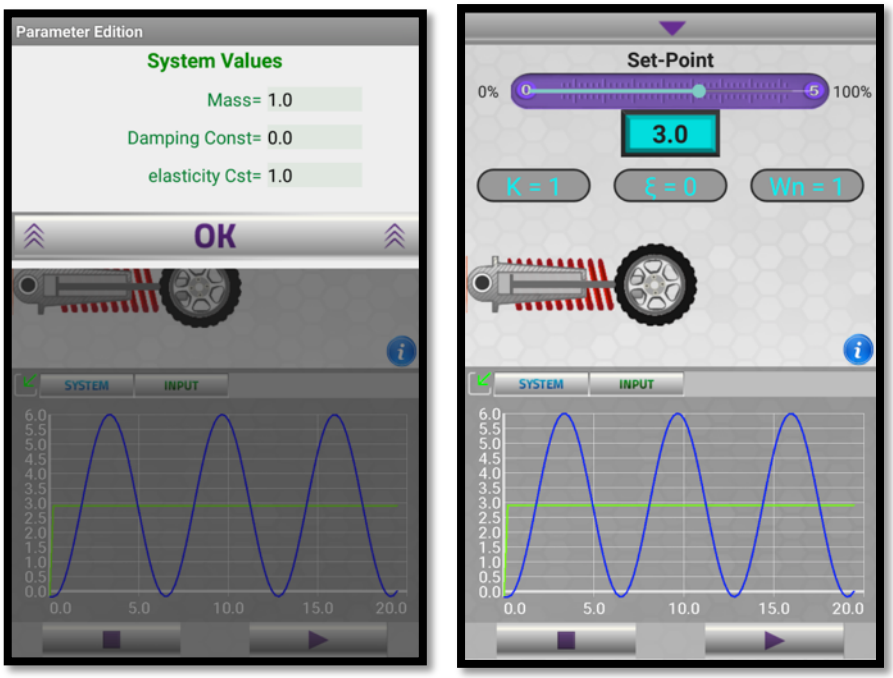


Figura 50. Respuesta de DSC en el caso D

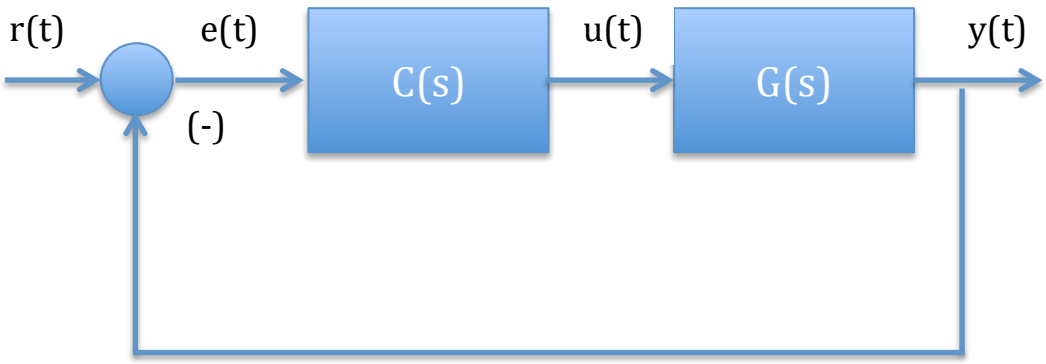


Se observa que la respuesta de la aplicación es similar a la obtenida a través de Matlab para los mismos parámetros. Ambas oscilan con picos ente 0 y 6 e

igualmente corresponden a una oscilación no amortiguada, respuesta típica para sistemas con coeficiente de amortiguamiento cero.

3.4.2 Prueba funcional del sistema en lazo cerrado. Se realizarán ahora las pruebas de lazo cerrado usando para ello un esquema como el que se muestra a continuación:

Figura 51. Esquema de un sistema controlado en lazo cerrado



Fuente: elaboración propia

Donde C(s) será un controlador PID y G(s) será la planta a controlar , se usará el caso A y caso D para observar el funcionamiento del sistema.

Estructura del controlador

$$U(s) = \left(K + \frac{I}{s} + Ds\right) E(s)$$

- **Caso A (Controlador P)**
  - Masa (M) = 1
  - Constante del amortiguador (B) = 0.5
  - Constante del resorte (K) = 1
  - Referencia (set-point) = 3
  - Kp=1
  - Ki=0
  - Kd=0

Se obtiene la siguiente función de transferencia

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 0.5 + 1} \qquad Ec. 28$$

A continuación se muestra el esquema de simulación en Simulink de Matlab y su respuesta temporal.

Figura 52. Esquema de simulación del sistema en lazo cerrado en Simulink

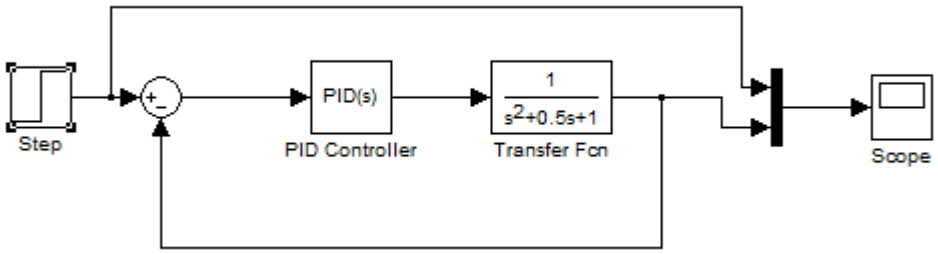
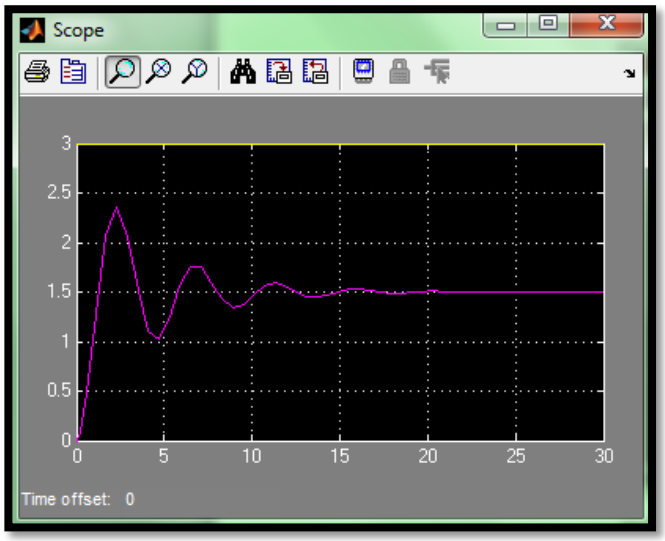
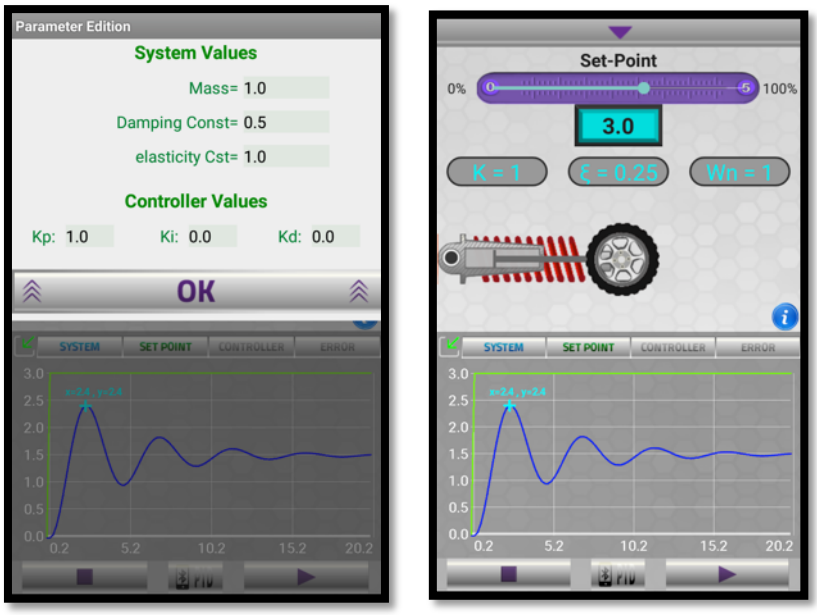


Figura 53. Respuesta de Simulink en el caso A



En el sistema DSC se ajustó el controlador PID de tal forma de tener sólo una acción proporcional con valores Kp=1 (Ki=0, Kd=0)

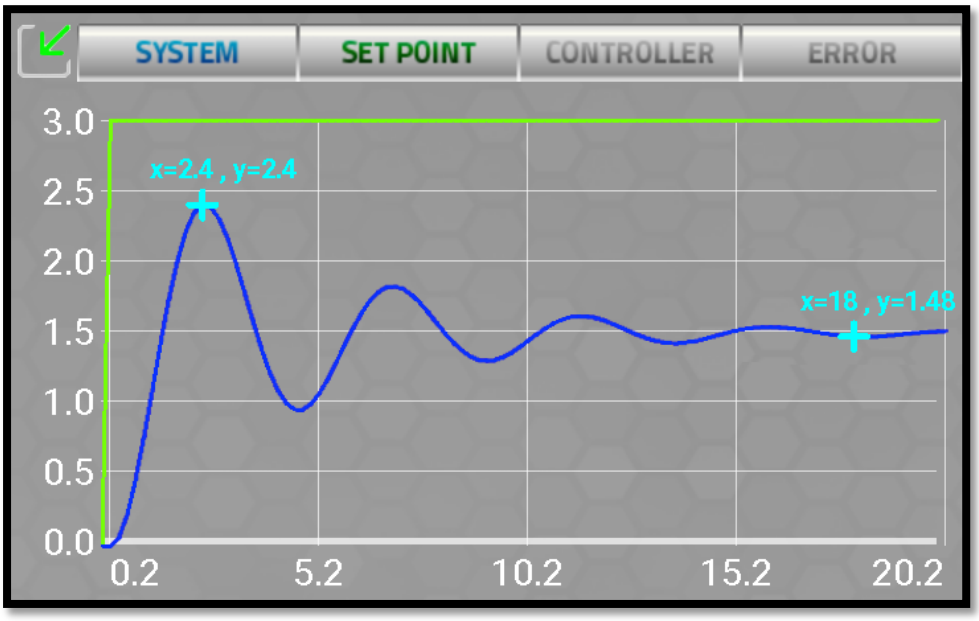
Figura 54. Respuesta de DSC en el caso A de lazo cerrado



Se observa que la respuesta del sistema corresponde en forma a la respuesta obtenida a través de Matlab, apreciándose la característica perdida de ganancia (50%) para este caso por ser realimentación unitaria con constante proporcional K del PID ajustada en K=1.

En la siguiente figura se aprecia que hay coincidencia en el máximo valor obtenido por la salida, estabilizándose ambos en el tiempo de estabilización, aproximadamente 18 segundos.

Figura 55. Máx. Sobrepico y tiempo de estabilización de DSC en el caso A de lazo cerrado

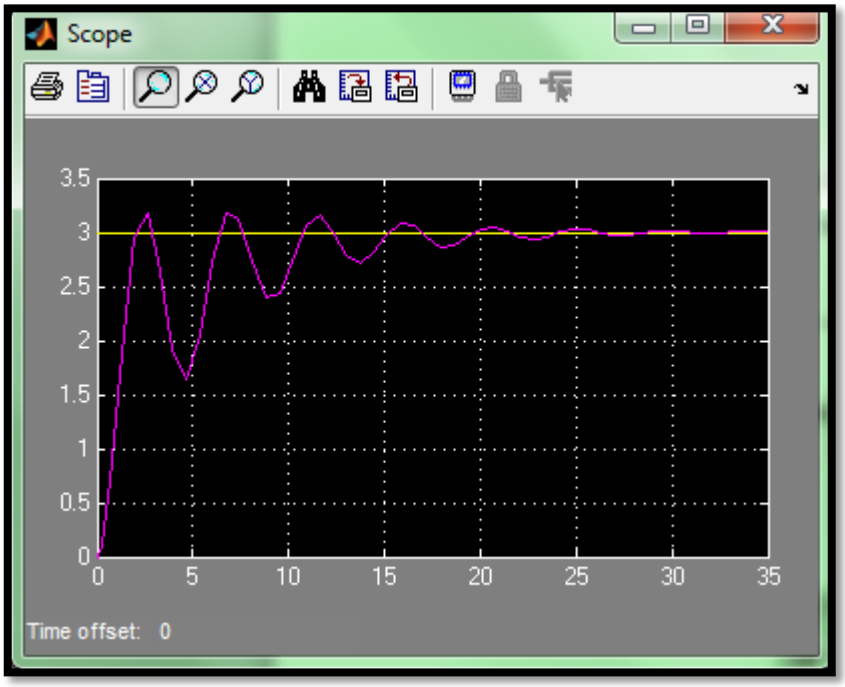


- **Caso B (Controlador PI)**
  - Masa (M) = 1
  - Constante del amortiguador (B) = 0.5
  - Constante del resorte (K) = 1
  - Referencia (set-point) = 3
  - Kp=1
  - Ki=0.4
  - Kd=0

Se mantiene el mismo esquema de lazo cerrado del caso anterior (Caso A)

A continuación se muestra la respuesta temporal en Simulink de Matlab

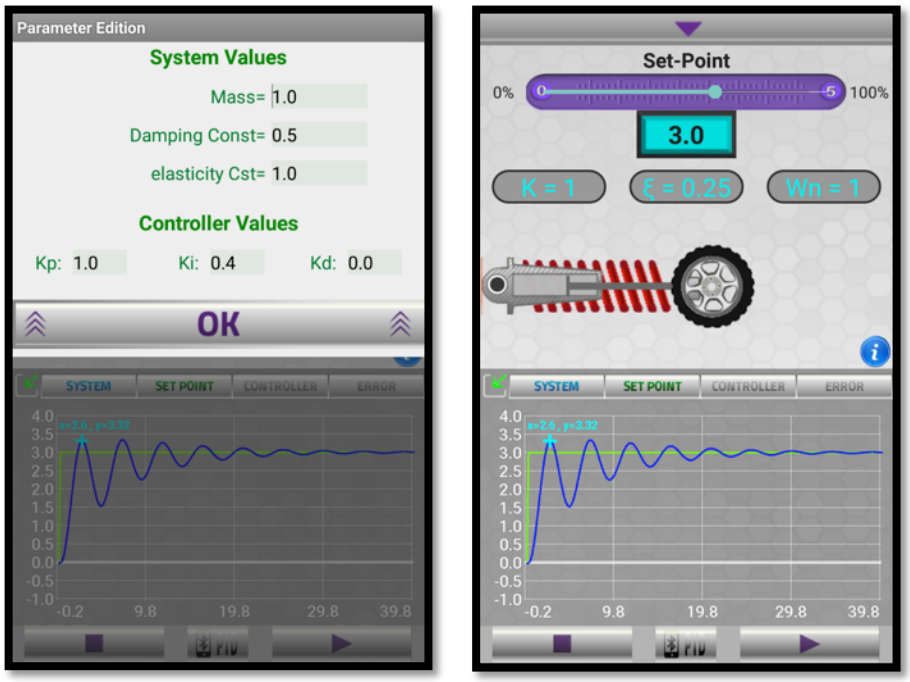
Figura 56. Respuesta de Simulink en el caso B



A diferencia del caso anterior (A) se debe apreciar que el sistema tiende a seguir la referencia (set-point) suministrado (escalón con ganancia 3), acorde con lo que se espera de un sistema PI debido al efecto de la acción integral (I) que apareció en este caso.

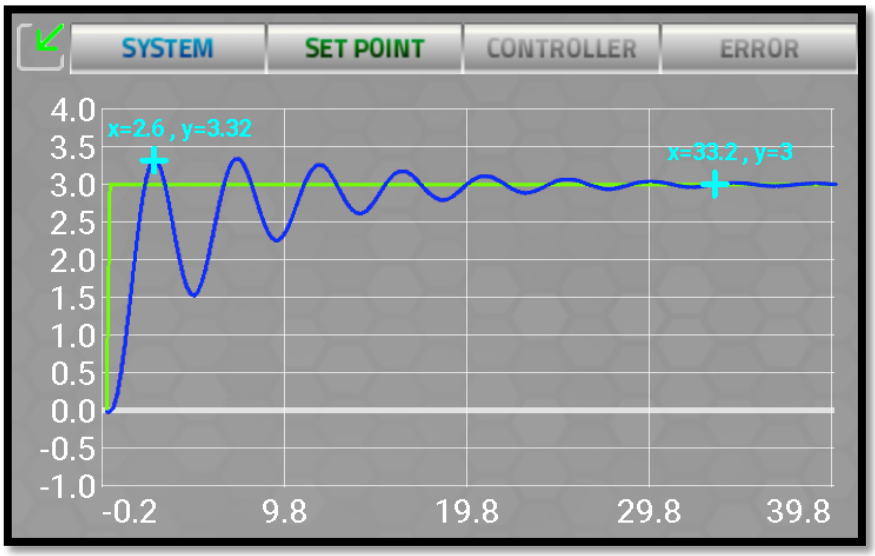
Se presentan ahora los resultados obtenidos en la aplicación DSC para los mismos parámetros.

Figura 57. Respuesta de DSC en el caso B de lazo cerrado



Se observa que la simulación obtenida a través de la aplicación DSC corresponde con la obtenida en Matlab tanto en forma como en valores característicos de la misma.

Figura 58. Máx. Sobrepico y tiempo de estabilización de DSC en el caso B de lazo cerrado

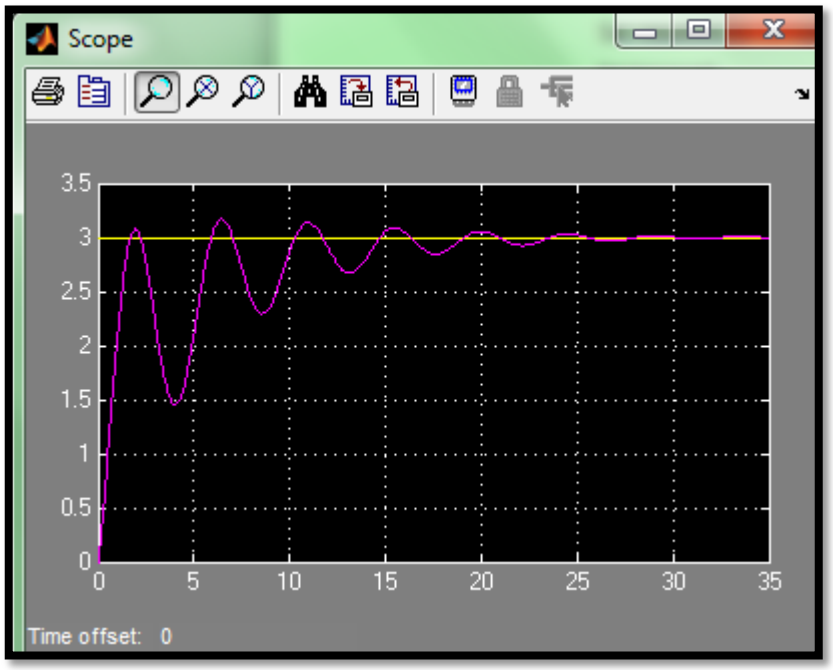


- **Caso C (Controlador PID)**
  - Masa (M) = 1
  - Constante del amortiguador (B) = 0
  - Constante del resorte (K) = 1
  - Referencia (set-point) = 3
  - Kp=1
  - Ki=0.4
  - Kd=0.5

En este caso se usará el mismo ejemplo del caso D en lazo abierto y se ajustará el controlador PID con los parámetros Kp=1, Ki= 0.4 y Kd=0.5

A continuación se muestra la respuesta temporal en Simulink de Matlab

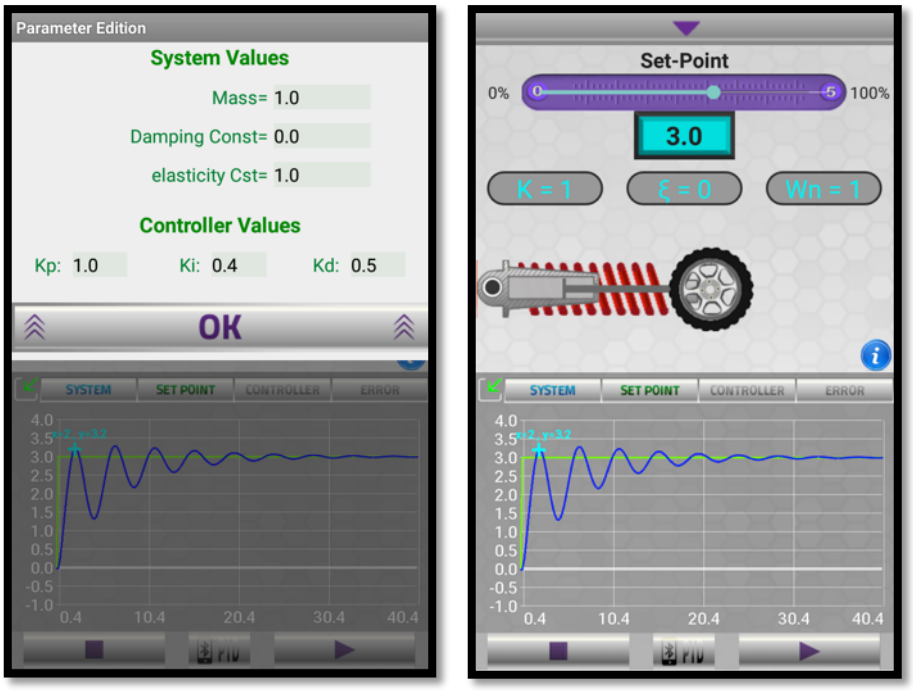
Figura 59. Respuesta de Simulink en el caso C



A diferencia del caso D en lazo abierto donde la respuesta era una oscilación no amortiguada por ser un sistema con coeficiente de amortiguación cero, en este caso la acción de control estabiliza el sistema llevándolo a su valor de referencia (set-point).

Se presentan ahora los resultados obtenidos en la aplicación DSC para los mismos parámetros.

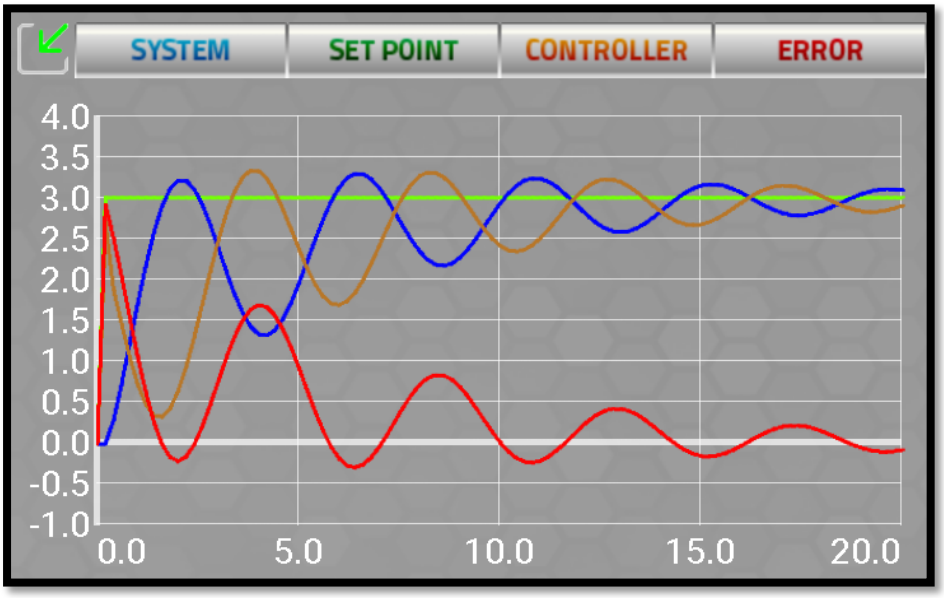
Figura 60. Respuesta de DSC en el caso C de lazo cerrado



Se observa que la simulación obtenida a través de la aplicación DSC corresponde con la obtenida en Matlab tanto en forma como en valores característicos de la misma.

Adicionalmente, la siguiente figura muestra las diferentes respuestas que pueden ser visualizadas con la aplicación DSC, en este caso la salida de la planta (azul), la referencia (verde), la salida del controlador (naranja) y el error (rojo).

Figura 61. Señales de visualización en la aplicación DSC del caso C de lazo cerrado

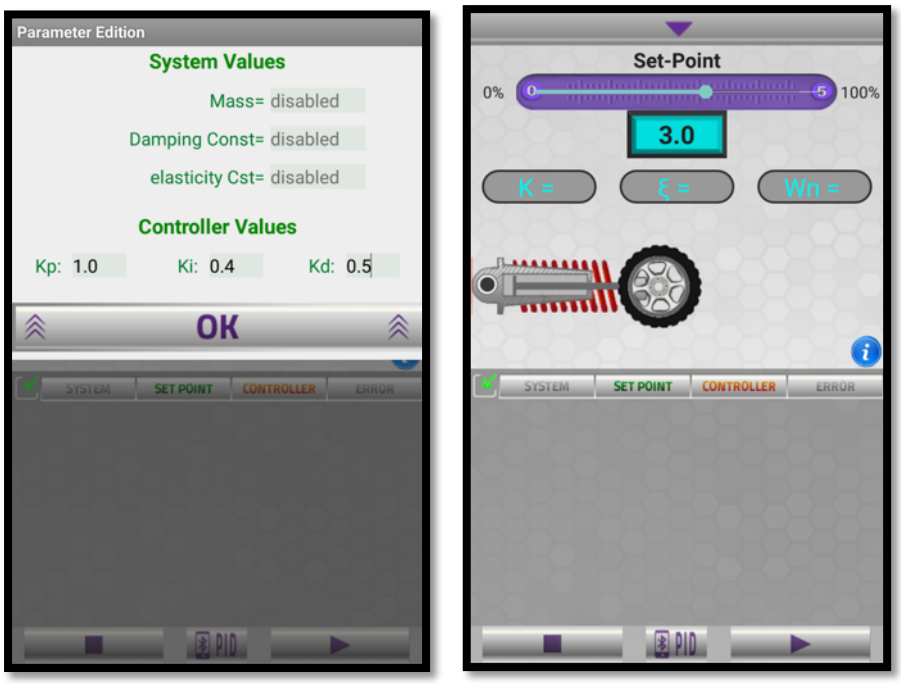


- **Caso D (Controlador PID externo)**
  - Planta
    - Masa (M) = 1
    - Constante del amortiguador (B) = 0
    - Constante del resorte (K) = 1
  - Controlador
    - Referencia (set-point) = 3
    - Kp=1
    - Ki=0.4
    - Kd=0.5

En este caso se usará el mismo ejemplo del caso anterior (caso C) y se ajustará igualmente el dispositivo que actúa como controlador PID con los parámetros Kp=1, Ki= 0.4, Kd=0.5 y Referencia=3, mientras en el dispositivo que actúa como planta se parametriza M=1, B=0 y K=1.

A continuación se observa la interfaz del dispositivo controlador y su parametrización.

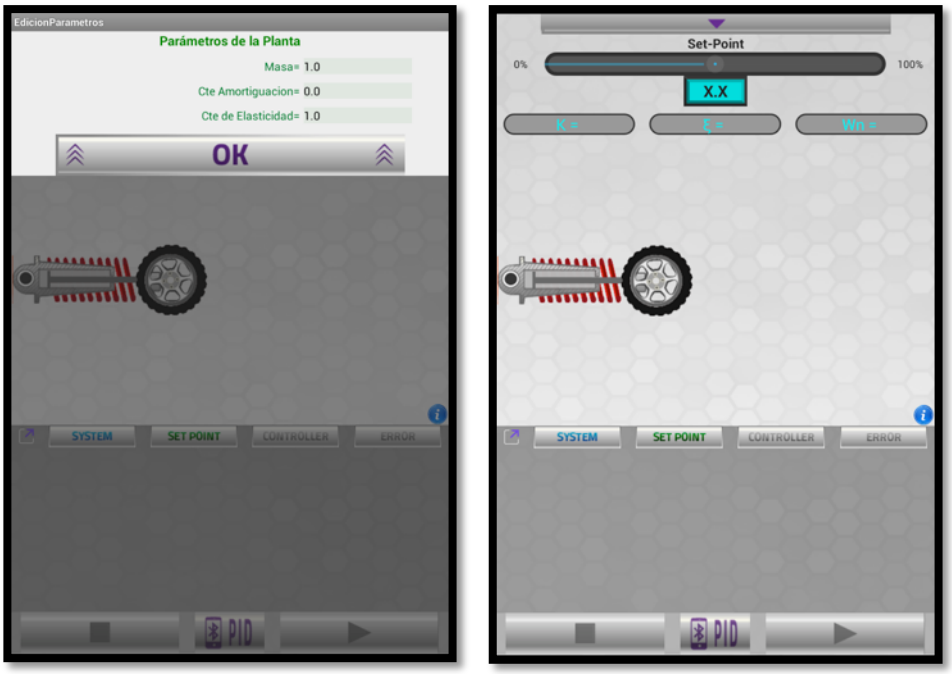
Figura 62. Interfaz de dispositivo controlador remoto



Se puede observar que la actividad en la cual se ingresan los parámetros del controlador se encuentran deshabilitados los campos de masa, constante de amortiguamiento y constante de elasticidad.

A continuación, se muestra la interfaz del dispositivo planta y su parametrización.

Figura 63. Interfaz de dispositivo planta remota



Se puede observar que la actividad en la cual se ingresan los parámetros de la planta no aparecen los campos de parametrización del controlador Kp, Ki, Kd. En la actividad de la simulación, se ha deshabilitado el control de la referencia (set-point), al igual que el indicador de dicha referencia y los controles de inicio de simulación (botón play) y detener animación (botón stop). Esto se hizo con el propósito de dejar el control del sistema del lado del dispositivo controlador y en este caso el dispositivo planta solo servirá como monitor en el cual se visualiza la animación del sistema y las señales que el usuario desee visualizar.

Una vez el usuario pulsa el botón de inicio de simulación en el dispositivo controlador, se inicia un intercambio de información vía Bluetooth entre los dos dispositivos. A continuación se puede ver la respuesta de cada dispositivo.

Figura 64. Respuesta de DSC en el caso D en lazo cerrado con controlador y planta separados.

(a) (b)

Se puede observar que las señales son las mismas presentadas en el caso C solo que esta vez se visualizan por separado en dos dispositivos. En la figura 64(a) se ve la salida del controlador (naranja), mientras en la figura 64(b) se ve la salida de la planta (azul)

4. PERSPECTIVAS FUTURAS

La plataforma DSC fue diseñada desde el principio con el propósito de que fuera modular y de esta forma ir extendiendo su funcionalidad adicionando nuevos sistemas dinámicos.

Se propone adicionar nuevas formas de onda en la señal de referencia y la posibilidad de seleccionar diferentes tipos de controladores para evaluar su eficiencia.

Adicionalmente, ya que los datos simulados se muestran de forma temporal, la plataforma podría incluir un sistema de almacenamiento para llevar un registro de los datos generados durante la simulación para posterior análisis.

De igual forma, se plantea la posibilidad de controlar una planta real haciendo uso de la tecnología Bluetooth con la cual ya cuenta este sistema en su funcionamiento en modo de control externo para lo cual será necesario el desarrollo de un componente hardware que permita recibir la señal de control via Bluetooth y transformarla en una señal estándar de control (4-20 mA, 0-5 volt).

5. CONCLUSIONES

- Con la encuesta realizada a los estudiantes de asignaturas del área de control automático, se percibe la aceptación por parte de ellos a este tipo de recursos y en general al aprendizaje móvil con lo que se espera que ellos se motiven a trasladar los espacios de aprendizaje a lugares distintos al aula de clase, dándoles la posibilidad de manejar el tiempo de estudio de acuerdo a sus propias necesidades.

- En el presente trabajo se mostró la factibilidad de usar dispositivos móviles para la enseñanza de conceptos relacionados con el comportamiento y análisis de sistemas dinámicos, lo anterior se hizo desarrollando una aplicación que simula un sistema dinámico estándar como lo es el masa-amortiguador-resorte siendo controlado por un controlador tipo PID.

- El uso de hardware y procesamiento matemático hizo necesaria la implementación del sistema a través de un desarrollo nativo, en este caso para el sistema operativo Android. Esto debido a que se requiere un alto rendimiento y las herramientas de desarrollo multiplataforma no optimizan el código reduciendo así el desempeño de la aplicación.

- El rendimiento de algunos dispositivos afecta directamente la simulación de un sistema, añadiendo retardos entre el controlador y la planta debido a la ejecución del Runge-Kutta para ambos elementos del sistema. Esto es particularmente notable cuando se trabaja en el modo de control externo, ya que la realimentación se hace vía comunicación Bluetooth.

- Las tiendas de aplicaciones móviles están en camino a convertirse en un repositorio de herramientas didácticas y educativas gracias al desarrollo de la metodología de aprendizaje móvil poniéndolas a disposición del mundo entero.

- Los Smartphones y Tablets se están convirtiendo en una herramienta esencial en el proceso enseñanza/aprendizaje debido a las capacidades de su hardware y software embebidas en un dispositivo con un costo relativamente bajo.

BIBLIOGRAFIA

ALLY, Mohamed. Mobile Learning (Transforming the Delivery of Education and Training). AU Press, Athabasca University. Canadá. eISBN: 9781897425442 [En línea]En: ebrary Digital Library 2009. [consultado el 20 de Abril de 2013] Disponible en Internet: http://site.ebrary.com.

ARROWSMITH, D.K. Y PLACE, C.M. An Introduction to Dynamical Systems. Cambridge University Press. 1994.432p

AZIZ A. "Mechanical Systems," in Systems Dynamics & Control. [En línea]. Opencourseware 2007. [consultado 10 de Octubre de 2014]. Disponible en Internet: http://opencourseware.kfupm.edu.sa/colleges/ces/me/me413/files%5C2-Chapters_CHAPTER_III____MECHANICAL_SYSTEMS_3.pdf

Android-Overview. [En línea]. OPEN HANDSET ALLIANCE. 2013. [consultado 07 de Junio de 2013]. Disponible en internet http://www.openhandsetalliance.com/android_overview.html

CHAPRA, Steven C y CANALE, Raymond P. Métodos numéricos para ingenieros Ed. 5. McGraw Hill. 2007. 961p.

FAKHROUTDINOV, Kirill. Android Application (UML Deployment Diagram Example). [En línea]. uml-diagrams.org 2014. [consultado 10 de Octubre de 2014]. Disponible en Internet: http://www.uml-diagrams.org/android-application-uml-deployment-diagram-example.html

GEHRING, Jonas. Librería GraphView. [En línea]. android-graphview 2013. [consultado 10 de Octubre de 2014]. Disponible en Internet: http://android-graphview.org/

GOK, Nizamettin y KHANNA, Nitin. Building Hybrid Android Apps. O'Reilly. USA. 2013. 156p.

GRUPO PRISA DIGITAL. Tipos de Sistemas de Control. [En línea]. co.kalipedia.com 2011. [consultado 31 de Marzo de 2013]. Disponible en Internet: http://co.kalipedia.com/tecnologia/tema/robotica

KIM, Daniel. What is User Centered Design. [En línea]. danielikim.2015. [consultado el 24 de Abril de 2015]. Disponible en internet <http://danielikim.com/what-is-user-centered-design/>

Introduction to Dynamic Simuation. [En línea]. NATIONAL INSTRUMENTS. 2015. [consultado 28 de Enero de 2015]. Disponible en internet http://www.ni.com/tutorial/11606/en/

OGATA, Katsuhiko. Ingeniería de Control Moderna 4 Ed. Pearson Educación, S.A. Madrid. 2003. 953p

----------. Dinámica de Sistemas. Prentice-Hall Hispanoamericana, S.A. Naucalpán de Juarez. 1987. 652p.

New Frontiers in Education. [En línea]. UNIVERSITY OF ROCHESTER. 2012. [consultado 10 de Marzo de 2015]. Disponible en internet: http://www.rochester.edu/it/pluggedin/new-frontiers-in-education/

SERRANO SANTOYO, Arturo y ORGANISTA SANDOVAL, Javier. Challenges and Opportunities to Support Learning with Mobile Devices. MexIHC'2010. [En línea] ACM Digital Library [consultado el 20 de Abril de 2013] Disponible en Internet: http://dl.acm.org

Smartphone OS Market Share, Q3 2014. [En línea] INTERNATIONAL DATA CORPORATION. 2015. [consultado 01 de Mayo de 2013]. Disponible en Internet: http://www.idc.com/prodserv/smartphone-os-market-share.jsp