

IF4000 - Arquitectura de Computadores Investigación (Noviembre 2023)

Jesner Melgara Murillo., & Alisson Rodríguez Mora.

Resumen—Este artículo presenta los resultados de una investigación sobre el desarrollo de un juego de memoria de colores que combina elementos físicos y virtuales. El objetivo principal es ofrecer a los jugadores una experiencia agradable al tiempo que se promueve el desarrollo cognitivo. Este juego interactivo de memoria, diseñado en una plataforma Arduino con un microcontrolador ATmega, tiene como objetivo estimular la memoria, la concentración y la retención de habilidades. Al fomentar la práctica regular de estas habilidades, el juego se convierte en una herramienta educativa y recreativa adecuada para personas de todas las edades. La metodología involucró la programación del microcontrolador, el diseño de la interfaz en Java y la sincronización de componentes físicos como pulsadores y LEDs, lo que proporciona una experiencia de juego dinámica y enriquecedora. Los resultados clave incluyen la implementación exitosa de la comunicación entre hardware y software y la introducción del concepto de "vidas físicas", que añade un desafío adicional. El estudio concluye que este enfoque combina eficazmente el entretenimiento y el aprendizaje.

Palabras clave—Hardware, Software, Arduino, Java, Juego, Memoria.

Abstract—This article presents the results of research on the development of a color memory game that combines physical and virtual elements. The primary objective is to offer players an enjoyable experience while also promoting cognitive development. This interactive memory game, designed on an Arduino platform with an ATmega microcontroller, aims to stimulate memory, concentration, and retention skills. By encouraging the regular practice of these abilities, the game becomes an educational and recreational tool suitable for individuals of all ages. The methodology involved microcontroller programming, Java interface design, and synchronization of physical components like buttons and LEDs, providing a dynamic and enriching gaming experience. Key outcomes include the successful implementation of hardware-software communication and the introduction of the concept of "physical lives," adding an extra challenge. The study concludes that this approach effectively combines entertainment and learning.

Keywords—Hardware, Software, Arduino, Java, Game, Memory.

I. INTRODUCCIÓN

En la era actual de la tecnología y la informática, la integración de hardware y software ha dado lugar a una amplia gama de aplicaciones y sistemas interactivos. Un campo que ha experimentado un marcado crecimiento es la creación de

juegos interactivos que involucran la comunicación entre microcontroladores y aplicaciones de software.

El objetivo fundamental de esta investigación es estimular la memoria, la concentración y la capacidad de retención de los jugadores. La promoción constante de estas habilidades se traduce en un enfoque educativo y entretenido, lo que lo convierte en un recurso beneficioso para personas de todas las edades. La esencia de esta investigación reside en la cooperación entre hardware y software, que da vida a una experiencia de juego dinámica y enriquecedora.

Para alcanzar este propósito, se ha diseñado un juego de memoria de colores que ofrece tres modos de dificultad: "Fácil," "Intermedio" y "Difícil." El proyecto se desarrolla en tres etapas distintas: la primera permite a los jugadores seleccionar su preferido entre los modos de juego disponibles. Tras esta elección, se presenta la pantalla de juego, desafiando a los jugadores a recordar y repetir secuencias de colores dentro de límites de tiempo específicos. La tercera etapa es la pantalla final, donde los jugadores tienen la opción de reiniciar el juego o volver al menú principal en caso de pérdida.

La singularidad de este proyecto radica en la sincronización de componentes físicos, como pulsadores y LEDs dispuestos en un protoboard, con la interfaz de Java a través del microcontrolador Arduino. Además, se introduce el innovador concepto de "vidas" físicas, que se reflejan tanto en el Arduino como en los LEDs, añadiendo un elemento adicional de desafío al juego.

II. MARCO TEÓRICO

A. Marco Conceptual

A continuación se presenta una breve descripción de conceptos importantes que brindan contexto a este trabajo.

1) Hardware

Podemos definir el hardware como la parte física del ordenador: tanto la caja y los componentes internos (placa base, disco duro...) como los elementos conectados a él (teclado, ratón, monitor, impresora...) [1].

2) Software

El software es la parte inmaterial, o intangible, que hace que funcione el hardware. Mediante el software, el ser humano puede comunicarse e interactuar con los distintos elementos informáticos. El software lo constituyen los programas, las instrucciones, las aplicaciones informáticas y el sistema operativo [1].

3) Java

Java es un lenguaje de programación de alto nivel y una plataforma de desarrollo informático. Fue creado por Sun Microsystems (ahora propiedad de Oracle Corporation) en la década de 1990 y se ha convertido en uno de los lenguajes de programación más populares y ampliamente utilizados en el mundo [2].

¹ Universidad de Costa Rica. Sede del Atlántico. II Semestre 2023.

4) *Biblioteca de Java*

Una "biblioteca" o "librería" de Java, se refiere a un conjunto de clases y métodos predefinidos que proporcionan funcionalidades específicas para tareas comunes en desarrollo de software. Estas bibliotecas están diseñadas para facilitar el desarrollo de aplicaciones al proporcionar funciones y herramientas reutilizables [2].

5) *C/C++*

C++ es un lenguaje de programación creado en la década de 1980 por Bjarne Stroustrup. Fue diseñado para extender el lenguaje C con la capacidad de trabajar con objetos, lo que lo convierte en un lenguaje orientado a objetos. Con el tiempo, se agregaron características de programación genérica, lo que lo hace un lenguaje multiparadigma, admitiendo programación estructurada, orientada a objetos y genérica [3]. En resumen, C++ es un lenguaje híbrido que combina múltiples enfoques de programación.

6) *Microcontrolador*

Un microcontrolador es una pequeña computadora en un solo chip de circuito integrado (CI) de óxido metálico semiconductor (MOS). Un microcontrolador contiene una o más unidades centrales de procesamiento (núcleos de procesador) junto con memoria y periféricos de entrada/salida programables [4].

7) *Arduino*

Arduino es una plataforma de código libre diseñada para facilitar proyectos de electrónica. Posee un entorno gráfico de desarrollo que usa un lenguaje de programación processing/wiring y un gestor de arranque; en lo que respecta al hardware está compuesta por un microcontrolador y puertos de entrada y salida [5].

8) *Pines Digitales*

Los pines digitales son las conexiones digitales de los dispositivos conectados en la placa. Según [5] La placa de Arduino cuenta con 14 pines digitales, que van del 0 al 13. Los pines digitales de Arduino pueden ser usados tanto de entrada como de salida; y solo entienden dos valores: 0-1, o lo que es lo mismo, 0V o 5V.

9) *Pines Analógicos*

Arduino proporciona pines analógicos que permiten la lectura de señales analógicas, pueden medir valores intermedios entre 0V y 5V, como tensiones o corrientes variables, en lugar de simplemente señales digitales (encendido/apagado). Estos pines analógicos se utilizan comúnmente para leer sensores analógicos, como sensores de luz, temperatura o humedad. [5].

10) *Resistencias*

Son un tipo de componente electrónico pasivo que actúa como cargas de manera que no generan ni amplifican la señal [6].

11) *Diodo LED*

Son diodos luminiscentes ya que en ellos, la energía liberada se irradia en forma de luz. Existen diodos LED de

diferentes formas, tamaños y colores y son muy empleados en circuitos electrónicos [6].

III. DISCUSIÓN

A. *Experimentos*

Los experimentos llevados a cabo en esta investigación se centraron en la implementación y evaluación del juego de memoria de colores que combina elementos físicos y virtuales, con un enfoque en la comunicación efectiva entre una aplicación Java y un microcontrolador Arduino utilizando la biblioteca JSerialComm. Los experimentos se diseñaron para evaluar la funcionalidad, el rendimiento y la experiencia del usuario del juego, así como su valor educativo y recreativo.

1) *Configuración de los Experimentos*

Para llevar a cabo los experimentos, se utilizó un entorno de laboratorio equipado con un protoboard, un Arduino, tres LEDs (que representan las vidas del jugador) y cuatro botones (pulsadores) que permiten a los jugadores interactuar con el juego. La comunicación entre la aplicación Java y Arduino se estableció mediante la biblioteca JSerialComm. La aplicación Java, diseñada con programación orientada a objetos (POO), controla la lógica del juego, la interfaz de usuario y la comunicación con Arduino.

2) *Resultados y Observaciones*

Durante los experimentos, se observó que la comunicación entre la aplicación Java y Arduino funcionó de manera efectiva. Los comandos y datos se transmitieron correctamente, lo que permitió una interacción fluida entre los componentes físicos y virtuales del juego. Además, se implementó con éxito el concepto de "vidas" físicas representadas por los LEDs en el protoboard, lo que agregó un desafío adicional para los jugadores.

3) *Valor Educativo y Recreativo*

Un aspecto destacado de los experimentos fue el valor educativo y recreativo del juego. Se encontró que el juego no solo entretenía a los jugadores, sino que también fomentaba el desarrollo cognitivo. El ejercicio constante de las habilidades de memoria y concentración a través del juego lo convierte en una herramienta beneficiosa para individuos de todas las edades. Los participantes expresaron su satisfacción con la experiencia y destacaron la combinación efectiva de entretenimiento y aprendizaje.

4) *Limitaciones y Recomendaciones*

A pesar de los resultados positivos, es importante señalar algunas limitaciones. La comunicación entre la aplicación Java y Arduino depende de la estabilidad y disponibilidad de los puertos serie, lo que podría generar problemas en sistemas con configuraciones específicas. Además, se recomienda una mayor optimización del rendimiento de la aplicación Java y una exploración de posibles mejoras en la interfaz de usuario.

Como recomendación para futuras implementaciones, se sugiere la exploración de nuevas características y niveles de juego para enriquecer aún más la experiencia de los jugadores y fomentar su desarrollo cognitivo de manera efectiva.

En resumen, los experimentos demostraron que la comunicación entre la aplicación Java y Arduino mediante la biblioteca JSerialComm es eficaz y confiable, y que el juego de memoria de colores logra sus objetivos educativos y recreativos. Las limitaciones identificadas brindan oportunidades para futuras mejoras en la implementación y enriquecimiento de la experiencia del usuario.

En cuanto a la experiencia del usuario, se observó que el juego de memoria de colores logró su objetivo de estimular la memoria, la concentración y la capacidad de retención de los jugadores. La dificultad ajustable de los niveles "Fácil," "Intermedio" y "Difícil" permitió adaptar el juego a diferentes audiencias, lo que lo convierte en una herramienta educativa versátil y recreativa.

B. Implementación

1) Componentes físicos

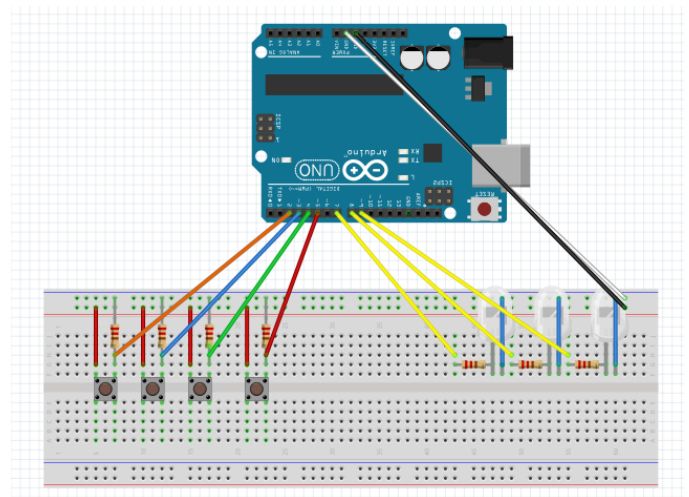
En la implementación física de este proyecto, se utilizan diversos componentes para crear la interfaz tangible del juego. Estos elementos incluyen una protoboard, un Arduino, tres LEDs y cuatro botones, junto con resistencias y cables de conexión. Además, el Arduino se conecta a una computadora a través del puerto USB para facilitar la programación y la comunicación.

En lo que respecta a los LEDs, se han conectado a los pines digitales de salida 7, 8 y 9 del Arduino. Cada LED representa una de las vidas disponibles para el jugador en cada partida del juego. La conexión se realiza mediante resistencias para limitar la corriente y garantizar un funcionamiento adecuado.

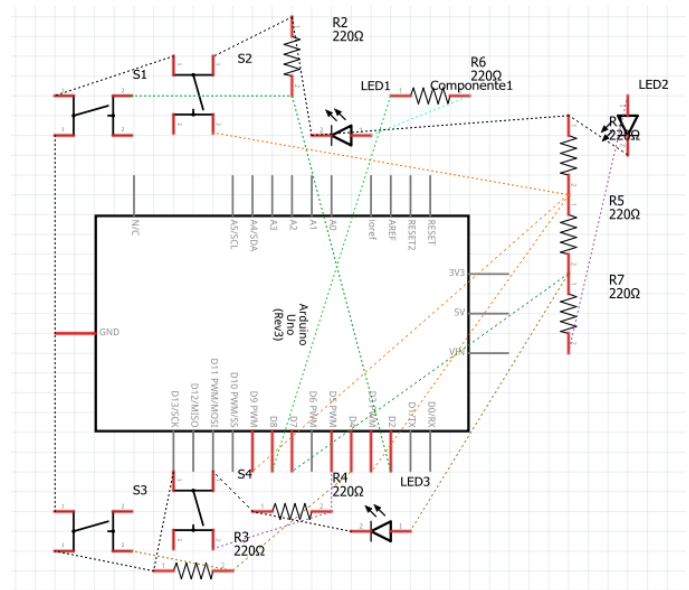
Los botones, por otro lado, están conectados a una fuente de alimentación, con un cable positivo y otro negativo para asegurar el flujo de energía necesario. Además, estos botones envían señales digitales a los pines de entrada 2, 3, 4 y 5 del Arduino. Cada botón se asigna a una función específica en el juego y permite al jugador interactuar con la interfaz.

En conjunto, esta configuración física permite la interacción efectiva entre el jugador y el juego. Los LEDs reflejan las vidas disponibles, mientras que los botones permiten al jugador seleccionar opciones y responder a los desafíos del juego. La conexión y disposición de estos componentes son fundamentales para la dinámica del juego y contribuyen a la experiencia interactiva que se busca lograr.

a) Componentes físicos



b) Diagrama esquemático



2) Aplicación de Software

a) JAVA

En esta fase de la implementación, se desarrolló la aplicación de software, que contiene su parte lógica (Domain) de la aplicación en JAVA utilizando programación orientada a objetos (POO). Esta parte incluye la lógica del juego, como la generación de secuencias de colores basadas en el nivel de dificultad seleccionado, así como el manejo del temporizador para establecer límites de tiempo, entre otras funcionalidades que hacen que el juego de memoria se ejecute de manera correcta.

1) Sincronización de Componentes Físicos y Software

La interfaz gráfica (GUI) se diseñó para la interacción del jugador, permitiendo la visualización de la secuencia de colores y las opciones de juego. Además, se estableció una comunicación efectiva con Arduino a través del puerto COM3.

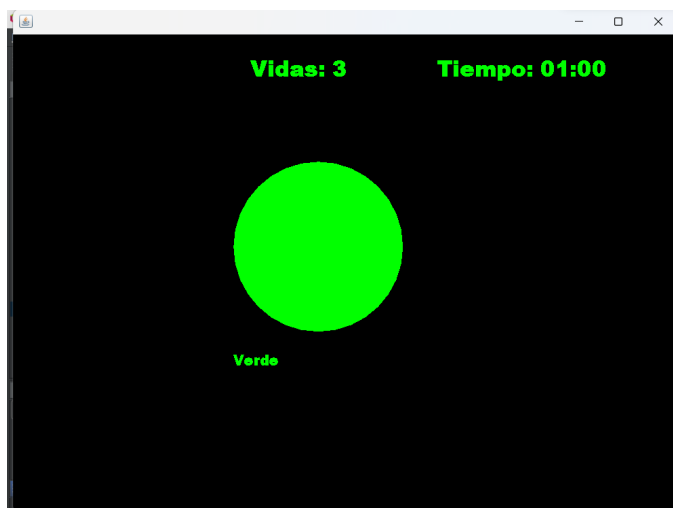
Para facilitar esta comunicación, se utilizó la biblioteca JSerial-common.

A continuación, se mostrarán las interfaces de cada una de las pantallas presentes en el juego de memoria.

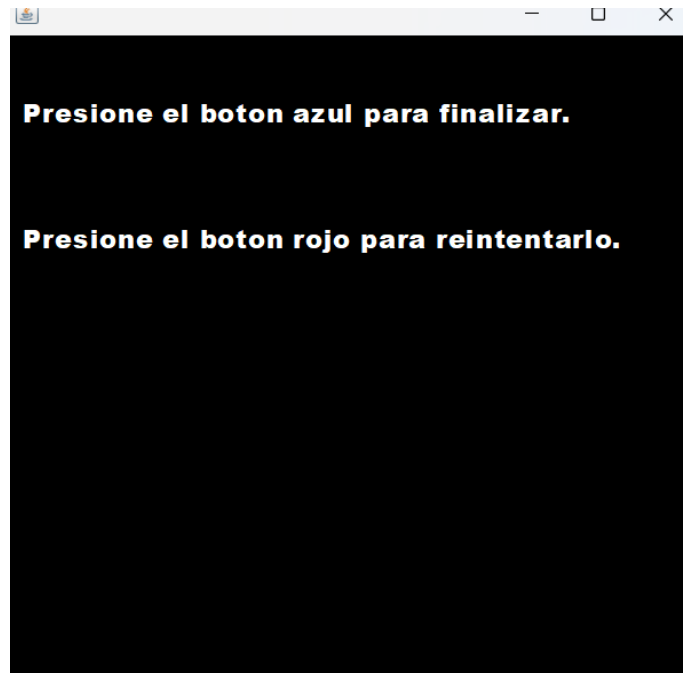
a) 1° Pantalla (Modo de juego)



b) 2° Pantalla (Juego)



c) 3° Pantalla (Finalización)



2) JSerial-comm: Una Biblioteca Java para Comunicación con Puertos Serie

Tal como lo indica [7], JSerialComm es una biblioteca Java independiente de la plataforma que ofrece acceso a puertos serie estándar sin requerir bibliotecas externas ni código nativo. Algunas de sus características clave incluyen la configuración personalizable de puertos, el control de flujo configurable, la lectura y escritura directa de datos, la integración con flujos de bytes de Java y las notificaciones basadas en eventos.

Por lo que, en la presente investigación, la biblioteca JSerialComm se utilizó para establecer una comunicación eficaz y confiable entre la aplicación Java y Arduino a través de puertos serie. Esto permitió que la aplicación Java enviara comandos a Arduino y recibiera datos, lo que es fundamental para el funcionamiento de su juego de memoria de colores. A continuación se mencionan los pasos de implementación de la siguiente manera:

1. Importar la Biblioteca JSerial-Comm

Lo primero que se hizo fue importar la biblioteca JSerialComm en el proyecto de Java. Esto se logra mediante la inclusión del archivo JAR de la biblioteca en el proyecto y la importación de las clases necesarias en sus archivos Java.

2. Obtener una Lista de Puertos Serie Disponibles

Utilizando la función estática `SerialPort.getCommPorts()`, se generó una lista de todos los puertos serie disponibles en el sistema, tanto reales como virtuales. Esto proporciona una visión completa de los puertos que la aplicación puede utilizar.

3. Seleccionar el Puerto Serie para Comunicación

Una vez que se tiene la lista de puertos disponibles, se puede seleccionar el puerto serie específico que se utilizará para la comunicación con Arduino. Esto se hace creando un objeto `SerialPort` a partir del descriptor del puerto deseado utilizando `SerialPort.getCommPort(String portDescriptor)`.

4. Configurar el Puerto Serie

Luego, se configuran los parámetros del puerto serie, como la velocidad en baudios, bits de datos, bits de parada y paridad, según las necesidades de la comunicación con Arduino.

5. Abrir el Puerto Serie

El puerto serie seleccionado se abre para permitir la comunicación. Una vez que está abierto, la aplicación Java puede enviar y recibir datos a través de este puerto.

6. Enviar y Recibir Datos con Arduino

La biblioteca `JSerialComm` permite enviar y recibir datos directamente a través del puerto serie. Esto es esencial para la comunicación bidireccional con Arduino. La aplicación Java puede enviar comandos o datos a Arduino y recibir respuestas o información de estado.

7. Manejo de Eventos y Devoluciones de Llamada

La biblioteca también admite notificaciones basadas en eventos, lo que significa que la aplicación Java puede recibir devoluciones de llamada cuando ocurren ciertos eventos, como la disponibilidad de nuevos datos para leer, la escritura exitosa de datos, cambios en las líneas de control del módem, entre otros. Esto facilita la sincronización de la comunicación.

8. Cerrar el Puerto Serie

Al finalizar la comunicación, es importante cerrar el puerto serie para liberar los recursos y permitir su uso por otras aplicaciones.

b) Arduino (C/C++)

En la implementación de Arduino, se abordó la lógica que controla las señales digitales recibidas y enviadas. Los pines de entrada 2, 3, 4 y 5 se asignaron a los pulsadores correspondientes: Naranja (pin 2), Azul (pin 3), Verde (pin 4) y Rojo (pin 5). Arduino constantemente monitorea estas señales y envía números específicos cuando se presiona un botón (por ejemplo, "2" para Naranja, "9" para Azul, "6" para Verde y "8" para Rojo) hacia la aplicación JAVA.

1) *Sincronización de Componentes Físicos y Software en Arduino*

La implementación de Arduino controla los tres LEDs que representan las vidas del jugador. Estos LEDs se encuentran en los pines de salida 7, 8 y 9. Al inicio del juego, los LEDs se encienden (HIGH) para mostrar las vidas disponibles. Cuando se recibe un número específico a través de la comunicación

serial desde JAVA (por ejemplo, "7", "8" o "9"), se interpreta como una pérdida de vida y se apaga el LED correspondiente.

c) Valor Educativo y Recreativo

La implementación conjunta de JAVA y Arduino ha permitido la creación de un juego de memoria de colores que no solo entretiene, sino que también fomenta el desarrollo cognitivo de los jugadores. El juego ofrece una experiencia educativa y recreativa beneficiosa para personas de todas las edades, al estimular habilidades como la memoria y la concentración.

d) Limitaciones de la Implementación

A pesar del éxito general de la implementación, es importante reconocer las limitaciones. Puede haber desafíos técnicos o restricciones de recursos que afecten la calidad y el alcance del juego. Identificar estas limitaciones es esencial para futuras mejoras.

e) Recomendaciones para Futuras Implementaciones

Como trabajo futuro, se sugiere explorar posibles mejoras y expansiones en la implementación. Esto podría incluir la adición de nuevas características, niveles de dificultad o modos de juego para enriquecer la experiencia del jugador y promover aún más el desarrollo cognitivo.

IV. CONCLUSIONES

En el transcurso de esta investigación, se logró con éxito la integración de hardware y software al conectar un microcontrolador ATmega, presente en la placa Arduino, con una interfaz de usuario desarrollada en el lenguaje de programación Java. Esta integración permitió la creación de un juego de memoria de colores que combina elementos físicos y virtuales, ofreciendo a los jugadores una experiencia interactiva y entretenida. La capacidad de sincronizar componentes físicos, como pulsadores y LEDs, con la interfaz Java a través del Arduino demostró ser efectiva y agregó un desafío adicional al juego.

Uno de los objetivos principales de esta investigación era estimular habilidades cognitivas, como la memoria, la concentración y la retención de información. El juego diseñado en esta plataforma posee potencial como una herramienta educativa y recreativa beneficiosa para personas de todas las edades. La inclusión de tres modos de dificultad, "Fácil," "Intermedio" y "Difícil," garantiza una experiencia escalable que se adapta a las preferencias y habilidades de los jugadores, ampliando su atractivo y utilidad.

Además, la introducción del concepto de "vidas" físicas, que se reflejan en LEDs, no solo añadió un elemento de desafío al juego, sino que también lo convirtió en una experiencia de aprendizaje adicional. Estas "vidas" no solo entretienen, sino que también fomentan el desarrollo cognitivo de los jugadores. Esto sugiere un potencial educativo importante para este juego.

de memoria de colores.

Como trabajo futuro, se buscará potenciar y expandir esta plataforma, explorando nuevas características y posibilidades para mejorar aún más la experiencia de los jugadores y seguir promoviendo su desarrollo cognitivo de manera efectiva. Este proyecto sienta las bases para futuras investigaciones y mejoras que puedan enriquecer aún más el juego y sus beneficios.

REFERENCIAS

- [1] Instituto de Educación Secundaria Playa Honda. (2018). Arquitectura de un PC. Unidad 1. Fundamentos De Informática [En línea]. Disponible en:
<https://www3.gobiernodecanarias.org/medusa/ecoblog/mgoncal/files/2018/11/unidad-1-hardware-tic-4-eso-ies-playa-honda-1.pdf>
- [2] J. M Ladrón de Guevara. *Fundamentos de programación en Java*. Facultad de Informática. Madrid: Editorial EME, 2013.
<https://www.tesuva.edu.co/phocadownloadpap/Fundamentos%20de%20programacion%20en%20Java.pdf>
- [3] F. Martínez del Río. *Fundamentos Básicos de Programación en C++*. Jaén: Universidad De Jaén, 2015.
<https://www4.ujaen.es/~fmartin/apuntesC++.pdf>
- [4] Arduino. (s.f). Y. D. Acevedo., A. H. Colón & T. N. Avilés [Online]. Disponible en:
<https://indico.cern.ch/event/1068475/contributions/4493027/attachments/2296022/3904870/Arduino%20.pdf>
- [5] O. Vázquez Bautista. (s.f). Componentes de la Placa de Arduino. Universidad Autónoma del Estado de Hidalgo. [En línea]. Disponible en:
<https://repository.uaeh.edu.mx/bitstream/bitstream/handle/123456789/19851/componentes-placa-arduino.pdf?sequence=1&isAllowed=y>
- [6] Federación de la Enseñanza CC.OO de Andalucía. “COMPONENTES DE CIRCUITOS ELECTRÓNICOS EMPLEADOS EN TECNOLOGÍA”. Noviembre de 2005.
<https://www.feandalucia.ccoo.es/docu/p5sd6324.pdf>
- [7] JSerialComm. Platform-independent serial port access for Java. GitHub Pages, s.f.. [En línea]. Disponible en:
<https://fazecast.github.io/jSerialComm/>