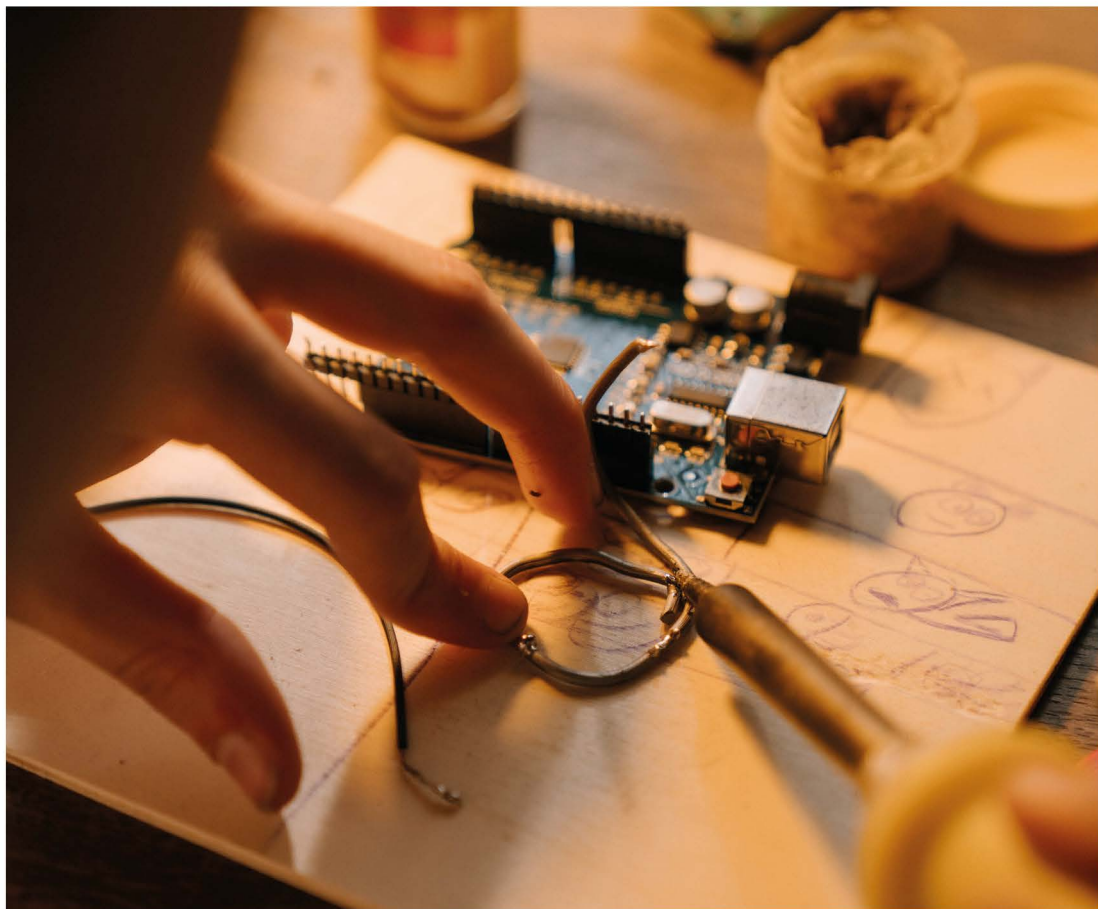


PRINCIPIOS DEL DISEÑO MECATRÓNICO



COLECCIÓN MATERIAL DOCENTE INGENIERÍAS



Diego Andrés Campo Ceballos. Docente programa de Ingeniería Mecatrónica de Unicomfauca de la ciudad de Popayán. Ingeniero Físico de la Universidad del Cauca y Magister en Sistemas Mecatrónicos de la Universidad de Brasília.

PRINCIPIOS DEL DISEÑO MECATRÓNICO

Diego Andrés Campo Ceballos



Catalogación en la publicación – Biblioteca Nacional de Colombia
Campo Ceballos, Diego Andrés.
Principios del diseño mecatrónico / Diego Andrés Campo Ceballos.
----1ª ed.--- Popayán: Sello editorial Unicomfacaucá, 2022.
p. 65
Contiene datos de los autores.

ISBN (Digital): 978-958-53187-7-9

Principios del diseño mecatrónico
© Corporación Universitaria Comfacaucá – Unicomfacaucá
© Autores: Diego Andrés Campo Ceballos

Primera edición en español
Sello Editorial Unicomfacaucá, septiembre de 2022
ISBN Digital: 978-958-53187-7-9

Diseño Editorial: Sello Editorial Unicomfacaucá – Corporación
Universitaria Comfacaucá - Unicomfacaucá
Corrección de Estilo: La Peregrina Estudio – larailustracion@gmail.com
Diagramación: La Peregrina Estudio – larailustracion@gmail.com
Editor General de Publicaciones: Julio Eduardo Mejía Manzano

Sello Editorial Unicomfacaucá
Calle 4ta # 8-30 Centro Histórico
Popayán, Colombia
Teléfono: 602 8386000 Ext. 118
www.unicomfacaucá.edu.co/investigacion/sello-editorial/



Licencia Atribución-NoComercial-SinDerivadas 2.5 Colombia (CC BY-NC-ND 2.5 CO)

Contenido

Resumen	9
Fundamentación teórica	10
· ¿Qué es la mecatrónica?	16
Fundamentación metodológica	16
Fundamentación curricular y didáctica	18
· Mecánica	18
· Control	18
· Electrónica	19
· Informática	19
· Robótica	19
· Electivas	19
Capítulo 1: Introducción al diseño mecatrónico: cómo construir tu propio brazo robótico	20
· Resumen	20
· ¿Cómo enfrentar un problema de ingeniería?	20
· Ejercicio práctico	22
· Ejercicios de evaluación	22
· Lecturas y sitios web complementarias	22
Capítulo 2: El principio de integración mecatrónico: brazos robóticos automatizados	23
· Resumen	23
· El diseño 3D, la electrónica y la programación	23
· Ejercicios prácticos	26
· Brazo robótico controlado por Arduino	26
· Mano robótica controlada por bluetooth	30
· Ejercicios de Evaluación	37
· Para el brazo robótico controlado por Arduino	37
· Para la mano robótica controlada por bluetooth	37
· Lecturas y consultas adicionales	37
Capítulo 3: La interfaz mecatrónica 1: el primer paso del prototipado	38
· Resumen	38
· El pseudocódigo y los diagramas de flujo	38
· Ejercicios prácticos	39
· Instrumentar el algoritmo en el cual se ingresan tres números reales y luego se calcula su suma, el promedio y el menor de los tres números	39
· Ejercicios de evaluación	48

Capítulo 4: La interfaz mecatrónica 2: herramientas Python para diseño	52
· Resumen	52
· Python	52
· Page y activetcl	52
· Ejercicios prácticos	53
· Ejercicios de evaluación	58
Glosario	60
Referencias	61

RESUMEN

Aunque el proceso de aceptación de la mecatrónica como campo de estudio e investigación en el mundo académico ha sido relativamente lento, en la actualidad se puede afirmar que ya ha alcanzado reconocimiento. “No es una nueva disciplina, sino un enfoque de aplicación de las últimas técnicas en ingeniería mecánica de precisión, teoría del control, ciencias computacionales y electrónica, al proceso de diseño, para la creación de productos funcionales, integrados y adaptables” (Universidad de Pamplona, 2011, p. 6). Es decir, innovadores. Así, se requiere comprender que la electrónica y el control computacional son componentes fundamentales del proceso de diseño, a la vez que se integran al proceso de fabricación digital. La formación del Ingeniero mecatrónico necesita estar fuertemente relacionada con los principios de diseño para atender las necesidades del sector productivo de la región, abordados desde una metodología concurrente que consolida la democratización, adaptación e integración de tecnologías a sus productos, procesos y servicios. Este libro es una herramienta inicial para generar productos, procesos y sistemas que integren tecnologías de la mecánica, la electrónica y la informática.

La guía aborda desde la construcción de un brazo robótico simple hasta el desarrollo de interfaces mecatrónicas simples, aplicando técnicas y métodos de diseño conocidos, fomentando el aprendizaje significativo y experiencial del estudiante. Se da especial atención al desarrollo de ejercicios para evaluar funcionalidad, apariencia, ahorro de energía, relación con el entorno, etc., propios de sistemas de alta complejidad tecnológica y que prometen solventar las necesidades del sector productivo del departamento del Cauca y Colombia.

Palabras clave: diseño mecatrónico, CAD, Arduino, Python, GUI.

INTRODUCCIÓN

La estructura curricular y el plan de estudios del programa de ingeniería mecatrónica, están orientados a la apropiación del conocimiento de las nuevas tendencias tecnológicas referidas a las diferentes áreas del conocimiento que exige, las cuales son una sinergia entre automatización, diseño mecatrónico, control, robótica, electrónica y sistemas informáticos. Además, fortalece el desarrollo humano, social y el emprendimiento mediante las respectivas cátedras orientadas para tal fin. (Universidad de Pamplona, 2011, p. 19)

Este texto tiene asociado un proceso de mejoramiento continuo y como producto de la autoevaluación se ajustó y actualizó a sus procesos académicos, también ha sido construido de acuerdo con los estándares para programas de educación superior. El compromiso de la corporación empieza con el fortalecimiento del programa y sus líneas de acción, que se reflejan con el apoyo a la publicación de material docente para la formación de cada uno de los estudiantes y de la comunidad en general y la constante modernización de sus equipos y espacios de laboratorio, que actualmente son el de electrónica, mecatrónica, automática, electromecánica, física, y su laboratorio de fabricación digital FabLab (Unicomfacaucá, 2018).

El perfil profesional del Ingeniero Mecatrónico de la Corporación Universitaria Comfacaucá se ha orientado a la solución de diversas necesidades y problemas de la región y en especial del sector productivo del departamento. Prevalece, como en la mayoría de sus pares a nivel nacional e internacional, una fuerte formación en el área del diseño de ingeniería, que permiten a los estudiantes abordar con gran destreza los temas propios de la carrera, siendo actores innovadores de procesos en pro de la sociedad en general y el sector productivo (Unicomfacaucá, 2018). El programa está fundamentado en seis líneas específicas: control de procesos, ingeniería sostenible, integración empresarial, materiales de ingeniería, robótica, visión de máquina e inteligencia artificial. Fomentando en los profesionales formados la capacidad de diseñar, desarrollar y adaptar soluciones innovadoras con visión integral, orientada al diseño de productos o procesos, para que puedan presentar prototipos y esquemas innovadores funcionales que resuelvan satisfactoriamente las problemáticas, sustentadas en el modelado fisicomatemático y apoyado en herramientas software de diseño y simulación.

Es así como esta guía proporciona las bases para que el estudiante del programa de Ingeniería Mecatrónica, realice una inmersión inicial a los procesos de diseño de ingeniería desde la perspectiva de la ingeniería mecatrónica y aborde satisfactoriamente los problemas locales, regionales y nacionales. Esto le permitirá convertirse en actor de la democratización de la tecnología en los campos de la rehabilitación médica, el desarrollo y soporte de servicios tecnológicos para la industria, el diseño de productos y servicios mecatrónicos, así como la generación de contenidos educativos, basados en la robótica y la automatización con un alto conocimiento de programación de las plataformas electrónicas más conocidas como Arduino.

Los temas abordados por el libro introducen al estudiante a que desarrolle actividades de diseño, basado en la metodología de diseño mecatrónico. Se inicia el recorrido por el mundo del diseño de ingeniería con la temática de “Cómo construir tu propio brazo robótico” y “El principio de integración mecatrónico”, gestando las bases de mecatrónica con las actividades de aprendizaje significativo en brazos robóticos automatizados. Después, se llega al estudio de

“La interfaz mecatrónica”, con el primer paso del prototipado y una etapa complementaria de herramientas Python para diseño mecatrónico. Con esta contribución se pretende fortalecer las actividades que realiza el programa de Ingeniería Mecatrónica de Unicomfauca en su evento semestral denominado *Showroom Mechatronics*, y se consolide como pilar fundamental de la formación de ingenieros innovadores para el Cauca y Colombia.

FUNDAMENTACIÓN TEÓRICA

En este libro es tan importante hacer referencia tanto a los componentes básicos del diseño mecatrónico, como a su contexto internacional que tiene que ver con los países y culturas que han adoptado estos conceptos para la potencialización de sus conocimientos y su aplicación en la solución de problemas de la vida cotidiana y la industria.

A nivel internacional, como referente es inevitable iniciar con Japón, teniendo en cuenta que fue el lugar en donde surgió el concepto de mecatrónica. La Universidad de Tokio es la más antigua y prestigiosa Universidad Nacional de Japón, se encuentra entre las primeras 20 universidades dentro del ranking mundial y su filosofía destaca la conjugación de la tradición cultural japonesa y el desarrollo de la ciencia y la ingeniería como herramientas para la creación de una nueva cultura, donde el sentido crítico de los individuos juega un papel importante. La Universidad cuenta con un Laboratorio de Mecatrónica Avanzada de gran prestigio, donde estudiantes y docentes realizan importantes aportes a la industria nacional e internacional, siendo el componente investigativo uno de los más importantes para el egresado. El programa de Mecatrónica hace parte del departamento de Ingeniería mecánica y las competencias del profesional se refieren al uso de actuadores, el manejo y manipulación de nuevas ingenierías, la interacción humano-computador/máquina, la bio-mecatrónica y las micro-máquinas.

Otro referente importante es Estados Unidos, donde se destaca el Instituto Tecnológico de Massachussets (MIT), la Universidad de Berkeley y la Universidad de Illinois, catalogadas dentro de las 25 más importantes desde el punto de vista investigativo. El MIT es uno de los más importantes referentes a nivel de docencia e investigación, tanto en Estados Unidos como a nivel mundial, pues es considerada una de las mejores universidades de ciencia e ingeniería del mundo. El MIT ofrece estudios en Mecatrónica dentro del departamento de Ingeniería Mecánica y cuenta con un Laboratorio de Investigación en Mecatrónica que se dedica al control, sistemas dinámicos y diseños desafiantes asociados a los campos de la nano-ingenería, bioingeniería y robótica.

La Universidad de Illinois, al igual que el MIT, cuenta con un Laboratorio en Mecatrónica que fue creado en 1995 con el objetivo de dar soporte a la educación basada en proyectos en Mecatrónica y sistemas embebidos en tiempo real. Así, hoy cuenta con ocho estaciones de trabajo completamente equipadas para el diseño, construcción y análisis de sistemas Mecatrónicos.

La Universidad de Berkeley es reconocida como la universidad pública más importante en los Estados Unidos y en el mundo. La Universidad ofrece el programa de Ingeniería Mecatrónica, el cual pertenece al Departamento de Ingeniería Mecánica. El programa de Mecatrónica involucra tanto actividades de enseñanza como de investigación con el objeto realizar actividades relacionadas con la ingeniería en el diseño y desarrollo de productos inteligentes.

En el caso de la comunidad europea, existe el Centro Europeo de Mecatrónica (European Centre for Mechatronics – APS) que opera en Aachen – Alemania. Este centro se ocupa de áreas como la robótica, sensores e ingeniería de la información y la comunicación. Los ingenieros y técnicos desarrollan su trabajo en cooperación con industrias nacionales e internacionales, instituciones públicas e investigadores, donde se busca apoyar a empresas pequeñas y medianas en la introducción y utilización de herramientas innovadoras de base tecnológica. Así, el centro se especializa en: técnicas de la automatización, técnicas de soldadura, Ingeniería de sensores, sistemas robóticos, comunicación hombre - máquina, modelado y simulación, servicio y diagnóstico, multimedia en la técnica de producción (APS, s.f.).

También existe el Centro de Investigación y Desarrollo para la Mecatrónica (Research and Development Center for Mechatronics – RDCM), el cual fue un proyecto del RDCM desarrollado con el auspicio del Consejo de Investigación de la Universidad Nacional (del Ministerio Rumano de Educación e Investigación) con fondos del Banco Mundial, para montar un centro de investigación multiusuario, donde los investigadores de otras universidades, institutos de investigación y compañías industriales pudieran usar equipos de altas prestaciones como los laboratorios de fabricación o FabLab, centros de mecanizado, caracterización e investigación en nuevos materiales, prototipado y diseño de circuitos PCBs, diseño asistido por computador, simulación mecánica y física. Desde el centro se promueve la colaboración científica internacional con 14 universidades de Europa y América, actualmente distribuidas así: Francia, Alemania, Italia, Polonia y Canadá (FabLab, s.f.).

En España existe la Mecatrónica como una especialidad de la Ingeniería Industrial, tal y como lo demuestra la oferta de la Universidad Politécnica de Cataluña. En esta Universidad existe el Centro de Diseño de Equipos Industriales, que es un centro de servicios de estrategia, innovación, desarrollo y optimización de bienes, equipos y productos que proporcionan ventajas competitivas a las empresas. Los estudiantes de Mecatrónica realizan sus prácticas en este laboratorio y proyectan soluciones empresariales adaptadas a necesidades puntuales del sector.

En el contexto latinoamericano, México ofrece un amplio espectro de Instituciones y Universidades donde las personas interesadas pueden estudiar Mecatrónica. La Universidad Nacional Autónoma de México-UNAM es una de las universidades con mayor reconocimiento académico en Iberoamérica. La UNAM creó el programa de Ingeniería Mecatrónica en el año 2003, de acuerdo con el Plan Nacional de Desarrollo 2012, donde señala que el desarrollo y aplicación de la Mecatrónica es fundamental para promover el desarrollo integral del país. En la Tabla 1, se muestra la oferta del programa Ingeniería Mecatrónica en Universidades de Latinoamérica. Con esto se muestra la importancia de conocer a fondo de que se trata la mecatrónica y como funciona para la solución de problemas de ingeniería

País	Universidad	Denominación Académica	Duración (semestres)
México	Instituto Tecnológico y de Estudios Superiores de Monterrey	Ingeniería Mecatrónica	9
	Universidad del Valle de México	Ingeniería Mecatrónica	9
Perú	Universidad Nacional de Ingeniería	Ingeniería Mecatrónica	10
	Universidad Peruana de Ciencias	Ingeniería Mecatrónica	10
	Universidad Tecnológica del Perú	Ingeniería Mecatrónica	10
Brasil	Universidad de Sao Paulo	Ingeniería Mecatrónica	10

Ecuador	Universidad Internacional del Ecuador	Ingeniería Mecatrónica	9
	Universidad Politécnica Salesiana	Ingeniería Mecatrónica	10
Argentina	Universidad Nacional de Cuyo	Ingeniería Mecatrónica	10
Bolivia	Universidad del Valle	Ingeniería Mecatrónica	10
	Universidad Católica Boliviana "San Pablo"	Ingeniería Mecatrónica	10
Venezuela	Universidad Nacional Experimental Politécnica Antonio José de Sucre	Ingeniería Mecatrónica	10
Chile	Universidad de Talca	Ingeniería Mecatrónica	10
Paraguay	Universidad Nacional de Asunción	Ingeniería Mecatrónica	10
Panamá	Universidad Latina de Panamá	Licenciatura en Ingeniería Mecatrónica	10
Costa Rica	Instituto Tecnológico de Costa Rica	Licenciatura en Ingeniería Mecatrónica	10
Nicaragua	Universidad Tecnológica La Salle	Ingeniería en Mecatrónica	10
Guatemala	Universidad del Valle de Guatemala	Ingeniería Mecatrónica	10
	Universidad Galileo	Ingeniería en Mecatrónica	8 ciclos
El Salvador	Universidad Don Bosco	Ingeniería Mecatrónica	10
Honduras	UNITEC	Ingeniería en Mecatrónica	10

Tabla 1

Oferta de Ingeniería Mecatrónica en universidades latinoamericanas

Fuente: Unicomfauca (2018).

En cuanto al panorama nacional, las universidades colombianas han realizado en los últimos años esfuerzos para elaborar una oferta pertinente y consecuente con las necesidades del país, de la mano con las políticas nacionales creadas como apoyo a la innovación y el desarrollo tecnológico, entre las que se encuentran los incentivos tributarios, los programas estratégicos de desarrollo tecnológico, el fomento a redes de innovación, el fomento al desarrollo de clúster, la integración de las cadenas de producción, la relación universidad- empresa-estado-sociedad y los programas de transferencia de Ingenierías. Con este panorama, la pertinencia de la oferta de los programas de ingeniería se constituye en un factor determinante de desarrollo y específicamente el programa de Ingeniería Mecatrónica, pues tiene como base el uso y apropiación de las nuevas tecnologías, los mecanismos mecatrónicos, la inteligencia artificial, el uso de las herramientas de la información y la comunicación y la gestión del trabajo empresarial interdisciplinario necesario en las empresas innovadoras que generen valor y competitividad en el mercado. En Colombia, es un hecho el crecimiento de los egresados en los últimos años. En la figura 1, se evidencian las áreas de conocimiento con más egresados del país. Los datos fueron obtenidos del sistema de información del observatorio laboral de la base de datos del mes de mayo de 2017 (Unicomfauca, 2018).

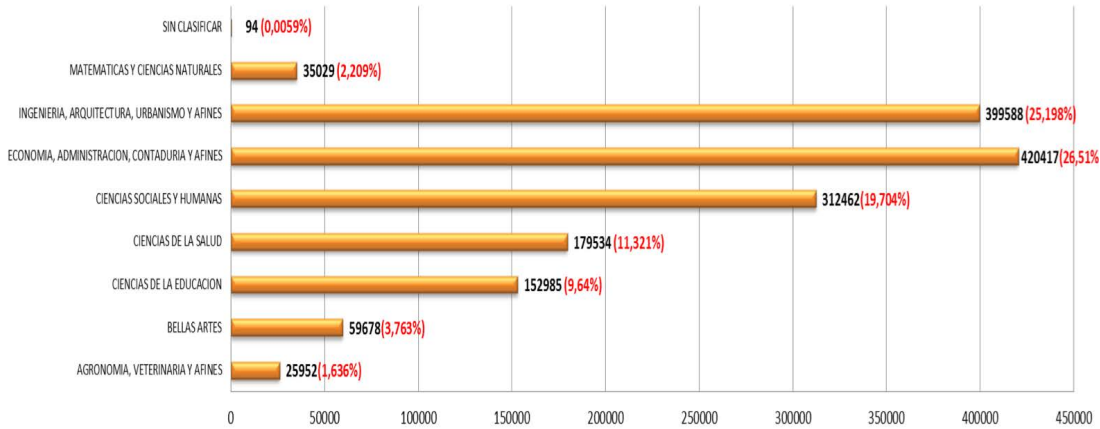


Figura 1
Egresados por áreas de conocimiento en Colombia
Fuente: Unicomfauca (2018).

Los programas pertenecientes al área de Ingeniería, Arquitectura y Urbanismo presentan un porcentaje de egresados del 25,19 % (399.588) del total de 1.585.739 egresados de las áreas de conocimiento más demandadas. El programa de Ingeniería Mecatrónica se encuentra en el área de formación académica de Ingeniería Mecánica y afines donde se cuenta con 27.772 egresados de los cuales el 3,22 % pertenecen al programa de Ingeniería en Mecatrónica y el 9,08 % pertenecen al programa de Ingeniería Mecatrónica, posicionándose en el segundo lugar con más egresados en esta área de conocimiento como se evidencia en la siguiente figura.

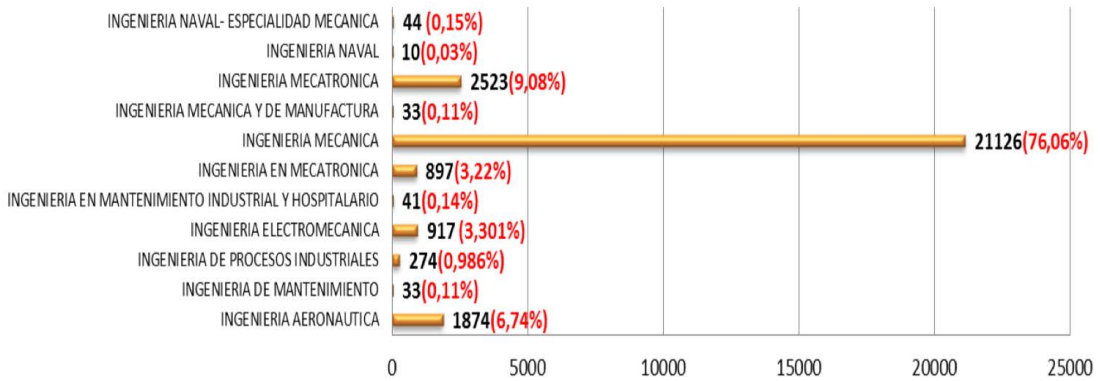


Figura 2
Egresados por áreas de conocimiento en Colombia
Fuente: Unicomfauca (2018).

A continuación, en la tabla 2, se relacionan las diferentes ciudades y Universidades donde actualmente se ofrece el programa Ingeniería Mecatrónica.

Ciudad	Universidad	Denominación Académica	Duración (semestres)
Envigado	Escuela de Ingeniería de Antioquia	Ingeniería Mecatrónica	10
Bogotá D.C.	Escuela Tecnológica Instituto Técnico Central	Ingeniería Mecatrónica Por ciclos	11
	Fundación Universitaria Agraria de Colombia	Ingeniería Mecatrónica	10
	Corporación Universidad Piloto de Colombia	Ingeniería Mecatrónica	10
	Universidad San Buenaventura	Ingeniería Mecatrónica	10
	Universidad Nacional de Colombia	Ingeniería Mecatrónica	10
	Universidad Militar Nueva Granada	Ingeniería en Mecatrónica	10
	Universidad ECCI (Escuela Colombiana De Carreras Industriales)	Ingeniería en Mecatrónica	10
Cali	Universidad Autónoma de Occidente	Ingeniería Mecatrónica	10
Bucaramanga	Universidad Autónoma de Bucaramanga	Ingeniería Mecatrónica	10
	Universidad Santo Tomas	Ingeniería Mecatrónica	10
Pamplona	Universidad de Pamplona	Ingeniería Mecatrónica	10
Manizales	Universidad de Caldas	Ingeniería Mecatrónica	10
Cartagena	Universidad Tecnológica de Bolívar	Ingeniería Mecatrónica	9
Barranquilla	Universidad Autónoma del Caribe	Ingeniería Mecatrónica	10
	Instituto Tecnológico Soledad Atlántico	Ingeniería Mecatrónica	10
Pereira	Universidad Tecnológica de Pereira	Ingeniería Mecatrónica Por ciclos	10
Medellín	Politécnico Colombiano Jaime Isaza Cadavid	Ingeniería Mecatrónica	10
	Instituto Tecnológico Metropolitano	Ingeniería Mecatrónica	10
Tunja	Universidad de Boyacá	Ingeniería Mecatrónica	10
Popayán	Corporación Universitaria Comfacauc	Ingeniería Mecatrónica	10

Tabla 2

Oferta de Ingeniería Mecatrónica en Colombia

Fuente: elaborado a partir de Unicomfacauc (2018) y ACOFI (s.f.).

Después de conocer este panorama, se hace aún más importante reconocer el protagonismo de la ingeniería mecatrónica como base tecnológica de progreso en la democratización de la tecnología, potencializando la ingeniería a nivel local, nacional y global. Como herramienta importante de mayor uso en esta rama de la ingeniería, se encuentran el diseño en ingeniería, que tiene que ver con propuestas metodológicas simples, para abordar los problemas, desde el punto de vista sistemático y basado en el método científico e ingenieril.

Un egresado de ingeniería mecatrónica tiene conocimiento en herramientas de diseño innovador, concurrente y didáctico, para enfrentar un problema y encaminarlo directamente con la adaptación e integración de soluciones innovadoras con visión de explotar todos los recursos de infraestructura tecnológica, generación de nuevos productos y el soporte para que un proyecto pueda desarrollarse con toda rigurosidad. Esta guía ofrece las bases para adquirir un aprendizaje significativo para similar los conceptos fundamentales de la mecatrónica y su aplicación en diferentes contextos.

¿Qué es la mecatrónica?

Muchos autores han definido la mecatrónica en el correr del tiempo. Sin embargo, una definición coherente es la que ofrece la revista MECHATRONICS, una revista científica de la IFAC, the International Federation of Automatic Control.

The Science of Intelligent Machines

Mechatronics is the synergistic combination of precision mechanical engineering, electronic control and systems thinking in the design of products and manufacturing processes. It relates to the design of systems, devices and products aimed at achieving an optimal balance between basic mechanical structure and its overall control. (Mechatronics, s.f.)

Es natural no reconocerla como ciencia, pues integra la definición de ingeniería original a la que se refería el más grande genio de la humanidad Leonardo Davinci, pero lo que sí es cierto que esta integradora y sinérgica rama de la ingeniería, ha estado promoviendo la competitividad en las empresas y potencia toda la capacidad innovadora en las organizaciones (Bishop, 2002).

FUNDAMENTACIÓN METODOLÓGICA

La metodología que sigue el presente documento fue diseñada con el enfoque mixto, basado en la metodología de diseño concurrente (Shetty & Kolk, 2011). Esta metodología presenta una serie de procesos que han sido agrupados en tres fases. Cada una de las fases cumple con el desarrollo de actividades específicas para cumplir con los objetivos del proyecto mecatrónico. La mecatrónica es la ciencia de las máquinas inteligentes, se define como la integración sinérgica entre la mecánica de precisión, la electrónica, la informática y el control para el diseño y fabricación de productos, servicios y procesos innovadores (Mechatronics, s.f.). Según lo anterior, este material docente surge de la necesidad de compilar una guía práctica y experiencial para que el estudiante inicie el proceso de diseño mecatrónico adoptando una metodología organizativa, o también llamada diseño concurrente, compuesta de tres fases: modelado y simulación, prototipado, y despliegue y ciclo de vida (Shetty & Kolk, 2011).

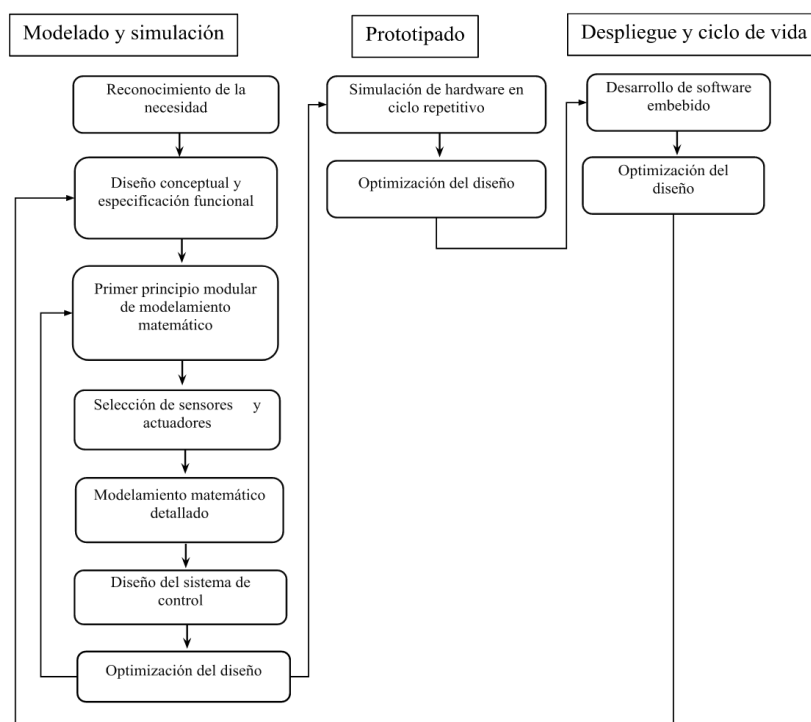


Figura 3
Metodología del diseño mecatrónico
Fuente: elaborado a partir de Shetty & Kolk (2011, p. 7).

Siguiendo a Shetty y Kolk (2011), en cada una de estas fases se encuentran subsistemas, en los cuales, se desarrollan actividades que pretenden que el estudiante se sumerja en todos los elementos que intervienen en un abordaje de diseño mecatrónico. Desde el reconocimiento de la necesidad, idea o concepto inicial, hasta obtener el producto final, pasando por el desarrollo del diseño conceptual, especificación funcional, análisis matemáticos, instrumental requerido, sistemas de control, y software embebido de manera simultánea, asegurando un producto de excelente calidad, con facilidad de mantenimiento y satisfacción de la necesidad del cliente.

Por otro lado, sin duda, es vital que el estudiante maneje los conceptos básicos del dibujo asistido por computador. Se trata de un curso inicial del programa de ingeniería mecatrónica que se ha considerado un aliado importante en el proceso diseño mecatrónico. Consiste en el uso de un paquete computacional especializado que ayuda a realizar dibujos, usando un procesador gráfico vectorial y volumétrico, dando paso a la obtención de planos y objetos en tres dimensiones, permitiendo crear, modificar, analizar y documentar toda la información de la representación gráfica del objeto físico. El software de dibujo es muy utilizado actualmente en áreas como: dibujo vectorial, dibujo de planos, dibujo en tres dimensiones, simulación, animación, renderización y efectos especiales, sobre todo para la presentación de prototipos de productos para la industria. En esta guía se mostrarán algunos diseños construidos en TINKERCAD, plataforma en línea de la casa Autodesk, que permite empezar con el mundo del diseño mecánico, electrónico y programación (Tinkercard, s.f.).

Como caso de estudio, se presentará el uso de TINKERCAD tanto para el modelado mecánico e impresión 3D, como para el diseño y simulación electrónica en Arduino y otros elementos pasivos que contemplan el aprendizaje didáctico con resistencias, capacitores, fuentes de poder. Una de las aplicaciones enfocadas en el campo ingenieril (Sloan, 2015), más fácil y versátil de usar a la hora de diseñar los sistemas de control electrónicos, dándole vida así a innumerables diseños que van desde los mecanismos cotidianos pero modernos, hasta productos innovadores que permiten mejorar nuestra calidad de vida y la competitividad del sector médico, de servicios, sustentable y de la industria en general (Öchsner& Altenbach, 2019).

FUNDAMENTACIÓN CURRICULAR Y DIDÁCTICA

La formación en Ingeniería aplicada contempla las áreas específicas de la formación del ingeniero mecatrónico; tiene en cuenta aspectos generales de la profesión y aspectos específicos del quehacer en cada ciclo de formación. Están orientadas a habilitar al profesional para desarrollar funciones productivas propias de su ocupación. Las actividades curriculares de ingeniería aplicada en las áreas de mecánica, informática, electrónica, control, robótica y asignaturas electivas aportan al ingeniero las herramientas conceptuales y profesionales para el diseño y desarrollo. Estas actividades curriculares hacen del ingeniero mecatrónico un profesional acorde con las necesidades tecnológicas del país y de la región y le permiten conjugar de forma armoniosa diferentes disciplinas que aportan soluciones innovadoras (Bradley *et al.*, 2015).

En este texto se podrán ver elementos de los siguientes campos del conocimiento, los cuales se integran sinérgicamente con la labor de diseño de nuestros ingenieros mecatrónicos y del quehacer de la ingeniería (Bolton, 2013).

Mecánica

Este componente se especializa en los fundamentos teóricos y prácticos de los procesos de fabricación y mecánica de precisión: piezas, elementos y mecanismos. Los estudiantes estarán en capacidad de comprender los principios que rigen el diseño, planificación y optimización de procesos de manufactura. Las actividades de investigación se realizan en los laboratorios de electromecánica y sala de sistemas para diseño en un software CAD (Inventor, SolidWorks, ANSYS, FreCAD, TinkerCad, entre otros). Las asignaturas correspondientes a esta área de formación son: Introducción a la ingeniería, Procesos de manufactura, CNC (control numérico computarizado), Diseño mecatrónico I, Diseño mecatrónico II, siguiendo los planteamientos de Billingsley (2006), Norton (2006) y Gentle *et al.*, (2001).

Control

Este componente promueve y desarrolla la investigación con los fundamentos basados en el modelamiento de sistemas eléctricos, mecánicos, hidráulicos y térmicos. Con el fin de brindar herramientas para el diseño de sistemas de automatización de líneas de producción. Las asignaturas correspondientes a esta área de formación son: Instrumentación industrial, PLC (controladores lógicos programables), SCADA (sistemas de supervisión y adquisición de datos), Sistemas físicos, Control automático y Sistemas neumáticos e hidráulicos (Bolton, 2006).

Electrónica

Esta área tecnológica estudia las características básicas y las propiedades de los elementos electrónicos fundamentales con los que puede manipularse la corriente eléctrica de acuerdo con la teoría de circuitos y dispositivos electrónicos, con el fin de desarrollar e integrar sistemas electrónicos, ya sea para el mejoramiento de sistemas existentes o para el desarrollo de nuevas tecnologías. Las asignaturas correspondientes a esta área de formación son: Circuitos eléctricos, Electrónica analógica I, Electrónica analógica II, Electrónica digital, Microcontroladores, Máquinas eléctricas (Bishop, 2005).

Informática

Este componente brinda la formación conceptual y las metodologías de desarrollo de habilidades en los aspectos relacionados con programación estructurada y programación orientada a objetos con el fin de desarrollar soluciones a problemas básicos de ingeniería, mediante los entornos de desarrollo y los lenguajes de programación. Las asignaturas correspondientes a esta área de formación son: Fundamentos de programación y Programación orientada a objetos (Bolton, 2015).

Robótica

Este componente tecnológico fomenta las competencias de modelamiento matemático y el diseño mecánico y eléctrico de robots. Al finalizar el estudio de esta área los estudiantes desarrollan un prototipo que cumple con los requisitos iniciales de funcionamiento, innovación y versatilidad para desarrollar tareas asignadas de forma supervisada o autónoma. La asignatura correspondiente a esta área de formación es Robótica (Rizk & Awad, 2019).

Electivas

Este componente permite al estudiante profundizar en campos de acción de la ingeniería mecatrónica. El programa ofrece dos líneas de profundización que son: Electiva en robótica, visión de máquina e inteligencia artificial y la Electiva en ingeniería sostenible. Además de esto los estudiantes cuentan con una actividad de extensión tal como la práctica empresarial y con una actividad de emprendimiento de base tecnológica, derivada de procesos de investigación aplicada, a través de la asignatura Opción de grado, la cual es obligatoria para obtener el título de Ingeniero mecatrónico. Las asignaturas correspondientes a esta área de formación son: Electiva de profundización I, Electiva de profundización II, Electiva de profundización III, Electiva de profundización IV, Práctica empresarial, Opción de grado.

Como se observa, el diseño mecatrónico es transversal en la formación del ingeniero y constituye una de las herramientas que debe estar presente en el ámbito del diseño de ingeniería como insumo para iniciar y crear una conciencia de la importante tarea de un ingeniero para abordar los problemas con la cultura innovadora Unicomfacaucana.

CAPÍTULO 1:

Introducción al diseño mecatrónico: cómo construir tu propio brazo robótico

Resumen

Considerando la oferta de cursos que existen para promover y afianzar la enseñanza de la ingeniería en todo el mundo, el Instituto de Ingeniería Eléctrica y Electrónica- IEEE (s.f.), la asociación más grande del mundo en ingeniería y tecnología, consideró apropiado abrir una serie de planes didácticos que tienen que ver con el diseño concurrente. Uno de ellos consiste en presentar al estudiante un problema, el esquema de preparación para su planteamiento, los materiales, las restricciones y los resultados a obtener.

De esta forma se muestra cómo se puede iniciar en el abordaje del diseño mecatrónico a través de un problema sencillo que se plantea a nivel mundial, con la comunidad IEEE. Este trabajo se ha desarrollado consecutivamente durante 6 años en el programa de ingeniería mecatrónica de Unicomfauca y constituye la base para entender el proceso de diseño en ingeniería. Los resultados han sido satisfactorios y potencian las habilidades de los estudiantes para desplegar un plan para el desarrollo de productos mecatrónicos, basados en el método ingenieril, generar prototipos, exponerlos y presentar reportes escritos en formato LATEX (Borbón y Mora, 2016) (procesador de textos para alta calidad tipográfica).

¿Cómo enfrentar un problema de ingeniería?

Como fundamento de ingeniería se debe empezar a analizar el problema, luego se diseñan las estrategias para abordarlo y finalmente se implementa para obtener un resultado. Este proceso a grandes rasgos tiene segmentos que no se tienen en cuenta, por lo que muchas veces el proceso de análisis se vuelve interminable. Es aquí donde es recomendable que se inicie con el principio de observación que presenta el método científico, “observar a su alrededor”. La tarea ahora es darse cuenta de si realmente se tiene un verdadero problema para resolver. En este caso lo primero que se hará será conocer la técnica para resolver problemas, que además de ser aplicada a la ingeniería, también se puede implementar a la vida cotidiana. En ingeniería, existe un procedimiento general para resolver un problema (Freddi & Salmon, 2019), detallado y sintetizado de la siguiente forma.

1. Formulación del problema: una pregunta que inicia con un por qué, cómo, cuándo, de qué manera, etc., y que delimita la situación. No debe admitir un “sí” o “no” como respuesta.
2. Análisis del problema: se contemplan los recursos que existen para dar respuesta a la pregunta formulada, se consideran antecedentes, marcos teóricos, referentes recientes y previos.
3. Búsqueda de soluciones creativas: se usa la lluvia de ideas, los mapas mentales, los mapas conceptuales para agrupar conceptos y teorías, que resultan de una investigación y búsqueda sistemática de la literatura y la experiencia y se agrupan.
4. Decisión: del grupo de alternativas de solución, se evalúan las que sean óptimas, pero a la vez realistas y concretas. Fáciles de implementar pero que sean presupuestalmente válidas.

5. Reporte ingenieril: la solución elegida se detalla por escrito con planos, modelos, esquemas, bocetos, ecuaciones, gráficos de desempeño y descripción de las pruebas finales y presentación de los resultados.

Este procedimiento se muestra de forma gráfica en la figura 4. Todo inicia con la fase 1: la planeación del producto que se requiere, se identifica la necesidad, se establecen las especificaciones del sistema, se generan, seleccionan y prueban conceptos y finalmente surge un primer prototipo. Luego se pasa a una fase 2, donde se diseña teniendo en cuenta la integración del sistema, es decir, el diseño detallado y se logra un segundo prototipo. En una tercera fase se confronta la evolución del prototipo con el ajuste del sistema de control y se establece su funcionalidad. Si el prototipo es óptimo pasará a la etapa de prototipado rápido, optimización, documentación, registro y escalado a nivel industrial, dado el caso. Si no, se debe volver a la generación, selección y prueba de conceptos que no se tuvieron en cuenta, y así continuar con el proceso de desarrollo. La vida útil de un producto es un ciclo, por eso ningún producto mecatrónico estará totalmente terminado, hay que recordar que las alternativas de solución seleccionadas son apenas aproximaciones cercanas a la solución óptima y siempre habrá espacio para mejorarlas confrontándolas con la optimización de su ciclo de vida. En ese recorrido se encontrarán restricciones o características de una solución que se fijan previamente por una decisión, por la naturaleza del problema, por requisitos legales, de responsabilidad social, ambiental o ética o por cualquier otra disposición (Shetty & Kolk, 2011; Bradley et al., 2015; Freddi & Salmon, 2019).

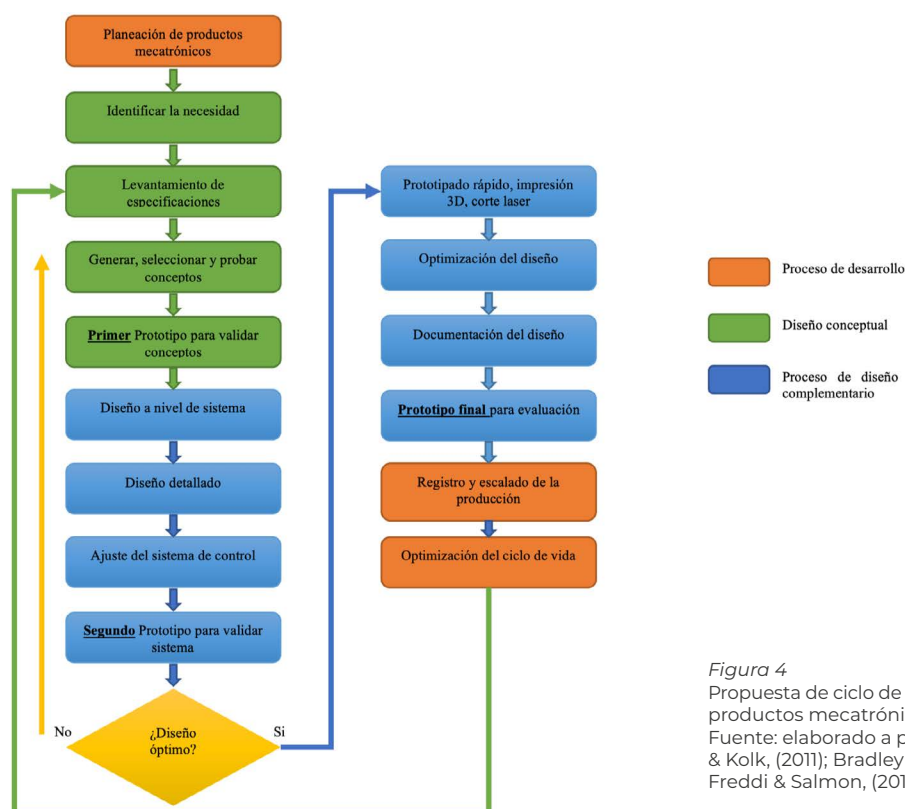


Figura 4
Propuesta de ciclo de desarrollo para productos mecatrónicos.
Fuente: elaborado a partir de Shetty & Kolk, (2011); Bradley et al., (2015); Freddi & Salmon, (2019).

Ejercicio práctico

Trabajando en equipo, los estudiantes exploran la efectividad del trabajo en grupo mientras aprenden el mecanismo de funcionamiento de robots simples usando materiales comunes. Esta es una estrategia adoptada por la IEEE para desarrollar competencias para la apropiación de la ingeniería en el mundo. Sin duda, es la base fundamental para lograr aprendizaje significativo en los estudiantes.

- i. Aprender sobre el diseño tecnológico
- ii. Uso de cálculos matemáticos para el diseño
- iii. Aprender sobre el movimiento y la fuerza
- iv. Practicar habilidades de comunicación a través de ejercicios de exposición y escritos en formato IEEE LATEX (Borbón y Mora, 2016; IEEE, 2019).

El desafío consiste en construir un brazo robótico que pueda recoger un vaso de plástico de 7 onzas lleno de agua, a una distancia de 50 cm. Levantarlo a una altura mínima de 20 cm, dejar el vaso en su posición original y soltarlo. El operador de brazo no puede entrar en contacto con el vaso de agua.

Para tal fin se implementan las siguientes restricciones didácticas. Solamente se pueden usar los siguientes materiales:

- Palitos de madera
- Alambre
- Tachuelas
- Hilo
- Clips
- Bandas elásticas
- Cinta
- Ganchos de ropa
- Lápices
- Cajas de cartón
- Broches para papel
- Nylon

Ejercicios de Evaluación

1. ¿Cuál parte fue la más crítica para la construcción de su brazo robot?
2. ¿Como ayudó el trabajo en equipo en el proceso de diseño?
3. ¿Que aprendió de los diseños desarrollados por otros equipos?
4. ¿Como podría ser usada esta actividad en la vida cotidiana?

Lecturas y sitios web complementarias

- TryEngineering (www.tryengineering.org)
- “Humans and Robots” NASA Educational Brief, 2001. (er.jsc.nasa.gov/seh/Robot_PDF_Files/humans_and_robots.pdf)
- FIRST Robotics Competition (www.firstinspires.org/robotics/frc)
- Robot Books (www.robotbooks.com)

CAPÍTULO 2:

El principio de integración mecatrónico: brazos robóticos automatizados

Resumen

En el campo de la ingeniería existen muchos caminos por recorrer, y si bien existe una metodología de resolución de problemas, esta logra ser mucho más versátiles e innovadoras entre más herramientas tecnológicas tengan a su alrededor (Shetty & Kolk, 2011; De Silva, 2005). En este capítulo se mostrará cómo se integran las herramientas de diseño asistido por computador, diseño electrónico, programación y control, de una mano robótica con el fin de automatizar los movimientos de los dedos. También se mostrará el desarrollo de un brazo robótico automatizado con Arduino para ejercer acciones de *pick-and-place* o manipulación de elementos y generar un primer producto mecatrónico completo. Se presta mucha atención al levantamiento de requerimientos de cada producto y a la integración didáctica de las plataformas trabajadas. Tinkercad, Arduino, Overleaf, MIT-APPinventor y Thingiverse son algunas de las plataformas usadas para desarrollar los proyectos. El estudiante alcanzará un nivel máximo de entendimiento en el área de integración y manejo de los elementos básicos de la mecatrónica con la fabricación de estos prototipos en un tiempo destinado de aproximadamente 4 meses (1 semestre académico).

El diseño 3D, la electrónica y la programación

La plataforma de Tinkercad (www.tinkercad.com), de la casa Autodesk, es una aplicación gratuita y fácil de usar de diseño 3D, electrónica y creación de código. Es usada por su versatilidad por diseñadores para imaginar, diseñar y crear casi cualquier cosa. Unido a esto, como herramienta fundamental para trabajar con el diseño, se ha generado un espacio de fabricación digital en la Corporación Universitaria Comfacauca, conocido mundialmente como FABLAB. En este espacio se dispone de las siguientes máquinas para fabricación:

- Máquina cortadora laser: En esta máquina se pueden hacer cortes y gravados a materiales blandos, como madera, aglomerados, sintéticos como acrílico, teflón entre otros. El principio de funcionamiento consiste en el uso de un haz de luz laser, concentrado ópticamente, cuya potencia es distribuida y utilizada según el nivel de profundidad de trabajo, su velocidad y el material trabajado. Particularmente se usa el software RDworks 8.0, el cual traduce los trazos de un dibujo 2D de formato DXF-*Drawing Exchange Format*, a cortes sistemáticos que generan la forma deseada.

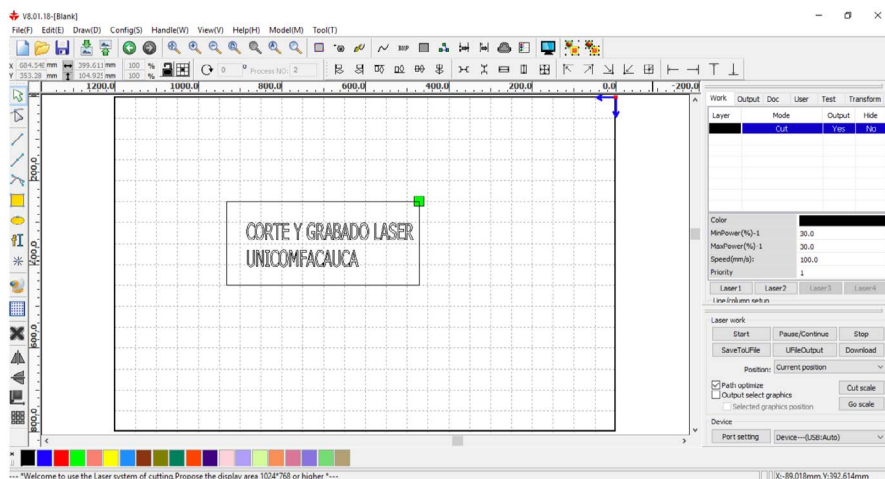


Figura 5

Software de diseño y lectura archivos DXF- para la máquina de corte láser

Fuente: elaboración propia.

· Máquina de impresión 3D. Estas impresoras son usadas para depositar capas muy delgadas de un material polimérico PLA o ABS entre otros materiales, a través de una ruta fijada por un diseño computarizado en tres dimensiones x, y, z y un archivo de impresión *.Gcode. Así, los objetos creados a partir de esta técnica generalmente requieren de un modelado estricto y el tiempo de fabricación depende de la velocidad de la máquina. A velocidades rápidas, menos calidad. A velocidades lentas mayor calidad, pero mayor tiempo de impresión.

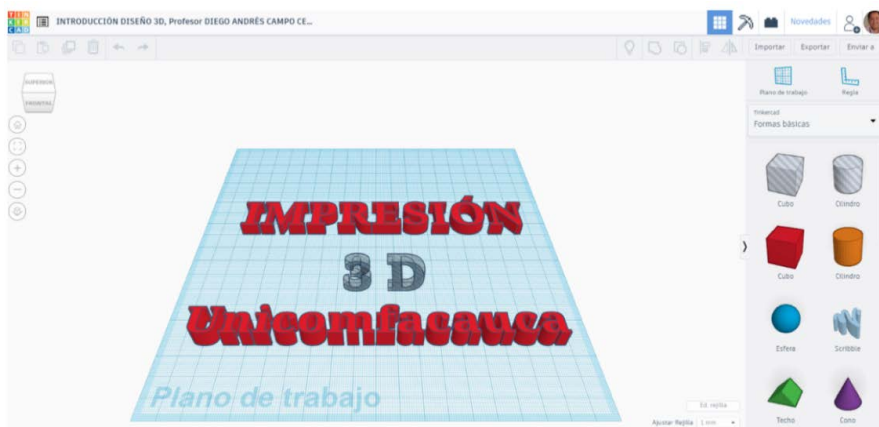


Figura 6

Introducción a la plataforma tinkercad 3D

Fuente: elaboración propia.

Ambas herramientas requieren un previo manejo del dibujo asistido por computador, que para este caso será la plataforma Tinkercad y su integración con Arduino.

Por otro lado, el diseño electrónico se puede realizar con Arduino, que provee toda una gama de plataformas electrónicas de desarrollo con aplicación inmediata para la etapa del primer prototipado que el estudiante está por abordar. Para efectos de simulación e integración real se contempla el uso del Arduino uno, con su microcontrolador AVR ATMEGA328. Como se observa en la figura, es una plataforma conformada por un cable de alimentación, que puede ser USB, o un adaptador de 5V-2A para uso en potencia.

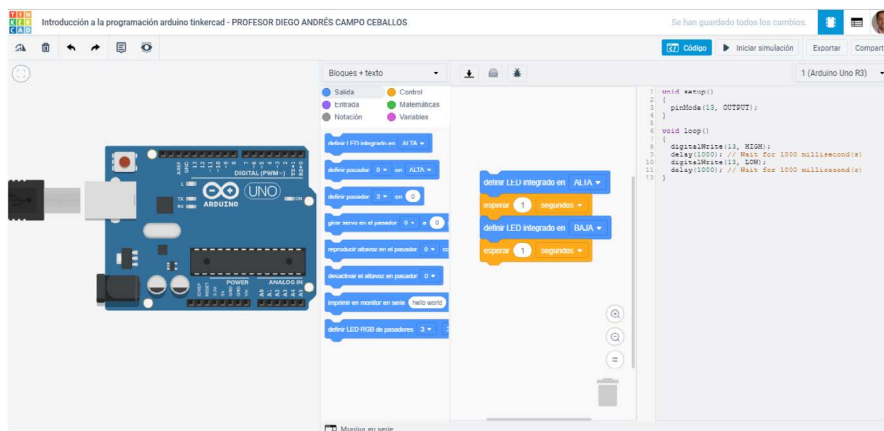


Figura 7

Plataforma Tinkercad para trabajar con Arduino uno. Modo bloque+texto

Fuente: elaboración propia.

En este caso existen 6 entradas analógicas, 14 pines digitales, con la particularidad de que el pin 13 tiene conectado un led de prueba. 2 pines de comunicación serial Rx-Tx y los niveles lógicos de conexión a 5 voltios,VCC, GND. La tolerancia máxima del voltaje de entrada Vin es de 7-20 voltios. Corriente DC por Pin de E/S: 20 mA. Es necesario conocer todo esto ya que será vital para el proceso de diseño electrónico.

Arduino se puede programar por bloques o por texto. Por bloques, se invoca una herramienta de entrenamiento y enseñanza muy conocido como el appinventor del MIT (s.f. a). Esta integración ha sido desarrollada para fines de enseñanza de la programación en niños, jóvenes y adultos que aún no han entrado en contacto con un lenguaje de programación. Por otro lado, se puede programar normalmente como en un entorno de desarrollo IDE de texto, mediante lenguaje C y C++ modificado de Arduino. Para efectos de enseñanza aprendizaje se recomienda iniciar con ambos, bloque+texto, para estudiar cómo se codifican los bloques en un lenguaje de programación, de forma automática.

Ejercicios prácticos

Brazo robótico controlado por Arduino

Inicialmente se plantea el ejercicio de fabricar un brazo robótico simple y fácil de modificar, para que la obtención de las piezas no sea tan demorada. Los planos de este brazo robótico propuesto fueron diseñados por MeArm Robotic Labs (www.mearm.com/), que desde 2014 inició con un proyecto código abierto (opensource) para fabricar unos pequeños brazos robóticos capaces de implementar múltiples tareas. La idea de los creadores fue brindar al mundo una herramienta funcional, económica y libre para experimentar con mecanismos y electrónica básica, para automatizar los movimientos y acciones en las tareas específicas que se requieren. El diseño puede ser fabricado por varias técnicas, sin embargo, son dos las preferidas, el corte láser y la impresión 3D. Estimando el tiempo de obtención de las piezas se puede pensar en usar la cortadora láser, o imprimir las piezas en una impresora 3D. En cursos posteriores se puede profundizar más en estos temas, pero seguro que siempre un corte láser demorará alrededor de 30 minutos cortando estas piezas, mientras que la impresora 3D, demoraría entre una hora y media y dos horas, aproximadamente. En las figuras se observan los dos métodos de fabricación que se pueden usar para la fabricación de las piezas de este proyecto. Sin embargo, en la impresión 3D se corre el riesgo de tener que usar herramientas mecánicas para ajustar los orificios de los tornillos y de los empalmes, ya que es inevitable la extrusión de material plástico no estructural dentro de ellos.

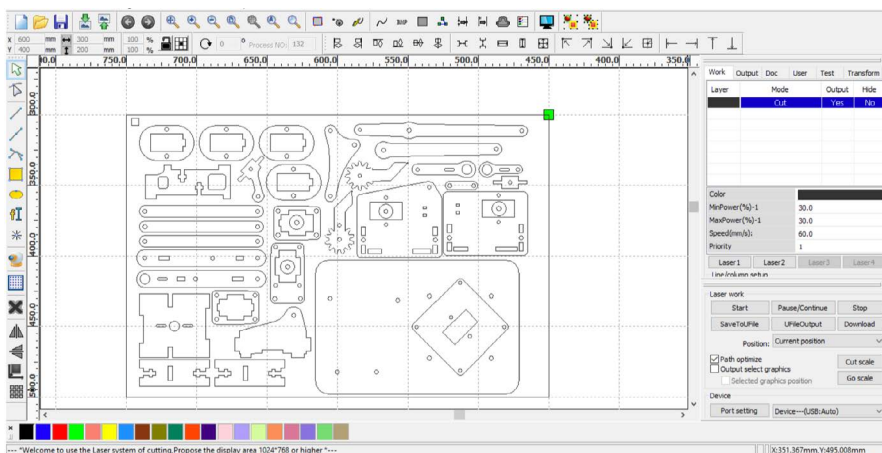


Figura 8

Fabricación de las piezas del brazo robótico por corte láser
Fuente: elaboración propia.

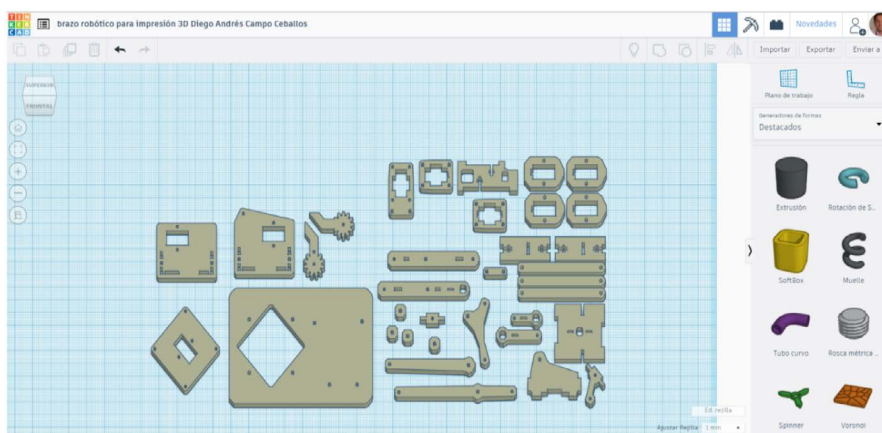


Figura 9

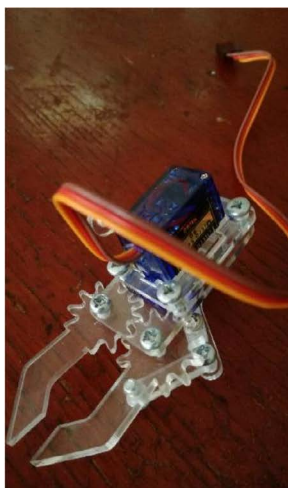
Fabricación de las piezas del brazo robótico con impresión 3D

Fuente: elaboración propia.

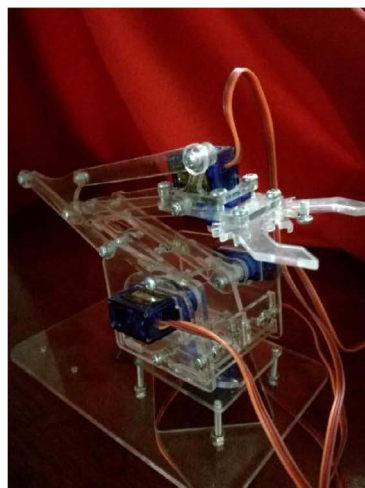
Una vez obtenidas las piezas se debe armar el cuerpo del brazo, a partir de la secuencia de armado desde la página de Instructables (www.instructables.com/id/Pocket-Sized-Robot-Arm-meArm-V04/) realizado por Phenoptix (s.f.), teniendo en cuenta que, en el diseño mecánico, para armar la pinza, figura 10b, se debe tener físicamente el servomotor junto con las piezas que lo conforman. El resto de piezas solamente se unen con los tornillos y los motores del brazo y hombro, simplemente se deben atornillar a su respectivo lugar. Los cuerpos de los robots deberán tener la siguiente apariencia.



a.



b.



c.

Figura 10

Prototipo

*Nota: a) prototipo en acrílico cortado en láser, b) detalles de pinza, c) armado final

Fuente: elaboración propia.

A continuación, se procede a introducir el Arduino como plataforma de control de los servomotores. Cada servomotor va a ser controlado por un canal PWM del Arduino, con el cual se va poder dar instrucciones del ángulo necesario para lograr un movimiento deseado. Los servomotores usados tienen la ventaja de solo limitarse a ángulos entre 0-180 grados. Sin embargo, deben ser calibrados antes de fijarse en cada uno de los mecanismos. El proceso de calibración puede ser manual o energizando los servomotores. En todos los casos debe ocurrir lo siguiente:

- La pinza debe abrir 180 y cerrar en 0 grados .
- El brazo debe bajar 180 y debe subir a 0 grados. No debe haber rozamiento entre mecanismos. Asegurarse de insertar las arandelas de separación si esto ocurre.
- El hombro debe ayudar a girar al máximo la pinza en 180 y a dejarla paralela al suelo en 0 grados.
- La base debe hacer girar todo el brazo robótico 180 grados en la dirección deseada.

Teniendo en cuenta el completo entendimiento de la calibración para evitar que los servomotores giren libremente, se ajustan con los tornillos destinados para tal fin y se procede a la fase de simulación y posterior programación del Arduino para el control de los servomotores a través de potenciómetros, para las funciones destinadas como se observa en la siguiente figura.

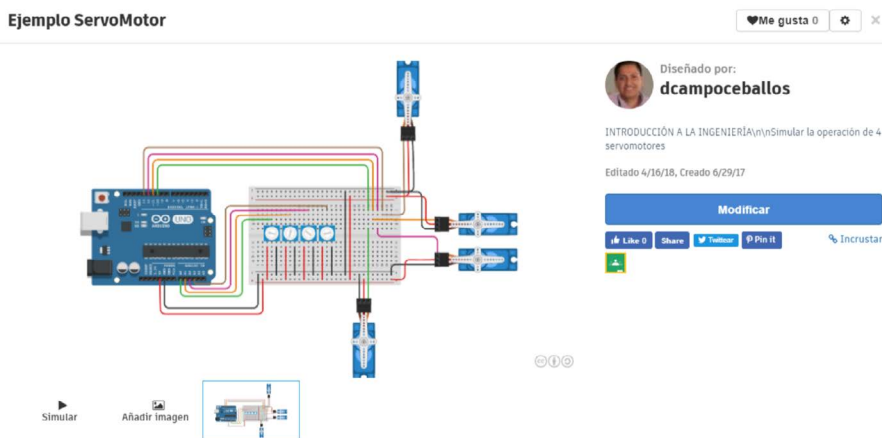


Figura 11

Conexiones físicas del sistema de control del brazo robótico usando potenciómetros
Fuente: elaboración propia.

En este punto es de vital importancia reconocer que cada potenciómetro es usado para enviar una señal analógica a los puertos A0, A1, A2 y A3 del Arduino uno. Esta señal de 0-5 V va a ser recibida por el conversor analógico-digital (ADC), para luego ser codificada por software a señales de 0-180 grados respectivamente. Igualmente, la señal de control será enviada a través de los puertos digitales PWM, 10,11,12,13 para el pin de control de cada servomotor. Así, cada vez que se gire un potenciómetro se moverá el servomotor correspondiente conectado a esa señal de salida.

A continuación, se detalla más a fondo este procedimiento en el código introducido en el Arduino.

```

Servo Servobase; //instancias u objetos de la clase Servo. Para crear el servo de la base o giratoria
Servo Servoderecha; //instancias u objetos de la clase Servo. Para crear el servo de la derecha o brazo
Servo Servoizquierda; //instancias u objetos de la clase Servo. Para crear el servo de la izquierda u hombro
Servo Servopinza; //instancias u objetos de la clase Servo. Para crear el servo de la pinza

int potenciometro1 = 0; //El pin analógico A0 donde va conectado el Potenciómetro 1
int potenciometro2 = 1; //El pin analógico A1 donde va conectado el Potenciómetro 2
int potenciometro3 = 2; //El pin analógico A2 donde va conectado el Potenciómetro 3
int potenciometro4 = 3; //El pin analógico A3 donde va conectado el Potenciómetro 4

int val1; //El valor a leer por el canal 0 del ADC
int val2; //El valor a leer por el canal 1 del ADC
int val3; //El valor a leer por el canal 2 del ADC
int val4; //El valor a leer por el canal 3 del ADC

void setup() // en este segmento se configuran los servomotores al PWM
{
  Servopinza.attach(13); //El pin de salida al Servo PINZA. Cuidado con este pin, pues con arduino tiene un led conectado.
  Servoderecha.attach(12); //El pin de salida al Servo DERECHA
  Servoizquierda.attach(11); //El pin de salida al Servo IZQUIERDO
  Servobase.attach(10); //El pin de salida al Servo BASE
}

void loop() //en este segmento se ejecuta continuamente el código inmerso
{
  // como el ADC del Arduino uno es de 10 bits podemos tener valores de 2 exp 10 = 1024 valores
  val1= analogRead(potenciometro1); //se lee el valor del potenciómetro, oscila entre 0 y 1023, es decir 1024 valores
  val1= map(val1, 0,1023, 0, 180); //Traduce la lectura analógica (0, 1023) a grados (0°, 180°)
  Servopinza.write(val1); //Mueve el Servo pinza según los grados correspondientes
  delay(15); //genera un retardo de 15 milisegundos para evitar cola de datos en el ADC

  val2= analogRead(potenciometro2); //se lee el valor del potenciómetro 2, oscila entre 0 y 1023, es decir 1024 valores
  val2= map(val2, 0,1023, 0, 180); //Traduce la lectura analógica (0, 1023) a grados (0°, 180°)
  Servoderecha.write(val2); //Mueve el Servo derecho según los grados correspondientes
  delay(15); //genera un retardo de 15 milisegundos para evitar cola de datos en el ADC

  val3= analogRead(potenciometro3); //se lee el valor del potenciómetro 3, oscila entre 0 y 1023, es decir 1024 valores
  val3= map(val3, 0,1023, 0, 180); //Traduce la lectura analógica (0, 1023) a grados (0°, 180°)
  Servoizquierda.write(val3); //Mueve el Servo izquierdo según los grados correspondientes
  delay(15); //genera un retardo de 15 milisegundos para evitar cola de datos en el ADC

  val4= analogRead(potenciometro4); //se lee el valor del potenciómetro 4, oscila entre 0 y 1023, es decir 1024 valores
  val4= map(val4, 0,1023, 0, 180); //Traduce la lectura analógica (0, 1023) a grados (0°, 180°)
  Servobase.write(val4); //Mueve el Servo base según los grados correspondientes
  delay(15); //genera un retardo de 15 milisegundos para evitar cola de datos en el ADC
}

```

La siguiente etapa es automatizar el movimiento de los potenciómetros, a través de un módulo bluetooth, es decir control inalámbrico, retirar por completo el uso de potenciómetros y convertirlo en un proyecto mecatrónico integrado. En este paso se utiliza la plataforma de desarrollo de MIT app inventor, (www.ai2.appinventor.mit.edu/) que permite crear aplicaciones para los teléfonos y tabletas Android.

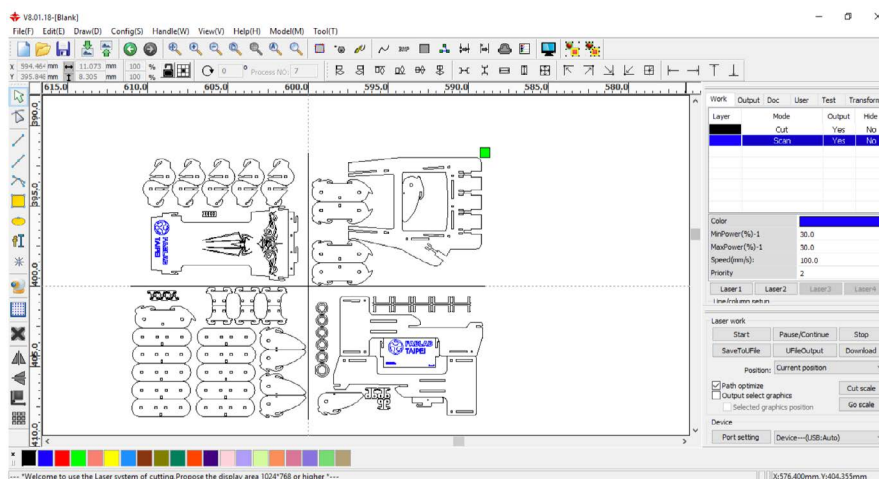


Figura 12

Prototipo de aplicación que controla el brazo robótico.

Fuente: elaborado a partir de MIT (s.f.).

Cada esquema creado para un dispositivo móvil o Tablet, podrá hacer uso de uno de sus herramientas de comunicación más versátiles, como lo es el módulo integrado bluetooth. Sin embargo, para que esta comunicación pueda existir, del lado del Arduino se debe adicionar un módulo compatible con el estándar bluetooth, para este caso se puede usar el HC-05 o HC-06.

La aplicación está lista cuando se programa la búsqueda, identificación y apareo o vinculación con un dispositivo bluetooth cercano, una vez se hace el vínculo entre módulos, se procede a enviar la información y ya físicamente los puertos de comunicación TX y RX del Arduino, son los que reciben dichos parámetros. Por eso, el código del Arduino debe coincidir con lo programado en la aplicación, para que se ejecute el comando que se desea. Este último diseño se deja como ejercicio, dado que existen muchas formas de desarrollarlo.

Mano robótica controlada por bluetooth

A través de tiempo siempre han existido operaciones manuales riesgosas, en las cuales se ven comprometidas partes de los cuerpos de los operarios. En ese sentido se identifica la necesidad de desarrollar una mano robótica, que permita estudiar los movimientos de cada dedo y poder simular estos movimientos e implementar acciones y estrategias de control. La mano robótica es un diseño abierto (*open source*) del FABLAB TAIPE, que permite observar cómo se realizan los movimientos de los dedos en la mano humana de una forma muy didáctica y con un diseño muy cercano a un exoesqueleto para la mano (www.instructables.com/id/Laser-cut-Mechanical-Claw/) realizado por Musicandsky (s.f.). Este diseño es fácil de modificar y ensamblar por los dos métodos descritos anteriormente. Cortadora laser o impresión 3D. Eventualmente la impresión de piezas crea la expectativa de mayor calidad y textura a las piezas; sin embargo, el tiempo destinado a la impresión completa de las piezas es muy largo comparado con el corte láser. En la práctica resulta de 4 horas de diferencia.

Por este motivo se recomienda que para la etapa de prototipado básico se trabaje con corte láser. La impresión 3D puede ser usada una vez se optimice el diseño y se realicen todos los ajustes que el grupo de trabajo decida.

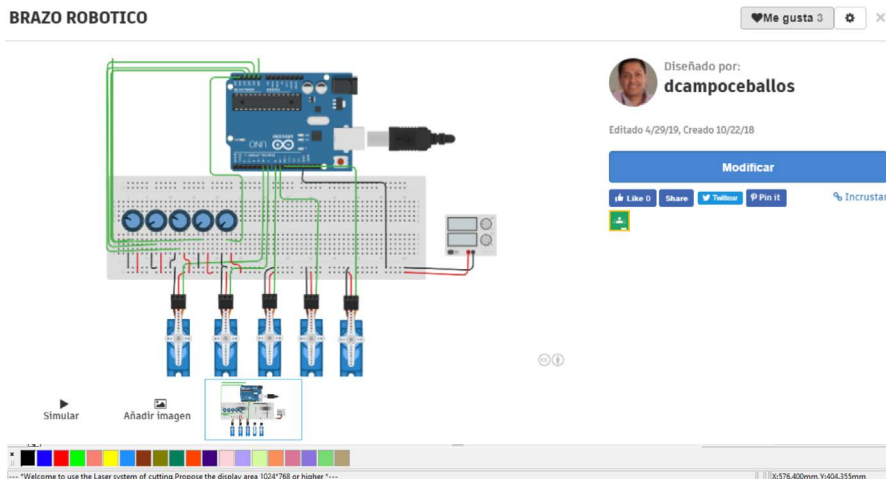


Figura 13

Corte laser de proyecto mano robótica, MDF(madera aglomerada) 3mm espesor.
Fuente: elaboración propia.

En el tema de control, por tratarse de un exoesqueleto, se requeriría controlar todos los dedos de la mano. Así, que ya con la experiencia inicial en tinkercad, se procede a montar el diseño del circuito con 5 servomotores para este nuevo proyecto, tal y como se observa en el gráfico.

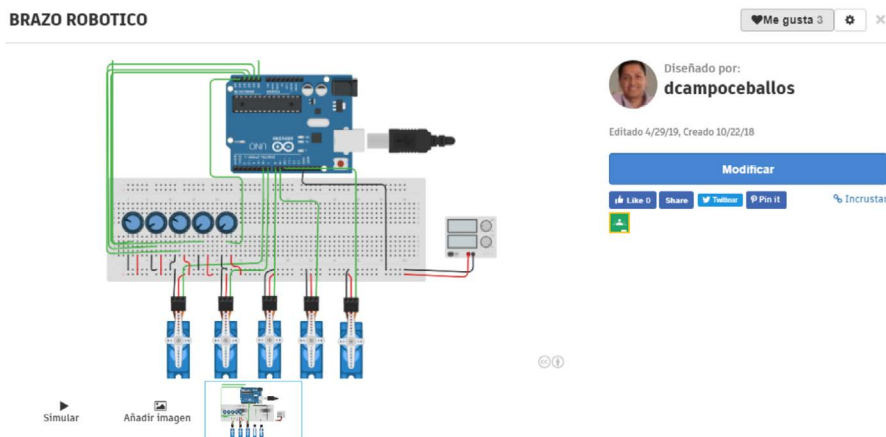


Figura 14

Conexiones físicas del sistema de control de la mano robótica usando potenciómetros
Fuente: elaboración propia.

El sistema es muy parecido electrónicamente hablando, al observado en la figura 11; sin embargo, la adición de un quinto servomotor cambia sustancialmente las condiciones de diseño en potencia. Es así como se necesita adicionar una fuente externa de 5 voltios y 3A, que alimentará a los motores para su correcto funcionamiento. Así mismo, el diseño integral admite un cambio sustancial en la forma de controlar las señales de entrada, e incluir un elemento muy versátil para el control. Las barras deslizantes, serán los elementos explorados en este proyecto para enviar las señales simuladas por los potenciómetros. Estas permiten enviar valores entre 0 y 180 grados al Arduino, a través del protocolo bluetooth leídos por el puerto serial, evitando la implementación convencional de los potenciómetros. Como se observa en la figura 14, los potenciómetros y otros elementos pasivos, suelen usarse para simular entradas de señal, que luego serán reemplazadas por las entradas reales del sistema.

La aplicación diseñada para el dispositivo Android, será entonces modificada para usar las barras deslizantes que deben configurarse para enviar los datos necesarios para que en el programa Arduino, se gestione la lógica de control que requiere el estudiante. En particular, los movimientos estudiados serán el de flexo-extensión, tanto de cada uno de los dedos como de la palma de la mano. Se programará igualmente la rutina de pinza (pulgar - índice). Las otras rutinas se dejarán como actividades extra, dado que se abordan diversos problemas como el uso de lenguaje de señas, el agarre, la rehabilitación entre otros.

El prototipo de la aplicación se muestra en la figura 15. Se puede observar que se ha incluido también el módulo de búsqueda, identificación y vinculación de dispositivos bluetooth, llamado conectar. Hay que recordar que se debe emparejar el dispositivo bluetooth antes de presionar el botón conectar, debido a los permisos que se deben conceder para el sistema Android de cada celular en particular y que permitirá generar el envío y recepción de datos bajo este protocolo de comunicación inalámbrico. El prototipo de las funciones usa 5 variables D1-D5, referente a cada uno de los dedos de la mano.

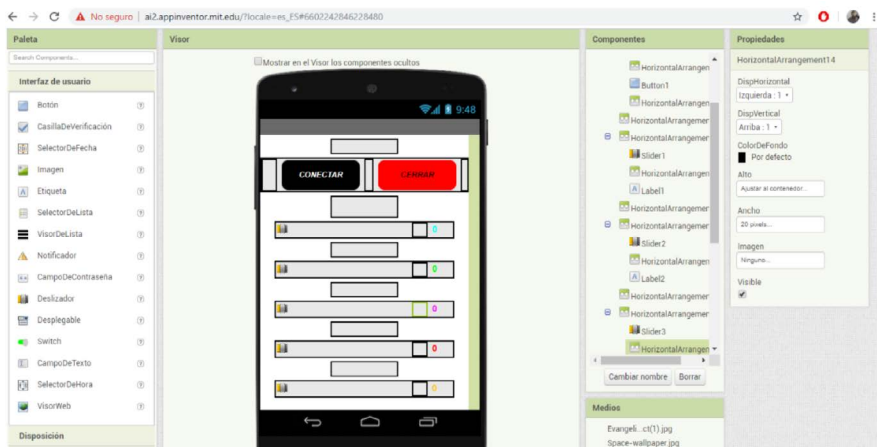
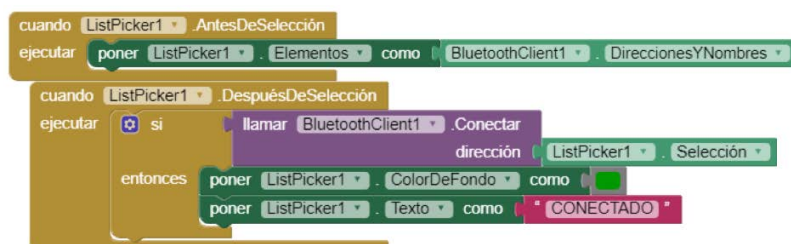


Figura 15

Prototipo de aplicación para controlar mano robótica, implementando barras deslizantes
Fuente: elaboración propia.

Aparentemente, incluir las herramientas de barras deslizantes en una aplicación de celular puede ser trivial. Sin embargo, implica un mayor conocimiento del envío de datos entre los dos sistemas y su programación tanto en APP-inventor como en el Arduino, tendrá un nivel de complejidad interesante en su implementación. En principio, se mostrará el diseño a nivel de integración de aplicación. En la figura 16, se observan los módulos que conforman cada una de las fases de programación. Por favor, no confundir la posición de cada barra deslizante o *thumb position*, con la posición del dedo pulgar. Se trata de una coincidencia de la traducción al español.



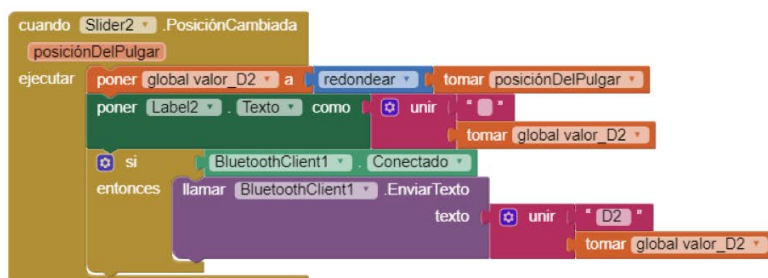
a. Módulo conexión bluetooth estándar.



b. Módulos de cerrar e inicializar variables de trabajo



c. Programación para la variable dedo1 D1.



d. Programación para la variable dedo 2, D2.



e. Programación para la variable dedo 3, D3.



f. Programación para la variable dedo 4, D4.



g. Programación para la variable dedo 5, D5.

Figura 16

Programación en bloques

*Nota: Programación en bloques para el Sistema de control de los dedos de la mano D1 a D5.

Fuente: elaboración propia.

En lo que respecta a la programación de Arduino, se deben considerar dos puntos.

1. Recibir un dato por el puerto serial, RX, TX, habilita el uso de la librería para trabajar con el bus de datos a una velocidad de 9600 bps. Cada cadena es de 8 bits.
2. Programar cada variable del Arduino para capturar el valor de la barra deslizante (valores 0-180 grados) correspondiente, asignado a cada salida del PWM donde se encuentra conectado cada servomotor.

Para aclarar más este tema, se recomienda instalar el aplicativo APK en los celulares, y así reconocer como se establece la comunicación entre el módulo HC-05 o HC-06 y el celular Android. Además, se debe estudiar el siguiente código de Arduino, diseñado para recibir la información de la aplicación del celular vía HC-05 o HC-06, conectado al puerto serial RX-TX del Arduino.

```
#include <SoftwareSerial.h> // libreria para el manejo de los datos seriales
#include <Servo.h> // libreria para el manejo de los servomotores
Servo pulgar, indice, corazon, anular, menhique;

// recordar que la conexión serial debe ser cruzada entre dispositivos RX con TX y viceversa
int bluetoothTx = 7; //define el pin del arduino para conectar el RX del HC-05 o HC-06
int bluetoothRx = 8; //define el pin del arduino para conectar el TX del HC-05 o HC-06

SoftwareSerial bluetooth(bluetoothTx, bluetoothRx); //configura la variable bluetooth para realizar la comunicación.
//constructor de la clase serial

void setup()
{
  pulgar.attach(9);
  indice.attach(10);
  corazon.attach(11);
  anular.attach(3);
  menhique.attach(5);
  //se configura la velocidad estándar de comunicación serial
  Serial.begin(9600);

  //se configura la velocidad estándar de comunicación de los datos recibidos desde Android
  bluetooth.begin(9600);
}

void loop()
{
  //lee desde el bluetooth y escribe en el puerto serial
  if(bluetooth.available() > 0){
    unsigned int servopos = bluetooth.read();
    unsigned int servopos1 = bluetooth.read();
    unsigned int realservo = (servopos1 * 256) + servopos;
    Serial.println(realservo); // posición actual de la barra deslizadora en el android

    //secuencia de activación del dedo pulgar
    if (realservo >= 1000 && realservo < 1180){
      int servo1 = realservo;
      servo1 = map(servo1, 1000, 1180, 0, 180);
      pulgar.write(servo1);
      Serial.println("pulgar activado");
      delay(10);
    }
  }
}
```

```

//secuencia de activacion del dedo indice
if (real servo >= 2000 && real servo < 2180){
  int servo2 = real servo;
  servo2 = map(servo2, 2000, 2180, 0, 180);
  indice.write(servo2);
  Serial.println("indice activado");
  delay(10);
}

//secuencia de activacion del dedo corazon
if (real servo >= 3000 && real servo < 3180){
  int servo3 = real servo;
  servo3 = map(servo3, 3000, 3180, 0, 180);
  corazon.write(servo3);
  Serial.println("corazon activado");
  delay(10);
}

//secuencia de activación del dedo anular
if (real servo >= 4000 && real servo < 4180){
  int servo4 = real servo;
  servo4 = map(servo4, 4000, 4180, 0, 180);
  anular.write(servo4);
  Serial.println("anular activado");
  delay(10);
}

//secuencia de activación del dedo meñique
if (real servo >= 5000 && real servo < 5180){
  int servo5 = real servo;
  servo5 = map(servo5, 5000, 5180, 0, 180);
  menhique.write(servo5);
  Serial.println("menhique activado");
  delay(10);
}

```

Cada barra deslizante debe ser programada con los valores descritos en el código descrito anteriormente, ya que los valores mínimo y máximo son los únicos que determinan la funcionalidad o activación de cada uno de los motores y su respectiva asignación a los dedos de la mano robótica. Es decir, para el dedo pulgar se ha designado valores de 1000 a 1180, como se observa en la figura 17.



Figura 17

Detalle de diseño para la barra deslizadora del APPinventor, respecto al código Arduino.
Fuente: elaboración a partir de Appinventor (MIT, s.f. b) y Arduino (s.f.).

Consecutivamente, cada barra deslizante puede configurarse con los valores consignados en el código Arduino o con un valor diferente del propuesto, siempre y cuando tengan un rango de 180 y las señales se puedan diferenciar una de la otra, en el bus de datos serial, con los niveles asignados, tanto en el código de Arduino como en la aplicación móvil.

Ejercicios de Evaluación

Para el brazo robótico controlado por Arduino

Desarrollar el código del Arduino para escuchar los controles de la aplicación desarrollada en el APPinventor. Recordar este segmento de código, tanto en Arduino como en APPinventor.

```
void setup() {
  Serial.begin(9600);
  pinMode(12,OUTPUT);
}
void loop() {
  respuesta = Serial.read();
  if (respuesta=='a')
  {
    digitalWrite(12,HIGH);
  }
  if (respuesta=='b')
  {
    digitalWrite(12,LOW);
  }
  delay(1);
}
```

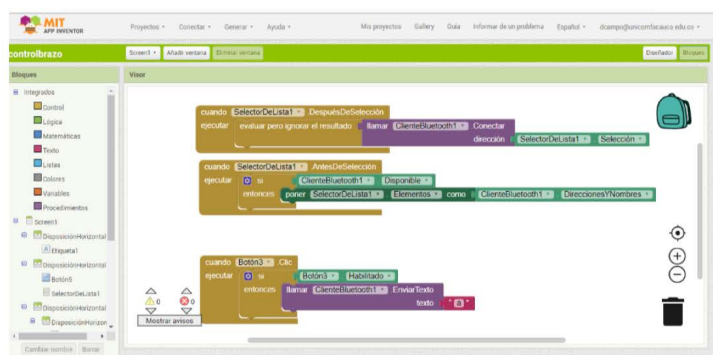


Figura 18
Ejercicio propuesto Bluetooth Arduino
Fuente: elaboración propia.

Para la mano robótica controlada por bluetooth

Diseñar una aplicación em MIT-APPinventor que sea capaz de ejecutar una secuencia automática de movimientos en los dedos, correspondiente a:

- Contar con los dedos del 1 al 5.
- Abrir y cerrar la mano 5 veces consecutivas Cada una de estas tareas debe ser programada y se debe ejecutar cuando se presiona un botón en la aplicación.

Lecturas y consultas adicionales

- MeArm Calibration: www.codebender.cc/sketch:148456#MeArm_Calibration.ino
- MeArm Robot: www.instructables.com/id/MeArm-Robot-Arm-Your-Robot-V10/
- Laser-cut mechanical claw: www.instructables.com/id/Laser-cut-Mechanical-Claw/
- App inventor: www.appinventor.mit.edu/explore/ai2/tutorials

CAPÍTULO 3:

La interfaz mecatrónica 1: el primer paso del prototipado

Resumen

La instrumentación electrónica siempre ha estado disponible para diseñar y desarrollar cualquier tipo de proyecto electrónico. Sin embargo, existen brechas que deben ser cerradas para garantizar el soporte, desempeño o el diseño de nuevos productos. Una de las brechas iniciales que se tienen en ingeniería es poder interconectar el software con el hardware, como estrategia de desarrollo de producto (Onwubolu, 2005). En este capítulo se desarrollará toda la secuencia procedimental para lograr una cohesión máxima entre software y hardware, iniciando desde el pseudocódigo, pasando por la elección de instrumentos pasivos, activos o plataformas electrónicas, hasta su diseño ergonómico final. El estudiante va a ser capaz de diseñar desde cero, con base en un problema cotidiano, de implementar interfaces de software y hardware para que el usuario final pueda disfrutar cómodamente de ellos.

El pseudocódigo y los diagramas de flujo

Cada vez que se trata de resolver problemas de ingeniería, es necesario contar con procedimientos sistemáticos que provean soluciones factibles y fáciles de implementar. Sin embargo, a veces lleva mucho tiempo brindar una solución apropiada tangible o instrumentada electrónicamente. El pseudocódigo es quizá la herramienta más poderosa para empezar con esta interesante tarea. Pero existen factores que no permiten pasar a un siguiente plano de diseño, como por ejemplo el desconocimiento de instrumentación electrónica, sensores, plataformas electrónicas, elementos de conversión de señales analógicas o digitales, interfaces de hardware o software entre otros (Braga, 2009).

Un sistema está conformado por las entradas, un bloque de proceso y una o varias salidas, dependiendo de la tipología. Técnicamente esto lleva a que se dude de que elemento real puede servir para instrumentalizar un algoritmo.

Un teclado matricial es un dispositivo que agrupa varios elementos llamados botones, que permiten enviar una señal codificada o no de forma individual. Esta señal suele ser un valor de voltaje o corriente que después un microcontrolador puede decodificar. Por ejemplo, el teclado del computador se conoce como una interfaz humano computador y permite, a través del código internacional ASCII, ingresar datos a un sistema informático. Así mismo, las pantallas son elementos tecnológicos que permiten visualizar los datos o la información suministrada. A cada uno de estos elementos se les conoce como interfaces software-hardware, y han estado con nosotros desde la era de la informática (Braga, 2009).

La idea, entonces, es conocer más afondo el funcionamiento tanto de los teclados como fuente de datos de entrada, como de las pantallas como elementos de visualización para el usuario. Al final, el estudiante podrá reconocer que los diagramas de flujo y su pseudocódigo reflejan la necesidad de ser modelados como instrumentos electrónicos tangibles y ser ejecutados en entornos reales.

Ejercicios prácticos

Con el fin de conocer el funcionamiento básico de un teclado y una pantalla LCD, se han organizado los siguientes ejercicios.

Instrumentar el algoritmo en el cual se ingresan tres números reales y luego se calcula su suma, el promedio y el menor de los tres números

Para dar respuesta a este ejercicio se asume el uso del diagrama de flujo como herramienta para analizar el problema y tener un primer prototipo software.

El prototipo del algoritmo de la calculadora permite hacer la suma, promediar y saber cuál es el número menor.

SUMA = $n1 + n2 + n3$

PROMEDIO = $(n1 + n2 + n3)/3$

MENOR ($n1 < n2 < n3$) sino ($n2 < n1 < n3$) o sino ($n3 < n2 < n1$)

Con esta lógica y operaciones matemáticas básicas, se logra construir un esquema de instrumentación a partir de las propias operaciones del diagrama de flujo y de su visualización.

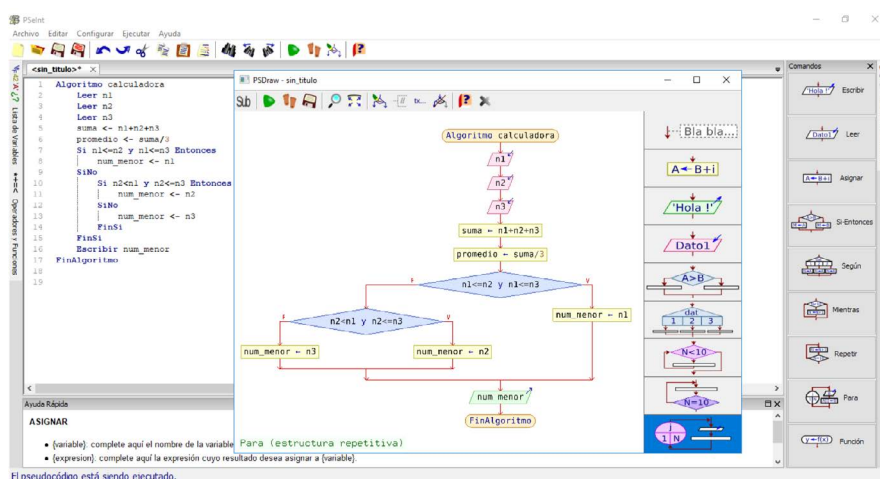


Figura 19

Diagrama de flujo del ejercicio propuesto implementado en software PSEINT
Fuente: elaboración propia.

Teniendo en cuenta la figura 19, y lo que significa cada uno de los diagramas de flujo, se puede iniciar un análisis comparativo para proyectar la instrumentación de este ejercicio. La captura de datos se puede hacer con un teclado matricial de membrana, las operaciones las deberá hacer un microprocesador y la visualización de los datos las debe contener una pantalla LCD. La primera forma de este algoritmo en hardware se muestra en la figura 20.

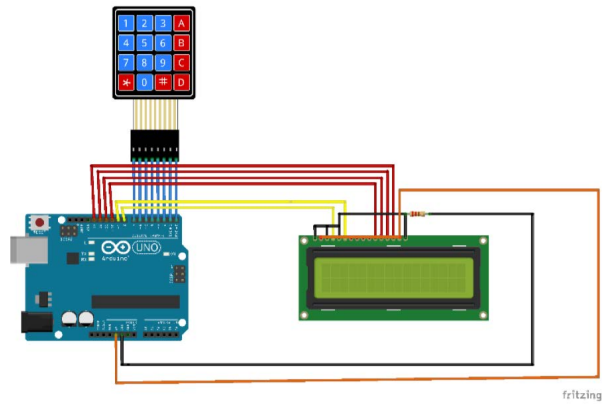


Figura 20
Instrumentación electrónica del diagrama de flujo
Fuente: elaboración propia.

Ahora, al final este ejercicio se encuentra en proceso de codificación. Arduino, el lenguaje basado en C++ (Braga, 2009), podrá implementarse para dar vida a la operación que se desea realizar. En este caso se trata de un diseño en tinkercad, cuyo código se le da al estudiante para que realice las operaciones de conexión y compruebe su funcionalidad.

```
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
#define row1 7
#define row2 6
#define row3 5
#define row4 4
#define col1 3
#define col2 2
#define col3 1
#define col4 0
```

```
LiquidCrystal_I2C lcd(0x27,16,2);
float num1,num2,num3,suma=0,n,i=0,menor,promedio;
```

```
void setup(){
    lcd.init();
    lcd.backlight(); // Activar luz de fondo
    lcd.clear(); // Borrar LCD
```

```
pinMode(row1, INPUT_PULLUP);
pinMode(row2, INPUT_PULLUP);
pinMode(row3, INPUT_PULLUP);
pinMode(row4, INPUT_PULLUP);
pinMode(col1, OUTPUT);
digitalWrite(col1, HIGH);
```

```

pinMode(col2, OUTPUT);
digitalWrite(col2, HIGH);
pinMode(col3, OUTPUT);
digitalWrite(col3, HIGH);
pinMode(col4, OUTPUT);
digitalWrite(col4, HIGH);
lcd.begin(16,2);
lcd.print("por favor ingrese");
lcd.setCursor(0,1);
lcd.print("3 numeros:");

}

char scan_keyboard(void)
{
    static int rows[] = {row1, row2, row3, row4};
    static int cols[] = {col1, col2, col3, col4};
    static char teclas[4][4] = { { '1', '2', '3', 'A' }, { '4', '5', '6', 'B' }, { '7', '8', '9', 'C' }, { '*', '0', '#', 'D' } };
    int row, col;
    int result = '\0';

    for (col = 0; col < 4 && result == '\0'; ++col){
        digitalWrite(cols[col], LOW);
        for (row = 3; row >= 0; --row){
            if (digitalRead(rows[row]) == LOW){
                result = teclas[row][col]; break;
            }
        }
        digitalWrite(cols[col], HIGH);
    }

    return result;
}

void loop()
{
    switch (scan_keyboard())
    {
        case '1': n=1;break;
        case '2': n=2;break;
        case '3': n=3;break;
        case '4': n=4;break;
        case '5': n=5;break;
        case '6': n=6;break;
        case '7': n=7;break;
        case '8': n=8;break;
        case '9': n=9;break;
        case '0': n=0;break;
        case 'A': num1=n;
            lcd.clear();
            lcd.print("numero 1:");
            lcd.setCursor(12,0);
            lcd.print(num1);break;
        case 'B': num2=n;
            lcd.clear();
            lcd.print("numero 2:");
            lcd.setCursor(13,0);
            lcd.print(num2);break;
    }
}

```

```

        case 'C': num3=n;
            lcd.clear();
            lcd.print("numero 3:");
            lcd.setCursor(13,0);
            lcd.print(num3);break;
        case '*':
            suma=num1+num2+num3;
            promedio=suma/3;
            if(num1 < num2 && num1 < num3)
            {
                menor=num1;
            }
        else { if(num2 < num1 && num2 < num3)
            {
                menor=num2;
            }
            else {    menor=num3;
                    }
            }

        lcd. clear ();
        lcd.print("suma:");
        lcd.setCursor(0,1);
        lcd.print(num1);
        lcd.setCursor(3,1);
        lcd.print("+");
        lcd.setCursor(5,1);
        lcd.print(num2);
        lcd.setCursor(9,1);
        lcd.print("+");
        lcd.setCursor(10,1);
        lcd.print(num3);
        lcd.setCursor(13,0);
        lcd.print(suma);
        delay (5000);
        lcd. clear ();
        lcd.print("promedio:");
        lcd.setCursor(13,0);
        lcd.print(promedio);
        delay (2000);
        lcd. clear ();
        lcd.print("Menor:");
        lcd.setCursor(13,0);
        lcd.print(menor); break;
    case '#': lcd.clear(); lcd.print("Ingrese n.");
        num1=num2=num3=0; break;
    case 'D':
        lcd. clear ();
        lcd.print("numeros:");
        lcd.setCursor(0,1);
        lcd.print(num1);
        lcd.setCursor(3,1);

```

```

    lcd.setCursor(3,1);
    lcd.print(" ");
    lcd.setCursor(5,1);
    lcd.print(num2);
    lcd.setCursor(9,1);
    lcd.print(" ");
    lcd.setCursor(10,1);
    lcd.print(num3);
    break;
default: break;
}
}

```

Cada línea de código está escrita para usar el sistema bajo el procedimiento descrito a continuación:

- Ingresa el primer número presionando en el teclado (1 al 9).
- Presionar el botón A para guardar ese número.
- Ingresa el segundo número presionando en el teclado (1 al 9).
- Presionar el botón B para guardar ese número.
- Ingresa el tercer número presionando en el teclado (1 al 9).
- Presionar el botón C para guardar ese número.
- Presiona el botón D para ver los números ingresados.
- Presiona * para ver suma, promedio y el número menor.
- Presiona # para reiniciar.

Los detalles del código se pueden reflejar en su lectura y denotan uno de los primeros acercamientos del software con el hardware. Alcanzar la experiencia en la conexión de la LCD, su arquitectura y sistema de presentación de caracteres, en conjunto con la programación de cada uno de sus aspectos integrado al sistema Arduino y sus periféricos, ya es un resultado tangible de la funcionalidad del sistema.

Se recomienda especial cuidado en el montaje e implementación real, dado que el simulador de tinkercad solo emula el procesamiento por medio de software. Se pueden encontrar conflictos de hardware, sobre todo en la dirección de memoria de la LCD, las conexiones de los bits de registro y las matrices simultaneas del teclado de membrana.

Finalmente, el propósito de este capítulo es ver que se debe apropiar cada elemento electrónico desde su arquitectura hasta su programación, pues se puede estar lejos de conseguir el montaje de un circuito funcional y autónomo, si se pierde la conexión entre los conceptos básicos de programación y cómo ellos pueden llegar a representarse en la realidad tangible. En la figura 21, se muestra un modelo del producto final implementado.

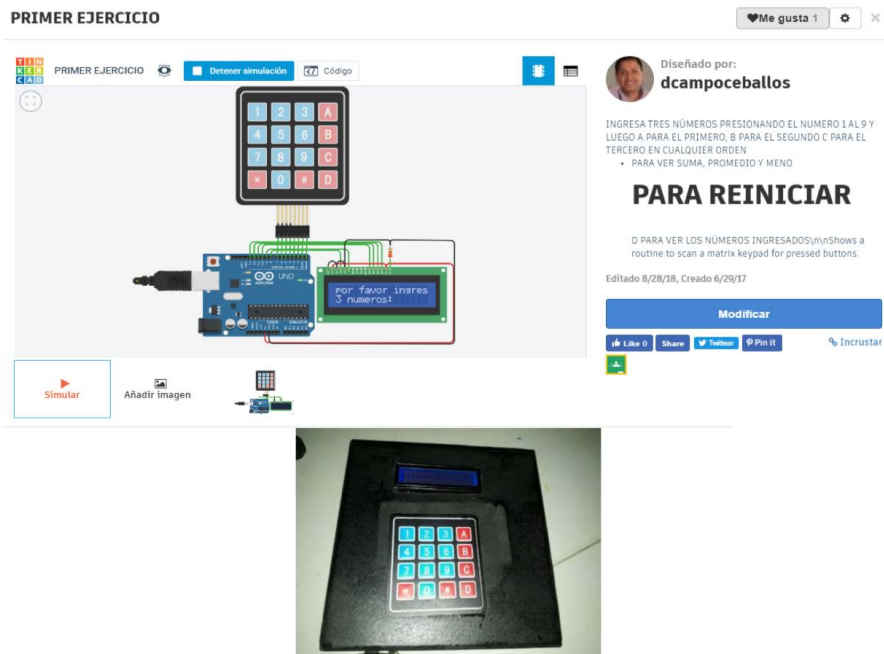


Figura 21

Simulación e instrumentación del ejercicio de la calculadora.

Fuente: elaboración propia.

a. Se requiere simular un sistema que funcione como una terminal bancaria. El saldo inicial es de 0.0 pesos. Se debe incluir acciones primarias de depósito y de retiro. Si el saldo es menor al valor de retiro, no se permite realizar la operación. En caso de depósito, el valor se sumará al saldo. Se permite realizar las acciones primarias cuantas veces quiera el usuario. Al final si desea salir, muestra el saldo total.

Para este caso, se toma el hardware ya creado y se escribe el algoritmo para después pasarlo a la simulación en el Arduino. Es muy importante tener en cuenta que se debe optimizar la funcionalidad de cada una de las teclas disponibles. A continuación, se presenta el código, para embeberlo en el arduino y comprobar su funcionalidad.

```
#include <Keypad.h>
```

```
#include <LiquidCrystal.h>
```

```
#define row1 7
```

```
#define row2 6
```

```
#define row3 5
```

```
#define row4 4
```

```
#define col1 3
```

```
#define col2 2
```

```
#define col3 1
```

```
#define col4 0
```

```
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
```

```

void setup(){
  pinMode(row1, INPUT_PULLUP);
  pinMode(row2, INPUT_PULLUP);
  pinMode(row3, INPUT_PULLUP);
  pinMode(row4, INPUT_PULLUP);
  pinMode(col1, OUTPUT);
  digitalWrite(col1, HIGH);
  pinMode(col2, OUTPUT);
  digitalWrite(col2, HIGH);
  pinMode(col3, OUTPUT);
  digitalWrite(col3, HIGH);
  pinMode(col4, OUTPUT);
  digitalWrite(col4, HIGH);
  lcd.begin(16,2);
  lcd.print("CUENTA BANCARIA");
  lcd.setCursor(1,1);
  lcd.print("A.depo B.retiro");
}

char scan_keyboard()
{
  static int rows[] = {row1, row2, row3, row4};
  static int cols[] = {col1, col2, col3, col4};
  static char teclas[4][4] = { { '1', '2', '3', 'A' }, { '4', '5', '6', 'B' }, { '7', '8', '9', 'C' }, { '*', '0', '#', 'D' } };
  int row, col;
  int result = '\0';

  for(col = 0; col < 4 && result == '\0'; ++col){

    digitalWrite(cols[col], LOW);
    for (row = 3; row >= 0; --row){
      if (digitalRead(rows[row]) == LOW){
        result = teclas[row][col];
        break;
      }
    }
    digitalWrite(cols[col], HIGH);
  }
  return result;
}

float acum=0;

void loop()
{

  float in,out;
  char key1,key2,key3,key4;
  int k,o,x=2;

  if (key1=scan_keyboard()=='A')
  {
    x=1;
    lcd.clear();
    lcd.print("VALOR INGRESO:");
  }
}

```

```

do {
    switch(key3=scan_keyboard())
    {

        case '1': in=10000;k=1;key3='\0';break;
        case '2': in=20000;;k=1;key3='\0';break;
        case '3': in=30000;k=1;key3='\0';break;
        case '4': in=40000;k=1;key3='\0';break;
        case '5': in=50000;k=1;key3='\0';break;
        case '6': in=60000;k=1;key3='\0';break;
        case '7': in=70000;k=1;key3='\0';break;
        case '8': in=80000;k=1;key3='\0';break;
        case '9': in=90000;k=1;key3='\0';break;
    }
} while (k!=1);
}

if (key2=scan_keyboard()=='B')
{
    x=0;
    if (acum==0)
    {
        lcd.clear ();
        lcd.print("saldo");
        lcd.setCursor(0,1);
        lcd.print("Insuficiente:");

    }
    else
    {
        lcd.clear ();
        lcd.print("VALOR RETIRO:");

        do {
            switch(key4=scan_keyboard())
            {

                case '1': out=10000;o=1;key4='\0';break;
                case '2': out=20000;;o=1;key4='\0';break;
                case '3': out=30000;o=1;key4='\0';break;
                case '4': out=40000;o=1;key4='\0';break;
                case '5': out=50000;o=1;key4='\0';break;
                case '6': out=60000;o=1;key4='\0';break;
                case '7': out=70000;o=1;key4='\0';break;
                case '8': out=80000;o=1;key4='\0';break;
                case '9': out=90000;o=1;key4='\0';break;
            }
        } while (o!=1);

    }

}

}

if (x==1)
{

```

```

    acum=acum+in;
    lcd.clear ();
    lcd.print("VALOR INGRESADO:");
    lcd.setCursor(0,1);
    lcd.print(in);
    delay(1000);
    lcd.clear ();
    lcd.print("NUEVO SALDO:");
    lcd.setCursor(0,1);
    lcd.print(acum);
    in=0;
    k=2;

}
if (x==0)
{
    if(acum>=out)
    {
        acum=acum-out;
        lcd.clear ();
        lcd.print("VALOR RETIRADO:");
        lcd.setCursor(0,1);
        lcd.print(out);
        delay(1000);
        lcd.clear ();
        lcd.print("NUEVO SALDO:");
        lcd.setCursor(0,1);
        lcd.print(acum);
        out=0; o=2;}
    else
    {
        lcd.clear();
        lcd.print("saldo");
        lcd.setCursor(0,1);
        lcd.print("Insuficiente:");
        delay(2000);
        lcd.clear();
        lcd.print("Saldo:");
        lcd.print(acum);
    }
}

if (scan_keyboard()=='C')
{
    lcd.clear();
    lcd.print("Saldo:");
    lcd.print(acum);
}

if (scan_keyboard()=='D')
{
    lcd.clear();
    lcd.print("CUENTA BANCARIA");
    lcd.setCursor(1,1);
    lcd.print("A.depo B.retiro");
}
}

```

La implementación de este ejercicio se realiza en la plataforma tinkercad y se observa en la figura 22.

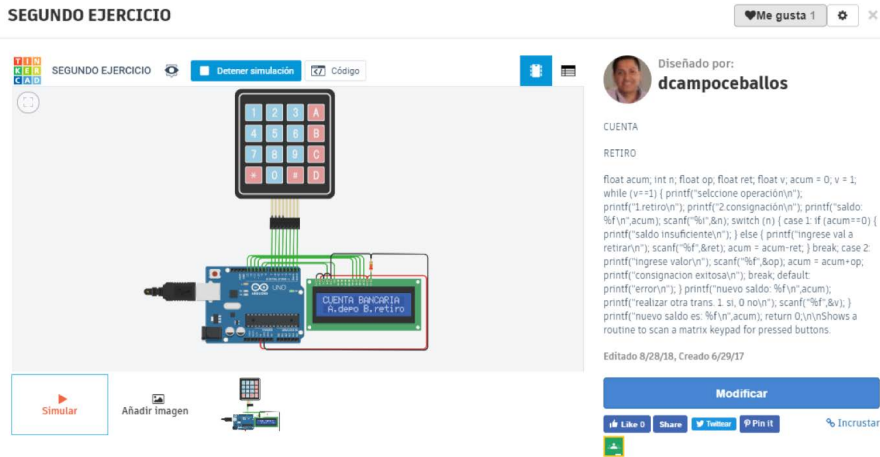


Figura 22

Simulación de algoritmo de cuenta bancaria en la plataforma tinkercad.

Fuente: elaboración propia.

Ejercicios de evaluación

1. Leer e implementar el siguiente código en el sistema TECLADO-ARDUINO-LCD. ¿Qué aparece en la LCD y cómo se controla? Explicar en la práctica.

```
#include <Keypad.h>
```

```
#include <LiquidCrystal.h>
```

```
#define row1 7
#define row2 6
#define row3 5
#define row4 4
#define col1 3
#define col2 2
#define col3 1
#define col4 0
```

```
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
```

```
// en binario (B)
byte Foto_0[8] = { B00000,B00000,B00000,B00110,B01011,B01101,B00110,B00011};
byte Foto_1[8] = { B00001,B00110,B01000,B00100,B01000,B10000,B11000,B01000};
byte Foto_2[8] = { B11110,B00001,B00000,B00000,B11100,B10010,B10000,B11110};
byte Foto_3[8] = { B0000,B10000,B01000,B00101,B01111,B01010,B00101,B11111};
byte Foto_4[8] = { B00001,B00001,B00010,B00010,B00100,B00100,B00100,B00100};
byte Foto_5[8] = { B11110,B11111,B00111,B00011,B00111,B01111,B00111,B00110};
byte Foto_6[8] = { B01110,B00000,B00000,B11111,B00001,B00011,B00011,B10010};
byte Foto_7[8] = { B10110,B01110,B11001,B10001,B11000,B11100,B11000,B11000};

byte Foto_8[8] = { B00000,B00000,B00000,B00110,B01011,B01101,B00110,B00011};
byte Foto_9[8] = { B00001,B00110,B01000,B10000,B11000,B10101,B10000,B11101};
byte Foto_10[8] = { B11110,B00001,B00000,B00000,B11100,B00100,B00100,B11100};
byte Foto_11[8] = { B00000,B10000,B01000,B10001,B01011,B00110,B01101,B01011};
byte Foto_12[8] = { B00001,B00001,B00010,B00010,B00100,B00100,B00100,B00100};
byte Foto_13[8] = { B10101,B11000,B01100,B00111,B01110,B11111,B01111,B01101};
byte Foto_14[8] = { B11000,B00001,B00011,B11101,B00001,B00011,B00011,B00100};
byte Foto_15[8] = { B11110,B11110,B11001,B10001,B11000,B11100,B11000,B11000};
//*****ejercicio manejo de display LCD 16X2 - 8x5*****
```

```
void setup(){

  pinMode(row1, INPUT_PULLUP);
  pinMode(row2, INPUT_PULLUP);
  pinMode(row3, INPUT_PULLUP);
  pinMode(row4, INPUT_PULLUP);
  pinMode(col1, OUTPUT);
  digitalWrite(col1, HIGH);
  pinMode(col2, OUTPUT);
  digitalWrite(col2, HIGH);
  pinMode(col3, OUTPUT);
  digitalWrite(col3, HIGH);
  pinMode(col4, OUTPUT);
  digitalWrite(col4, HIGH);
```

```
lcd.begin(16, 2); //
lcd.createChar(0,Foto_0);
lcd.createChar(1,Foto_1);
lcd.createChar(2,Foto_2);
lcd.createChar(3,Foto_3);
lcd.createChar(4,Foto_4);
lcd.createChar(5,Foto_5);
lcd.createChar(6,Foto_6);
lcd.createChar(7,Foto_7);
lcd.createChar(8,Foto_8);
lcd.createChar(9,Foto_9);
lcd.createChar(10,Foto_10);
lcd.createChar(11,Foto_11);
lcd.createChar(12,Foto_12);
lcd.createChar(13,Foto_13);
lcd.createChar(14,Foto_14);
lcd.createChar(15,Foto_15);
```

```

}

char scan_keyboard()
{
    static int rows[] = {row1, row2, row3, row4};
    static int cols[] = {col1, col2, col3, col4};
    static char teclas[4][4] = {
        { '1', '2', '3', 'A' },
        { '4', '5', '6', 'B' },
        { '7', '8', '9', 'C' },
        { '*', '0', '#', 'D' } };
    int row, col;
    int result = '\0';

    for(col = 0; col < 4 && result == '\0'; ++col){

        digitalWrite(cols[col], LOW);
        for (row = 3; row >= 0; --row){
            if (digitalRead(rows[row]) == LOW){
                result = teclas[row][col];
                break;
            }
        }
        digitalWrite(cols[col], HIGH);
    }

    return result;
}

long i=0;

void loop()
{
    float in,out;
    char key1,key2,key3,key4;
    int k,o,x=2;

    if (key1=scan_keyboard()=='A')
    {
        lcd.clear();
        i++;
        lcd.setCursor(i,0);
        lcd.print((char)0);
        lcd.setCursor(i+1,0);
        lcd.print((char)1);
        lcd.setCursor(i+2,0);
        lcd.print((char)2);
        lcd.setCursor(i+3,0);
        lcd.print((char)3);
        lcd.setCursor(i,1);
        lcd.print((char)4);
        lcd.setCursor(i+1,1);
    }
}

```

```

lcd.print((char)5);
lcd.setCursor(i+2,1);
lcd.print((char)6);
lcd.setCursor(i+3,1);
lcd.print((char)7);

delay(500);

}
}

```

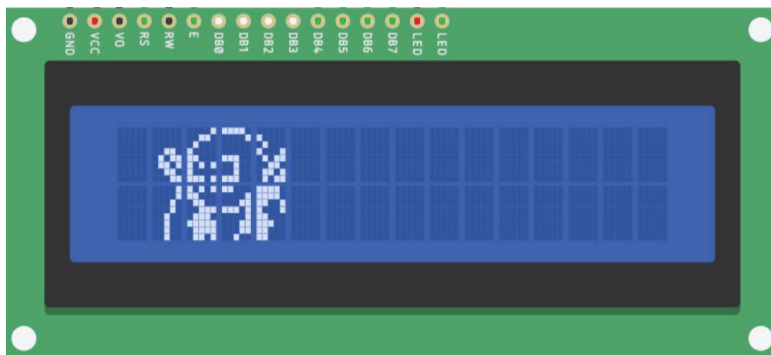


Figura 23

Resultado esperado de la implementación del código en el hardware

Fuente: elaboración propia.

CAPÍTULO 4:

La interfaz mecatrónica 2: herramientas Python para diseño

Resumen

Cuando se trata de generar productos innovadores, la revisión sistemática de la literatura y la vigilancia tecnológica son primordiales para conocer las tendencias que existen en el mercado y aquellas que aún están por explorar. Unas de las herramientas que más potencia el desarrollo de productos innovadores son los lenguajes de programación y sus entornos de desarrollo IDE. De los lenguajes más difundidos hoy en día se puede hablar del conocido Python, un lenguaje interpretado que garantiza una excelente legibilidad y desempeño. Además, contiene soporte para muchas interfaces de interés para el mundo de la integración software y hardware, motivo por el cual se propone como referencia.

En este capítulo se amplía la visión del estudiante, acercándolo cada vez más al diseño concurrente, basado en adaptabilidad y capacidad de los diseños mecatrónicos. Muchas veces los Arduino dificultan el proceso de procesamiento de los datos que se requieren en un proyecto, en esos casos es necesario realizar estrategias de diseño, donde estas plataformas electrónicas no tengan toda la responsabilidad, sino que sirvan apenas como intermediadores y/o como sistemas de adquisición o interpretación de datos. Se tendrán casos de estudio de análisis de datos a través de interfaces desarrolladas en Python y librerías de análisis y procesamiento de datos e imágenes.

Python

Las versiones de este aplicativo se encuentran siempre en actualización. Su repositorio está muy bien mantenido y a su alrededor hay una gran cantidad de discusiones y foros que alimentan unas de las más versátiles plataformas de desarrollo actual. Su página web (www.python.org/), contienen dos versiones la versión 2.7.x y la versión 3.x, ambas funcionan muy bien. Se diferencian, según Gollapudi (2019), por su kernel, al aceptar o no diversas librerías de procesamiento de datos o imágenes como Open Library Computer Vision, OPENCV (<https://opencv.org/>), que abarca un trabajo de diseño un poco más avanzado en el tópico de procesamiento digital de imágenes y visión por computador (Garrido & Joshi, 2018) el cual será abordado en etapas más avanzadas de la carrera .

La instalación de python en plataformas Windows es sencilla, y basta con descargar el archivo ejecutable de la versión deseada o compatible con las librerías que se requieren.

Page y activetcl

Este es un complemento de Python que utiliza la librería activetcl, para crear *framework* o macros de herramientas visuales, didácticas, interactivas y versátiles para manejar todo tipo de interfaces gráficas y capturar datos, presentar información, manipular y generar informes etc. En su página web se puede obtener el aplicativo y su respectiva librería que ayudará a

trabajar fácilmente con interfaces programadas (<http://page.sourceforge.net/>, <https://www.activestate.com/products/activetcl/>). El estudiante tendrá disponible una jornada de instalación previa de estos recursos en el aula.

Ejercicios prácticos

- Implementar una interfaz para el ejercicio del cajero automático, usando una interfaz en Python en un computador de mesa, pero con el procesamiento de Arduino.

Inicialmente el trabajo desde page se limita a ingresar los elementos que se requieren para dar vida a la interfaz. Usualmente, se dispone de botones, etiquetas, espacios de texto, menú desplegable, listas entre otros recursos de la aplicación. Particularmente a los botones se les puede asignar una imagen para mostrar, lo que hace atractivo y genera el realismo y valor de usabilidad para el ejercicio. Un prototipo de la interfaz de trabajo se presenta en la figura 24.

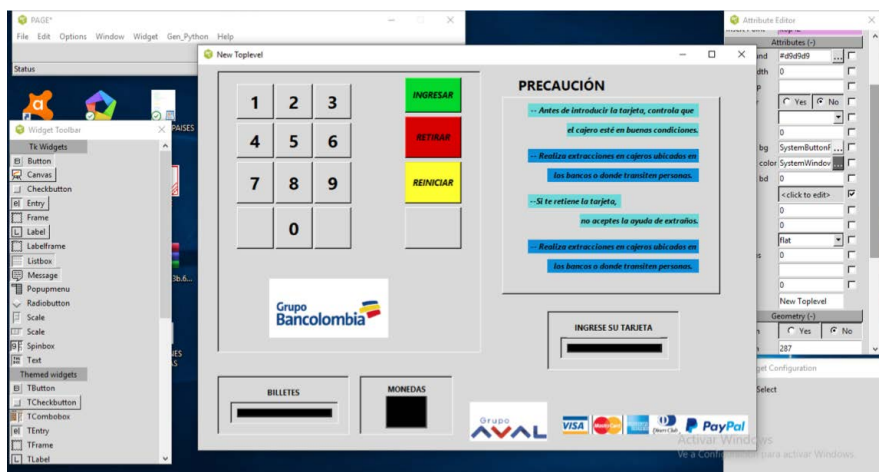


Figura 24

Prototipo de interfaz de usuario desarrollada en PAGE, una GUI para Python.

Fuente: elaboración propia.

Una vez realizado este proceso, la interfaz es codificada y se genera el código para ejecutarlo en el entorno de Python, con lo cual se puede iniciar el proceso de programación. Cabe decir que la interfaz PAGE no permite programar, solo es un recurso que ayuda a generar el código de Python, de acuerdo con la organización visual que se requiere.

Como hecho fundamental, Python requiere de la definición de unas funciones previas que le permitan ejercer acciones, al hacer click en cada uno de los botones diseñados para tal fin. Estas funciones son vitales para el proyecto y se pueden observar a continuación.

```

import sys
import serial # libreria para comunicacion serial arduino python
arduinoData=serial.Serial('COM3',9600,timeout=5) # configura la comunicacion arduino

# el nombre de la función es la que está después de la palabra reservada def
def UNO():
    arduinoData.write("1")# envia a Arduino el 1

def DOS():
    arduinoData.write("2")# envia a Arduino el 2

def TRES():
    arduinoData.write("3")# envia a Arduino el 3

def CUATRO():
    arduinoData.write("4")# envia a Arduino el 4

def CINCO():
    arduinoData.write("5")# envia a Arduino el 5
def SEIS():
    arduinoData.write('6') # envia a Arduino el 6

def SIETE():
    arduinoData.write('7') # envia a Arduino el 7

def OCHO():
    arduinoData.write('8') # envia a Arduino el 8

def NUEVE():
    arduinoData.write('9') # envia a Arduino el 9

def CERO():
    arduinoData.write() # envia a Arduino el 0

def INGRESAR():
    arduinoData.write('A') # envia a Arduino una A, para DEPOSITAR

def RETIRAR():
    arduinoData.write('B') # envia a Arduino una B, para RETIRAR

def SALDO():
    arduinoData.write('C') # envia a Arduino una C, para mostra SALDO

def REINICIAR():
    arduinoData.write('D') # envia a Arduino una D, para REINICIAR

```

```
self.NOMBRE_DEL_BOTON.configure(command=NOMBRE_DE_LA_FUNCION)
```

El código del Arduino correspondiente se muestra a continuación.

```
#include <Keypad.h>

#include <LiquidCrystal.h>

#define row1 7
#define row2 6
#define row3 5
#define row4 4
#define col1 3
#define col2 2
#define col3 1
#define col4 0

char dato;
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

void setup() {
  lcd.begin(16,2);
  lcd.print("CUENTA BANCARIA");
  lcd.setCursor(1,1);
  lcd.print("-Ingres -Retira");

  Serial.begin(9600);
}

float acum=0;

void loop()
{
  if (Serial.available()>0)
  {
    dato=Serial.read(); // LECTURA DEL PUERTO DONDE ESCRIBE PYTHON
    delay(1);
    Serial.print(dato);
    float in,out;
    char key1,key2,key3,key4;
    int k,o,x=2;

    if(dato=='A') // enviado desde la interfaz a través de la función de python
    {
      x=1;
      lcd.clear();
      lcd.print("VALOR INGRESO:");
      //lcd.setCursor(0,1);
      k=1;

      do {
```

```

        switch(dato)
        {

            case '1': in=10000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
            case '2': in=20000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
            case '3': in=30000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
            case '4': in=40000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
            case '5': in=50000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
            case '6': in=60000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
            case '7': in=70000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
            case '8': in=80000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
            case '9': in=90000;k=1;dato='\0';break; // enviado desde la interfaz por función de python
        }
    }while (k!=1);

}

if (dato=='B') // enviado desde la interfaz a través de la función de python
{
    x=0;
    // while (key!='\0'){
        if (acum==0)
        {
            lcd.clear ();
            lcd.print("saldo");
            lcd.setCursor(0,1);
            lcd.print("Insuficiente.");
        }
        else
        {
            lcd.clear ();
            lcd.print("VALOR RETIRO.");

            do {
                switch(dato)
                {
                    case '1': out=10000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                    case '2': out=20000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                    case '3': out=30000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                    case '4': out=40000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                    case '5': out=50000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                    case '6': out=60000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                    case '7': out=70000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                    case '8': out=80000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                    case '9': out=90000;o=1;dato='\0';break; // enviado desde la interfaz por función de python
                }
            }while (o!=1);
        }
    }
}

if (x==1)
{
    acum=acum+in;
    lcd.clear ();
    lcd.print("VALOR INGRESADO.");
}

```

```

lcd.setCursor(0,1);
lcd.print(in);
delay(1000);
lcd.clear();
lcd.print("NUEVO SALDO:");
lcd.setCursor(0,1);
lcd.print(acum);
in=0;////////////////////
k=2;
}
if (x==0)
{
  if(acum>=out)
  {
    acum=acum-out;
    lcd.clear();
    lcd.print("VALOR RETIRADO:");
    lcd.setCursor(0,1);
    lcd.print(out);
    delay(1000);
    lcd.clear();
    lcd.print("NUEVO SALDO:");
    lcd.setCursor(0,1);
    lcd.print(acum);
    out=0;////////////////////
    o=2;
  }
  else
  {
    lcd.clear();
    lcd.print("saldo");
    lcd.setCursor(0,1);
    lcd.print("Insuficiente:");

    delay(2000);
    lcd.clear();
    lcd.print("Saldo:");
    lcd.print(acum);
  }
}
if (dato=='C') // enviado desde la interfaz a través de la función de python
{
  lcd.clear();
  lcd.print("Saldo:");
  lcd.print(acum);
}

if (dato=='D') // enviado desde la interfaz a través de la función de python
{
  lcd.clear();
  lcd.print("CUENTA BANCARIA");
  lcd.setCursor(1,1);
  lcd.print("A.INGRE B.RETIR");
}
}
}
}

```

Finalmente, se ejecuta la aplicación completa en el computador, realizando apropiadamente la configuración del puerto del Arduino. El resultado esperado es como se muestra en la figura 25.



Figura 25
Interfaz PYTHON del cajero electrónico de la sección 3b.
Fuente: elaboración propia.

Ejercicios de evaluación

- Implementar una interfaz gráfica de usuario PAGE-GUI-PYTHON, para controlar el encendido y apagado de un sistema led. Se deja el código ejemplo para Arduino. También se deja un esquema ejemplo de cómo sería el resultado requerido.
- ¿Se podría realizar este procedimiento de forma inalámbrica y usando una plataforma de mensajería tipo whatsapp, telegram o twitter para ejecutar la instrucción de encendido o apagado?

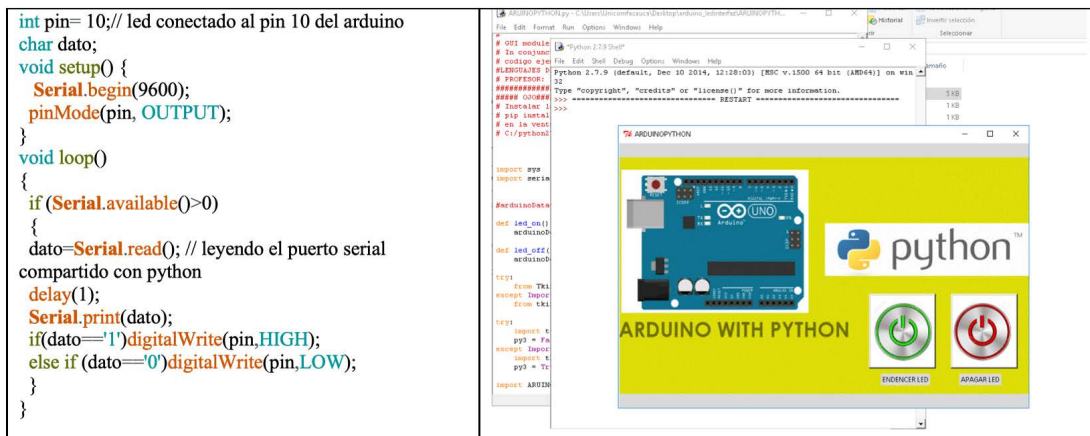


Figura 26
Ejercicio de diseño de producto con interfaz Python
Fuente: elaboración propia.

Recursos complementarios

- Page: (<http://page.sourceforge.net/html/intro.html>)
- PyCharm: (<https://www.jetbrains.com/pycharm/>)

GLOSARIO

- **APK**: aplicación compilada para ser instalada en sistemas operativos de celular basados en Android. Arduino: plataforma de desarrollo electrónico ampliamente difundida por su integración y facilidad de programación. Usa microcontroladores AVR (Advance Virtual Risc) que permite la conexión y flujo de datos USB, wifi, LAN, entre otros, convirtiéndolo en un sistema preferido por estudiantes, profesionales e investigadores.
- **C-C++**: lenguaje de programación muy difundido en ingeniería.
- **CAD (computer aided design)**: se trata de una técnica computacional que ayuda a realizar trazos digitales para construir un dibujo. Se destaca por su simplicidad, facilidad de uso y generación de contenidos multimedia, animados, simulados, en dos y tres dimensiones. Es la herramienta básica para el ingeniero mecatrónico.
- **Framework**: marco de aplicación o conjunto de bibliotecas orientadas a la reutilización para facilitar el desarrollo de aplicaciones.
- **GUI (Graphical User Interface)**: es un desarrollo software que permite la comunicación entre usuario y la Máquina de una forma amigable, utilizando botones, imágenes y gráficos para representar información.
- **IDE (Integrated Development Environment)**: o entorno integrado de desarrollo. Aplicación que ofrece una infraestructura de soporte al desarrollo de software, mediante el editor de código, compilador o intérprete.
- **IEEE (Institute of Electrical and Electronics Engineers)**: la asociación de ingeniería más grande del mundo.
- **Python**: lenguaje de programación interpretado
- **RIMA**: Red de ingeniería mecatrónica, automática y afines (ACOFI, s.f.).

REFERENCIAS

- ACOFI. (s.f.). *Capítulo de Ingeniería Mecatrónica y Automatización*. Asociación colombiana de facultades de ingeniería ACOFI. <https://www.acofi.edu.co/capitulos/aspectos-generales-del-capitulo-de-ingenieria-mecatronica-y-automatizacion/>
- APS, GmbH Europäisches Centrum für Mechatronik. (s.f.) *Home*. APS GmbH. <https://aps-aachen.de/>
- Arduino. (s.f.). *Education*. <https://www.arduino.cc/en/Main/Education>.
- Billingsley, J. (2006) *Essentials of Mechatronics*. Wiley Interscience.
- Bishop, R. (Ed.). (2002). *The mechatronics Handbook*. CRC Press Taylor & Francis Group, LLC.
- Bishop, R. (Ed.). (2005). *Mechatronics: An introduction*. CRC Press Taylor & Francis Group, LLC.
- Bolton, W. (2006). *Programmable Logic Controllers* (4th. ed.). Elsevier Newnes. [https://www.etf.ues.rs.ba/~slubura/Procesni%20racunari/Programmable%20Logic%20Controllers%204th%20Edition%20\(W%20Bolton\).pdf](https://www.etf.ues.rs.ba/~slubura/Procesni%20racunari/Programmable%20Logic%20Controllers%204th%20Edition%20(W%20Bolton).pdf)
- Bolton, W. (2013). *Engineering Science*, (5th ed.). Newnes.
- Bolton, W. (2015). *Mechatronics: Electronic control systems in mechanical and Electrical Engineering*. Pearson.
- Borbón A. y Mora W. (2016). *Edición de Textos Científicos con LATEX*. Revista digital, Matemática, Educación e Internet.
- Bradley, D., Russell, D., Ferguson, I., Isaacs, J., Macleod, A. & White, R. (2015). The Internet of Things - The future or the end of mechatronics. *Mechatronics*, 27, 57-74. <https://doi.org/10.1016/j.mechatronics.2015.02.005>
- Braga, N. (2009). *Mechatronics for the Evil Genius*. McGraw-Hill.
- De Silva, C. (2005). *Mechatronics: An integrated approach*. CRC Press, Taylor & Francis Group, LLC.
- FabLabs. (s.f.). *FabLabs*. <https://www.fablabs.io/>
- For Inspiration and Recognition of Science and Technology [FIRST]. (s.f.). FIRST® Robotics Competition. *First inspires, programs*. <https://www.firstinspires.org/robotics/frc>
- Freddi, A. & Salmon, M. (2018). *Design Principles and Methodologies. From Conceptualization to First Prototyping with Examples and Case Studies*. Springer.
- Garrido, G. & Joshi, P. (2018). *OpenCV 3.x with Python By Example (2da ed.) Second Edition Make the most of OpenCV and Python to build applications for object recognition and augmented reality*. Packt.

- Gentle, R., Edwards, P. & Bolton, B. (Ed.). (2001). *Mechanical Engineering Systems*. Butterworth-Heinemann. <https://www.sciencedirect.com/book/9780750652131/mechanical-engineering-systems#book-info>
- Gollapudi, S. (2019). *Learn Computer Vision Using OpenCV*. Apress Berkeley.
- IEEE. (2019). *Manuscript Templates for Conference Proceedings*. <https://www.ieee.org/conferences/publishing/templates.html>
- IEEE. (s.f.). *Home*. <https://www.ieee.org>
- IEEE. (s.f.). *TryEngineering*. <https://tryengineering.org/>
- Massachusetts Institute of Technology [MIT]. (s.f. a). *Mit App inventor*. <https://appinventor.mit.edu/>
- MeArm Robotic Labs. (s.f.). *MeArm_Calibration*. Codebender. https://codebender.cc/sketch:148456#MeArm_Calibration.ino
- MeArm Robotic Labs. (s.f.). *MeArm*. <https://mearm.com/>
- Mechatronics. (s.f.). *Journal: Mechatronics, the science of intelligent machines*. Elsevier Newnes. <https://www.journals.elsevier.com/mechatronics>
- MIT. (s.f. b). *Projects*. MIT App inventor. <http://ai2.appinventor.mit.edu/>
- Musicandsky. (s.f.) *Laser-cut Mechanical Claw*. Instructables. <https://www.instructables.com/id/Laser-cut-Mechanical-Claw>
- National Aeronautics and Space Administration [NASA]. (2001). *Educational Brief, Humans and Robots*. Nasa. https://er.jsc.nasa.gov/seh/Robot_PDF_Files/humans_and_robots.pdf
- Norton, R. L. (2006) *Diseño de Elementos de Máquinas* (4ta ed.). Pearson.
- Öchsner, A. & Altenbach, H. (Ed.). (2019). *Engineering design applications*. Springer.
- Onwubolu, G. (2005). *Mechatronics: Principles and Applications*. Elsevier, Butterworth-Heinemann.
- Open Source Computer Vision Library. (s.f.). *Opencv*. <https://opencv.org/>
- Overleaf. (s.f.) *Overleaf*. <https://es.overleaf.com/>
- Page. (s.f.) *Introduction*. Page sourceforge. <http://page.sourceforge.net/html/intro.html>
- Phenoptix. (s.f.a) *Pocket Sized Robot Arm MeArm V0.4*. Instructables. <https://www.instructables.com/Pocket-Sized-Robot-Arm-meArm-V04/>
- Phenoptix. (s.f.b) *MeArm Robot Arm - Your Robot - V1.0*. Instructables. <https://www.instructables.com/MeArm-Robot-Arm-Your-Robot-V10/>

- PyCharm. (s.f.) *PyCharm*. JetBrains. <https://www.jetbrains.com/pycharm/>
- Python. (s.f.). *Python*. <https://www.python.org/>
- Rizk R. & Awad, M. (Ed.). (2019). *Mechanism, Machine, Robotics and Mechatronics Sciences*. Springer Berlin Heidelberg.
- Robot Books. (s.f.). *Robot Books*. <http://www.robotbooks.com/>
- Shetty, D. & Kolk, A. (2011). *Mechatronics System Design*, (2a ed.). Cengage Learning.
- Sloan, L. (2015). *3D Printing with Autodesk 123D*. McGraw Hill TAB.
- Software Informer. (2022). RDWorks 8.0: Laser Cut Software. *3D modeling*. <https://rdworks.software.informer.com/8.0/>.
- Thingiverse. (s.f.). *MakerBot, Thingiverse*. <https://www.thingiverse.com/>
- Tinkercad. (s.f.). *Autodesk Tinkercad*. <https://www.tinkercad.com/>
- Unicomfacauc. (2018). *Documento maestro para renovación de registro calificado*. Corporación Universitaria Comfacauc.
- Universidad de Pamplona. (2011). *Proyecto educativo del programa de ingeniería mecatrónica*. Universidad de Pamplona. https://www.unipamplona.edu.co/unipamplona/portallG/home_31/recursos/01general/29052014/pep.pdf



Semillero de Mecatrónica

Aunque el proceso de aceptación de la mecatrónica como campo de estudio e investigación en el mundo académico ha sido relativamente lento, en la actualidad se puede afirmar que ya ha alcanzado reconocimiento. “No es una nueva disciplina, sino un enfoque de aplicación de las últimas técnicas en ingeniería mecánica de precisión, teoría del control, ciencias computacionales y electrónica, al proceso de diseño, para la creación de productos funcionales, integrados y adaptables” (Universidad de Pamplona, 2011, p. 6). Es decir, innovadores. Así, se requiere comprender que la electrónica y el control computacional son componentes fundamentales del proceso de diseño, a la vez que se integran al proceso de fabricación digital. La formación del Ingeniero mecatrónico necesita estar fuertemente relacionada con los principios de diseño para atender las necesidades del sector productivo de la región, abordados desde una metodología concurrente que consolida la democratización, adaptación e integración de tecnologías a sus productos, procesos y servicios. Este libro es una herramienta inicial para generar productos, procesos y sistemas que integren tecnologías de la mecánica, la electrónica y la informática.

La guía aborda desde la construcción de un brazo robótico simple hasta el desarrollo de interfaces mecatrónicas simples, aplicando técnicas y métodos de diseño conocidos, fomentando el aprendizaje significativo y experiencial del estudiante. Se da especial atención al desarrollo de ejercicios para evaluar funcionalidad, apariencia, ahorro de energía, relación con el entorno, etc., propios de sistemas de alta complejidad tecnológica y que prometen solventar las necesidades del sector productivo del departamento del Cauca y Colombia.

