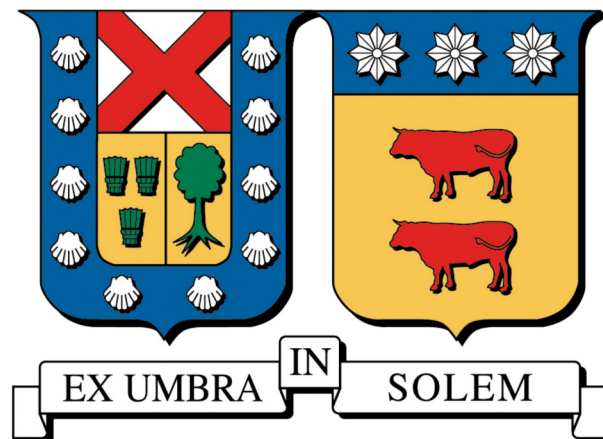


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



**“DISEÑO DE EXPERIENCIAS PARA TALLERES DE
ELECTRÓNICA ORIENTADOS A ESTUDIANTES DE
EDUCACIÓN PRIMARIA Y SECUNDARIA”.**

CAMILA VICTORIA HUERTA CÁDIZ

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRÓNICO**

PROFESOR GUÍA : GONZALO CAVAJAL
PROFESOR CORREFERENTE : MOHAMED ABDELHAMID

MAYO 2023

Agradecimientos

No fue fácil, no fue rápido y no fue como lo planeé, pero definitivamente será inolvidable. Dentro de los recuerdos de mi paso por la Universidad están los seres que hicieron todo esto inolvidable, de la forma en que cuando se me vienen a la mente, solo me queda sonreír y sentirme agradecida de que hayamos coincidido en nuestras vidas.

Un sincero agradecimiento a mi madre, mi hermano y padre, a mis amigos y amigas que conocí en este proceso, y a los que venían acompañándome desde antes, a las grandes personas que no están ahora en mi vida, pero que cuando lo estuvieron dejaron recuerdos maravilloso y finalmente a mis mascotas que amo.

Resumen

Las Academias de Tecnología del Departamento de Electrónica de la Universidad Técnica Federico Santa María (UTFSM), funcionan anualmente impartiendo clases de electrónica básica a estudiantes de enseñanza básica y media de establecimientos educacionales de la V Región, de mano de estudiantes del mismo departamento. Estas clases se centran en mostrar el mundo de la electrónica a través de sesiones experienciales (hasta año 2019) no coordinadas, dependientes de cada tutor o tutora (estudiante universitario) a cargo de la enseñanza, lo que provoca un nulo control sobre los tópicos enseñados a los participantes y el real aprendizaje de éstos y una capacidad de retroalimentación de el o la participante hacia las Academias inexistente.

Tomando en cuenta lo anterior es que surge la necesidad de estandarizar el aprendizaje y su proceso. Esto quiere decir: definir una lista de contenidos a entregar y el orden en el que se hará, además de planificar métodos de medición de aprendizaje de tal forma de poder analizar el impacto de las academias como tal en el aprendizaje de electrónica y competencias sello incluidas en la misión de éstas.

Este trabajo se basa en el proceso de diseño de Experiencias para estudiantes de enseñanza básica y media, cada experiencia cuenta con una Guía de Contenidos y un Instrumento de Medición. Dicho trabajo viene a intentar solucionar el problema de improvisaciones a la hora de la entrega de contenidos por parte del tutor o tutora encargada de los participantes y la medición de objetivos de aprendizaje. Dado el contexto de los últimos años (2019, 2020 y 2021) y en un afán de no cesar el funcionamiento de las academias se planteó la idea de realizar Academias de Tecnología en línea, las cuales con una visión a futuro, se volverán imprescindibles si se quiere evitar los límites geográficos para poder llegar a estudiantes de otras regiones del país, es por esto que las primeras experiencias fueron hechas en un contexto en línea, lo cual no restringe esta memoria de ningún modo a este tipo de ejecución, ya que la riqueza de ésta se basa en la explicación del proceso de estandarización y medición de aprendizaje de las Academias, ya sea a corto o largo plazo.

Abstract

The Academias de Tecnología of the Electronics Department at the Federico Santa María Technical University (UTFSM) operate annually by providing basic electronics classes to primary and secondary school students from educational institutions in the V Region. (These) classes are taught by students from the same department. Until 2019, these classes focused on introducing the world of electronics through experiential sessions that were not coordinated and depended on each tutor (university student) in charge of teaching. This lack of coordination resulted in no control over the topics taught to the participants, their actual learning, and the absence of feedback from the participants to the academies.

Considering the above, there arises the need to standardize the learning and its process. This means defining a list of contents to be delivered and the order in which they will be taught, as well as planning methods to measure learning in order to analyze the impact of the academies on electronics learning and the core competencies included in their mission.

This work is based on the process of designing experiences for primary and secondary school students. Each experience includes a Content Guide and a Measurement Instrument. This work aims to solve the problem of improvisation when delivering content by the tutor in charge of the participants and the measurement of learning objectives. Given the context of the past years (2019, 2020, and 2021) and with the intention of not interrupting the operation of the academies, the idea of conducting online Academias de Tecnología was proposed. With a future vision, these online academies will become essential to overcome geographical limitations and reach students from other regions of the country. Therefore, the initial experiences were conducted online. However, this does not restrict this thesis in any way to this type of execution, as its richness lies in explaining the process of standardization and measurement of learning in the Academias de Tecnología, whether in the short or long term.

Índice general

1. Introducción	6
1.1. Contexto y motivación	7
1.2. Planteamiento del problema	8
1.3. Objetivos	9
1.4. Alcances	10
1.5. Contribuciones	11
1.6. Organización del informe	11
2. Revisión de herramienta Tinkercad de simulación para aprendizaje de electrónica y programación en línea	12
2.1. Herramientas de simulación	12
2.2. Tinkercad	14
2.2.1. Entorno visual representativo	14
2.2.2. Errores controlados	20
2.2.3. Uso de Arduino	21
2.2.4. Uso de código	22
2.2.5. Función de clase	23
3. Diseño de experiencias y métodos de evaluación	25
3.1. Descripción del flujo de trabajo de esta memoria	26
3.1.1. Trabajo inicial	27
3.1.2. Adecuación de contenidos para estudiantes de enseñanza básica y media	28
3.1.3. Preparación de desafíos adecuados a los contenidos	28
3.1.4. Desarrollo del documento	28
3.1.5. Revisión de documento según apoyo académico [13]	29
3.1.6. Ejecución de documento	30
3.1.7. Recolección de retroalimentación de la sesión	31
3.1.8. Documento pedagógicamente validado	31
3.1.9. Corrección tomando en cuenta realimentación de ejecución en línea . .	31
3.1.10. Equipo inicial desarrollador hace realimentación a guía e instrumento evaluativo	31
3.1.11. Últimas correcciones y entrega del material guía de cada sesión	32
4. Material desarrollado	33
4.1. Guías de trabajo	33

4.1.1.	Contenidos generales	33
4.1.2.	Tipos de guías	33
4.1.3.	Sesiones	34
4.1.4.	Rúbricas de evaluación [14] [15]	36
5.	Conclusiones y recomendaciones	38
5.1.	Conclusiones	38
5.2.	Recomendaciones para trabajo futuro	38
	Bibliografía	40
	A. Anexo	42
	B. Anexo	88
	C. Anexo	140

1 | Introducción

Se realizó una investigación para conocer la situación actual del campo de la enseñanza de electrónica y programación para niños en Chile. El objetivo era conocer las opciones disponibles y determinar el enfoque necesario para diferenciarse de las alternativas existentes. Esta investigación proporcionó un panorama general y la necesidad de un enfoque único.

Se encontraron diversas opciones relacionadas con la programación, electrónica y robótica, que variaban en términos de pago, gratuitas, especializadas en robótica o programación, y ofrecidas en línea o de forma presencial. Sin embargo, no se encontró un taller que cumpliera con todas las características de las academias actuales.

Algunas de las iniciativas investigadas son:

- Fundación Mustakis[1]: esta fundación promueve el desarrollo integral de las personas, por lo que dentro de sus actividades realiza talleres gratuitos de diversas áreas, como el arte o STEM. Dentro de estos talleres STEM, surgen los talleres de robótica, en los cuales colaboran con universidades como la Universidad de Talca[2] y la Universidad Técnica Federico Santa María. Estos talleres tienen una duración aproximada de un semestre y son impartidos por estudiantes de Ingeniería capacitados para enseñar los contenidos establecidos.
- MCI Capacitación[3]: MCI Electronics es una tienda especializada en el diseño, fabricación y comercialización de dispositivos electrónicos para prototipos rápidos. Ofrece diversos cursos y capacitaciones, los cuales son de pago. La duración de estos cursos suele ser de una sesión, y la mayoría se enfoca en la programación de microcontroladores.
- Kodland[4]: es una organización educativa y plataforma en línea que se dedica a enseñar programación y habilidades tecnológicas a niños y jóvenes. La misión de Kodland es fomentar el aprendizaje de programación y habilidades digitales de manera divertida y accesible para los estudiantes. Además, ofrece clases gratuitas.
- Semillero de Innovación[5]: es una comunidad que busca globalizar el conocimiento de nuevas tecnologías y, para esto, realiza cursos de robótica, programación y electrónica para niños a partir de los 10 años. Existen cursos tanto pagados como gratuitos, con una duración aproximada de 4 sesiones. Estos cursos se imparten en línea y utilizan la herramienta Tinkercad para la simulación de circuitos y programación.
- Sysmic[6]: iniciativa estudiantil en la cual participan estudiantes de la Universidad Técnica Federico Santa María. Su objetivo es diseñar y desarrollar robots capaces de jugar fútbol. Con este propósito, asisten a torneos de este deporte para competir. Agrupa

en sus filas estudiantes de todas las áreas: informática, electrónica, mecánica, industrial, etc.

- Innovación y robótica USM[7]: iniciativa estudiantil que reúne a estudiantes de ingeniería de la Universidad Técnica Federico Santa María, quienes trabajan en proyectos relacionados con robótica y electrónica. Dado que es una iniciativa que lleva muchos años funcionando, existe una rotación natural en sus participantes, lo que permite que muchos estudiantes adquieran experiencia en el proceso.

El trabajo desarrollado en esta memoria de título se enmarca en el contexto de las actividades desarrolladas por la iniciativa Academias de Tecnología (ACATEC) del Departamento de Electrónica de la Universidad Técnica Federico Santa María (UTFSM), la cual tiene su foco en la difusión y enseñanza de conceptos fundamentales de la electrónica y programación para estudiantes de enseñanza básica y media. Si bien las ACATEC se dividen en talleres para principiantes y avanzados, lo cual es definido de acuerdo a los conocimientos previos que tengan los estudiantes, esta memoria solo se enfocará en el taller para principiantes.

Específicamente, el presente informe reporta el proceso de investigación, desarrollo y generación de documentación para guías de actividades y rúbricas de evaluación para las sesiones que se realizan a lo largo de un año académico. La generación de estas guías apunta a superar una debilidad identificada en versiones anteriores de las ACATEC, asociada a la poca sistematización en el proceso de entrega de contenidos hacia los participantes y la falta de indicadores objetivos que permitieran medir el impacto efectivo de la iniciativa en los escolares que participan en ellas. Para el desarrollo de este trabajo se contó con el apoyo de estudiantes participantes de las ACATEC, académicos del Departamento de Electrónica y, además, con la supervisión de especialistas en docencia escolar.

Este capítulo presenta el contexto y la motivación para el desarrollo de esta memoria, sus objetivos, los alcances considerados para el trabajo y las principales contribuciones, finalizando con la descripción de la organización del resto del informe.

1.1. Contexto y motivación

En 2016 se creó ACATEC, una iniciativa del Departamento de Electrónica de la UTFSM, como parte del eje de Vinculación con el Medio. Esta iniciativa tiene como objetivo acercar la Ingeniería Electrónica a estudiantes de educación primaria y secundaria a través de talleres supervisados por estudiantes del Departamento, conocidos como tutores. Los talleres se organizan en sesiones semanales que se llevan a cabo durante todo el año escolar, generalmente en las instalaciones de las escuelas donde los estudiantes participantes, o tutorados, están matriculados. Los tutorados se agrupan en academias, donde cada academia está compuesta por estudiantes de un mismo establecimiento. Cada academia es dirigida por un grupo de tutores que se encarga de dictar las sesiones, coordinar las actividades administrativas y de gestión con el establecimiento y mantener una relación constante con los tutorados. Durante la ejecución del taller, se busca mantener el mismo grupo de tutores asignado al inicio del mismo para fortalecer los lazos entre tutores y tutorados a través del acompañamiento continuo. Es importante mencionar que cada academia cuenta con la presencia de un docente del establecimiento para supervisar el trabajo de los tutores y apoyar en las labores pedagógicas y

administrativas asociadas a los reglamentos internos de cada establecimiento. Al final del taller, se realiza una ceremonia de clausura donde se presentan los proyectos creados por grupos de tutorados de cada academia durante las últimas tres a cuatro semanas del taller, utilizando los conocimientos adquiridos durante el mismo.

1.2. Planteamiento del problema

Hasta el final del año 2019, las actividades de las ACATEC se definían en torno a objetivos muy generales establecidos al comienzo de cada año escolar. Básicamente, los tutores definían una temática general a tratar durante el año (por ejemplo, el diseño e implementación de un vehículo robot de exploración), en base a lo cual se establecían algunos objetivos generales y contenidos mínimos necesarios para revisar durante el año. El objetivo era que al final de las academias, los tutorados pudieran enfrentar el desarrollo de un proyecto de aplicación asociado a la temática definida. Luego, teniendo en cuenta el calendario académico de las instituciones, se establecía una planificación de las sesiones a realizar durante el año y se asignaba un tutor a cada academia participante. Una vez iniciadas las actividades, cada tutor preparaba y dictaba las sesiones para entregar los contenidos en base a su propio criterio y habilidad. Solo se requerían algunos informes de progreso para verificar que las sesiones se hubieran realizado y transmitido correctamente, sin proporcionar muchos detalles sobre los contenidos a entregar y evaluar. Por ejemplo, uno de los objetivos generales planteados al inicio de las academias podría haber sido que los tutorados aprendieran a utilizar un sensor de ultrasonido, pero no había una planificación o cronograma que especificara en qué momento del taller debían aprenderlo, cómo se abordaría este contenido ni cómo se evaluaría si efectivamente los tutorados aprendieron a usarlo.

El enfoque anterior permitía que los tutores ajustaran los contenidos a la cantidad de sesiones disponibles y se adaptaran a las necesidades e intereses particulares de los tutorados. Esto resultó positivo en términos de la percepción y valoración de la actividad por parte de los tutorados y docentes de cada academia, quienes la veían como una oportunidad para aprender disciplinas como electrónica y programación de forma divertida, siguiendo su propio ritmo y con clases personalizadas. Un indicador positivo en este aspecto fue el creciente interés en participar en estas actividades por parte de los tutorados y establecimientos educativos. Se registró un aumento consistente en el número de participantes año tras año, de 5 academias y 56 tutorados en 2016 a 10 academias y 123 tutorados en 2019, según las listas de asistencia.

Sin embargo, en 2019, el aumento del número de inscritos y la dificultad para coordinar actividades entre distintas academias, magnificada por las interrupciones derivadas de la contingencia del país¹, provocaron que durante la realización de la actividad de cierre se hicieran evidentes las diferencias en la percepción de los resultados de aprendizaje alcanzados. Lo anterior se debió principalmente a la cantidad de sesiones realizadas en cada academia y la comparación de los contenidos generales vistos por los tutores. Esta situación dejó en evidencia la necesidad de contar con una metodología sistemática que facilitara el seguimiento de los avances y la evaluación sistemática del nivel de aprendizaje alcanzado, así como la identificación oportuna de fortalezas y debilidades. Esto permitiría tomar decisiones a corto y mediano plazo para aumentar el impacto efectivo de la iniciativa.

¹Estallido social ocurrido el 18 de octubre 2018 en Chile

Si bien la improvisación no tenía resultados negativos, la percepción de tutorados satisfechos impedía medir avances concretos en el aprendizaje de contenidos. Además, se detectaron falencias en competencias de interés definidas por las ACATEC.

A inicios del 2020, la contingencia asociada a la pandemia (COVID-19) impidió que las actividades de las ACATEC se realizaran en su formato regular, es decir, mediante actividades presenciales en los establecimientos. Luego de un análisis del contexto, el equipo de coordinación de ese entonces vislumbró la oportunidad de utilizar esta incertidumbre de acción para preparar una nueva versión de las Academias, adaptándose a la situación de pandemia del momento y aprovechando de replantear la forma en que se entregaba el contenido y se medía el impacto de la iniciativa.

Así, tomando en cuenta que todas las actividades docentes en universidades y colegios pasaron a modalidad en línea, y explotando la experiencia adquirida en algunas asignaturas de laboratorio del Departamento de Electrónica de la UTFSM, se pensó en preparar clases en forma de experiencias guiadas, con un enfoque en entornos de simulación que permitieran a los estudiantes trabajar sin necesidad de acceder al equipamiento físico.

Después de un proceso de autoevaluación y diagnóstico realizado entre los tutores, académicos y personal administrativo que participaron en la iniciativa, se identificó que uno de los factores que inciden directamente en la debilidad detectada es evitar que los tutorados vean las actividades como amenazantes o frustrantes. Por esta razón, se evitó la realización de evaluaciones y se optó por aligerar los contenidos en caso de que los tutorados se vieran sobre exigidos. Sin embargo, quedó claro que la única forma de reducir la subjetividad y poder medir el impacto efectivo de la iniciativa era mediante la sistematización del proceso de enseñanza y la realización de evaluaciones para medir el impacto, sin perder la esencia de las actividades de las ACATEC: una forma divertida y amena de difundir aspectos de ciencia y tecnología a la población escolar.

1.3. Objetivos

En la presente memoria de título se aborda el proceso de creación de material guía para su uso en experiencias en línea. Para ello, se contó con el apoyo de estudiantes y académicos del Departamento de Electrónica, además de la supervisión de especialistas en docencia de la Dirección de Vinculación con el Medio de la UTFSM.

Si bien este material fue diseñado para su ejecución en modalidad remota, es importante mencionar que el proceso de creación de material e instrumentos de medición de aprendizaje puede ser adaptado posteriormente a actividades presenciales.

Uno de los desafíos en el diseño de las sesiones fue asegurarse de que estas fueran completas en cuanto a contenido, didácticas y entretenidas, todo al mismo tiempo. De esta manera, se buscó captar la atención de los tutorados y fomentar su participación en el proceso de aprendizaje.

El objetivo de esta memoria de título es documentar el proceso de creación de material guía para la ejecución de sesiones de las ACATEC en modalidad en línea. Este material aborda contenidos específicos sobre electrónica y programación, ambos de nivel básico, e incluye

instancias de desarrollo y refuerzo de habilidades blandas, como el trabajo en equipo, la creatividad, la innovación, el pensamiento crítico, la resiliencia, el liderazgo y la capacidad de discurso. Además, se añaden instrumentos de medición para cada sesión, lo que permite detectar fortalezas y debilidades para mejorar continuamente el proceso.

Los objetivos planteados para este trabajo de título se muestran en los siguientes puntos:

- Sistematizar las experiencias de tal manera que cualquier tutor que se una a la iniciativa pueda ofrecer la misma calidad de contenido que aquellos con más experiencia en ella.
- Estructurar clases que entreguen los contenidos de forma didáctica y entretenida para los tutorados, a un bajo costo y respaldadas por expertos en pedagogía.
- Elaborar instrumentos que permitan medir el impacto de las ACATEC en cuanto a aprendizaje en los tutorados.

A continuación se introducen las etapas generales de la solución, a modo de introducción de la misma:

1. Investigación y reconocimiento del problema.
2. Selección de contenidos a entregar y determinación de orden en que estos se entregarán durante el taller.
3. Búsqueda de herramienta educativa que apoye el aprendizaje de electrónica y programación, orientada a niños.
4. Diseño de guías.
5. Ejecución de experiencias en su primera versión y recolección de realimentación de esta.
6. Mejora de experiencias a través del trabajo en conjunto con expertos en educación.
7. Incorporación de realimentación recolectada a través de la primera ejecución.

Con lo anterior se espera lograr una mejora continua como iniciativa en cuanto al rol social y al impacto medible en los tutorados y en la Universidad.

1.4. Alcances

A continuación se listan los alcances de esta memoria:

- Debido a la contingencia de la pandemia, se trabajó en el diseño de sesiones con contenidos de electrónica y programación que puedan ser realizadas en formato en línea, logrando así retomar el funcionamiento de las ACATEC.
- La herramienta de apoyo para el diseño de las sesiones será Tinkercad, una aplicación web gratuita dedicada al diseño 3D y circuitos. Además, se utilizarán dos plataformas de apoyo para facilitar la comunicación y el flujo de información: Zoom y Google Classroom.
- Se considera que cada sesión del taller tendrá una duración de ciento cinco minutos, repartiéndose este tiempo en dos bloques de cuarenta y cinco minutos (una hora pe-

dagógica equivale a cuarenta y cinco minutos), separados por un intermedio de quince minutos de descanso.

- Se considera el diseño de ocho sesiones que abarquen la totalidad de los contenidos planificados para el curso. En esta memoria se diseñará la sesión N°0, la cual corresponde a una evaluación de diagnóstico inicial. Cada sesión contará con un instrumento de evaluación diseñado en conjunto con una experta en pedagogía.

1.5. Contribuciones

La contribución más importante de esta memoria radica en la documentación de la estandarización del proceso de enseñanza y creación de material para las ACATEC, además de desarrollar instrumentos de evaluación que permitan evaluar de manera objetiva los aprendizajes alcanzados por los participantes [8].

Los entregables de esta memoria incluyen guías con los contenidos básicos de electrónica y programación que las ACATEC desean impartir, así como rúbricas que contienen dimensiones e indicadores relevantes para evaluar los objetivos planteados al inicio de cada sesión.

1.6. Organización del informe

Este documento se encuentra constituido de la siguiente manera:

- **Capítulo 1 - Introducción:** Se presenta la motivación, planteamiento del problema, los alcances del trabajo de título y la organización propia del informe.
- **Capítulo 2 Antecedentes: Exploración de herramientas y técnicas modernas:** Se presentan trabajos previos al presente trabajo de título, además de exponer los fundamentos para la elección del conjunto de herramientas utilizado en la generación del material.
- **Capítulo 3 - Diseño de experiencias y métodos de evaluación :** Se expone el trabajo de implementación paso a paso, acompañado de un análisis en cada una de las etapas de trabajo.
- **Capítulo 4 - Resultados:** Presentación de los resultados e insumos obtenidos.
- **Capítulo 5 - Conclusiones :** Planteamiento de las conclusiones generales acerca del trabajo realizado sobre el laboratorio y propuestas de trabajo que se pueden desarrollar para la continuidad del proyecto.

2 | Revisión de herramienta Tinkercad de simulación para aprendizaje de electrónica y programación en línea

En este capítulo se presentan las alternativas de herramientas de simulación. A partir de ellas, se seleccionará una como apoyo para la enseñanza de los contenidos en las sesiones. Además, se incluyen los razonamientos que llevaron a tomar esta elección. Finalmente, se abordará en detalle la herramienta seleccionada, Tinkercad, y se destacarán sus características más importantes para su uso en las sesiones proyectadas.

2.1. Herramientas de simulación

La presencialidad ofrece posibilidades y ventajas conocidas, como la interacción del tutorado con un entorno físico con componentes electrónicos y otras herramientas. Tomando esto en cuenta, se debía encontrar una herramienta en el campo de la electrónica y programación que, en comparación con una herramienta física, siguiera siendo interesante para los tutorados. La simulación de circuitos posee la ventaja de permitir trabajar directamente con la disciplina y observar resultados realistas, pero sin la necesidad de contar con un entorno físico tangible.

A continuación, se mencionan algunas características relevantes para esta investigación de las herramientas de simulación de circuitos analizadas en esta memoria, así como sus ventajas y desventajas con respecto a lo requerido.

Las tres herramientas investigadas fueron: LTSpice [9], EveryCircuit [10] y Tinkercad [11]. Todas ellas fueron diseñadas para ser utilizadas en dispositivos como computadoras, teléfonos inteligentes o tabletas, aunque no todas están disponibles para los tres tipos de dispositivos mencionados.

La primera herramienta investigada fue LTSpice, un software profesional de diseño y simulación de circuitos electrónicos disponible solo para computadoras. Sus ventajas son:

- Posee una gran variedad de componentes, los cuales son configurables.

- Permite visualizar señales como corriente y voltaje en el tiempo a través de gráficos, lo cual es útil para comprobar la parte teórica de la electrónica.
- Es gratuito.
- Puede utilizarse sin conexión a internet.
- Tiene una interfaz amigable.

Sus desventajas son:

- La visualización de señales puede resultar difícil para el público objetivo de las ACATEC.
- No posee funcionalidades que permitan interacción entre usuarios.

EveryCircuit [12], por otro lado, es una aplicación para tablets y smartphones que permite construir, visualizar y simular una gran variedad de circuitos. Sus ventajas son:

- Permite observar el camino físico de la corriente en el circuito de forma interactiva y en tiempo real.
- Tiene una interfaz amigable e intuitiva, lo que facilita la interacción con los usuarios.
- Cuenta con una gran comunidad de usuarios que interactúan en foros.

Sus desventajas son:

- Es una aplicación de pago, aunque tiene una versión gratuita de un día.
- No posee una función que permita la interacción entre usuarios en el mismo circuito.
- Está dirigida a un público con una edad superior a nuestro objetivo.

Finalmente, encontramos Tinkercad, una plataforma dedicada al diseño 3D, electrónica y programación, disponible para computadoras, tabletas o teléfonos inteligentes. Sus ventajas son:

- Fue diseñada para el uso de niños, por lo que su interfaz es llamativa e intuitiva. Además, las componentes electrónicas disponibles tienen un aspecto realista.
- Cuenta con dos herramientas adicionales además del diseño de circuitos: diseño en 3D y programación en bloques.
- Es gratuita.
- Permite crear una clase a través del registro de correos electrónicos de los participantes
- Tiene funciones que permiten la interacción entre usuarios y el trabajo colaborativo.

Sus desventajas: basándonos en los criterios establecidos, no se encontraron desventajas significativas que deban ser consideradas en la evaluación.

2.2. Tinkercad

La plataforma Tinkercad es amigable para niños y jóvenes que se inician en la electrónica. Además, es gratuita y ofrece cursos de diseño 3D, circuitos y programación en bloques a través de proyectos. Una de sus grandes ventajas es la forma en que presenta los componentes, ya que son imágenes con aspecto real, lo que facilita la conexión entre el mundo real y la plataforma.

Esta plataforma fomenta la creatividad gracias a la amplia variedad de herramientas y su facilidad de uso. Incluso en la página oficial de Tinkercad, se encuentran frases como *deja volar tu imaginación* o *los innovadores del mañana se forman hoy*, las cuales explican el objetivo que busca cumplir esta plataforma. Además, garantiza la seguridad de los contenidos disponibles, lo que significa que no incluye publicidad ni contenido inapropiado para niños. Asimismo, la plataforma facilita en gran medida el trabajo en equipo, permitiendo la creación de clases donde los alumnos y el profesor pueden ingresar y colaborar en proyectos grupales o individuales.

A continuación, en esta sección se profundizará en las características potenciales que hacen de esta herramienta una ayuda en el proceso de aprendizaje de las ACATEC.

2.2.1. Entorno visual representativo

Para hacer uso de Tinkercad, es necesario tener una cuenta de Google, la cual se utilizará para registrarse como nuevo usuario. Una vez creada la cuenta en Tinkercad, se accede al panel principal, donde se pueden observar tres secciones a la izquierda: Clases, Diseños y Aprendizajes. Al ingresar a la sección "Diseños", aparecen nuevamente tres opciones de trabajo:

1. Diseño 3D

Esta opción permite crear composiciones de figuras geométricas con volumen, tamaño y color. Estas figuras se colocan en un plano tridimensional donde se pueden modificar las posiciones según se desee. Además, la herramienta proporciona una vista fluida de todos los ángulos de la composición creada, lo que ayuda a desarrollar las habilidades espaciales del usuario.

La figura 2.1 muestra algunas de las figuras disponibles para seleccionar en la sección de diseño 3D. Aunque todas tienen un color predeterminado al agregarlas, es posible modificarlo, como se muestra en la figura 2.2, donde se puede elegir el color deseado, tamaño y posición para la esfera.

En la figura 2.3 se muestra una composición de varias figuras ubicadas en diferentes lugares de un plano tridimensional. Estas pueden ser visualizadas desde diversos ángulos y distancias según sea necesario.



Figura 2.1: Algunas opciones de figuras disponibles para el diseño.

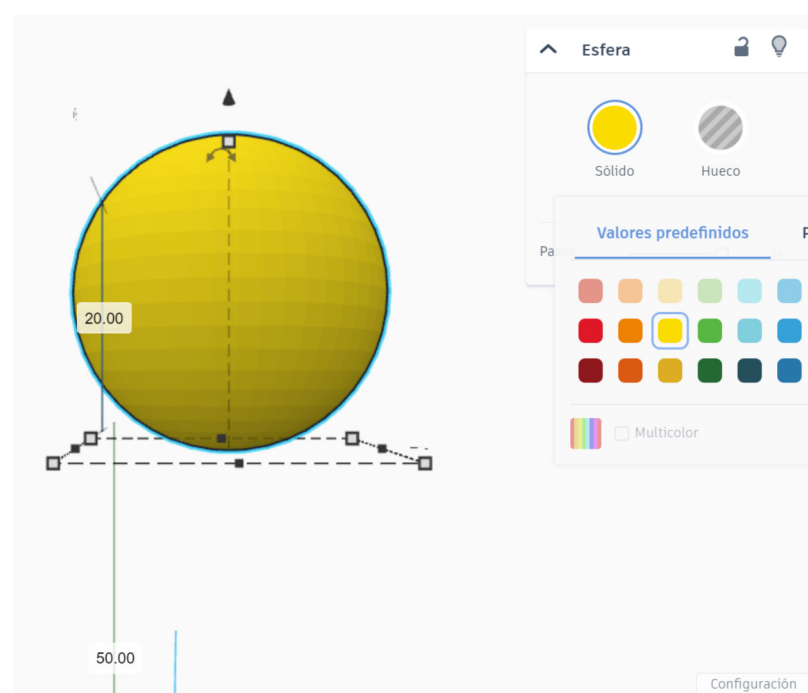


Figura 2.2: Opciones de color, tamaño y posición de forma.

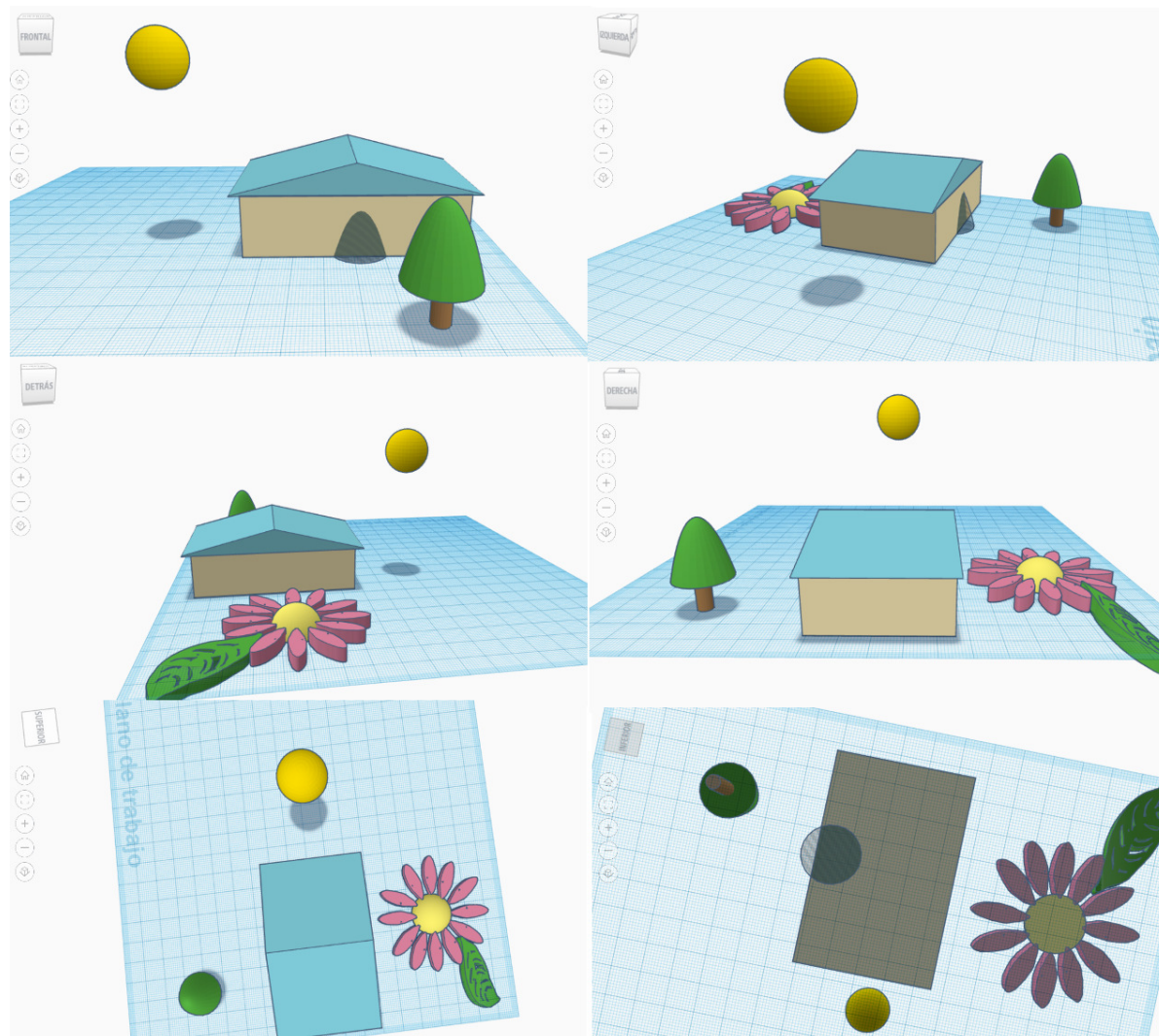


Figura 2.3: Vistas en 3D de una composición.

2. Circuitos

La función de circuitos en Tinkercad es lo que hizo que se considerara como una herramienta para el desarrollo de material para las ACATEC, ya que contiene todos los componentes necesarios para el diseño y simulación de circuitos electrónicos. Además, permite la programación de los circuitos diseñados mediante texto, bloques o una combinación de ambos.

La herramienta de simulación incluye prototipos de microcontroladores como Arduino, componentes electrónicos, sensores y actuadores. Estos se encuentran disponibles en una biblioteca dentro de la misma pantalla de diseño. Además, se cuenta con herramientas como osciloscopios, amperímetros, voltímetros y otras, que permiten visualizar variables eléctricas como corriente y voltaje.

Es importante destacar que los componentes disponibles en la simulación son representativos y se asemejan al aspecto físico real de los componentes correspondientes. Por ejemplo, un LED en la simulación se muestra de la misma manera en la que se vería físicamente un LED, y lo mismo ocurre con todos los demás componentes.



Figura 2.4: Algunas opciones de componentes.

En la figura 2.4 se puede apreciar que cada componente cuenta con una imagen representativa que se asemeja mucho a su apariencia real. Esta representación realista se mantiene durante el proceso de diseño del circuito, como se muestra en la figura 2.5. En la parte superior izquierda se muestra la vista predeterminada del diseño con gráficos reales de los componentes. En la esquina superior derecha se encuentra la vista esquemática del circuito, y finalmente, en la parte inferior de la imagen se encuentra una tabla con la lista de componentes que forman parte del diseño.

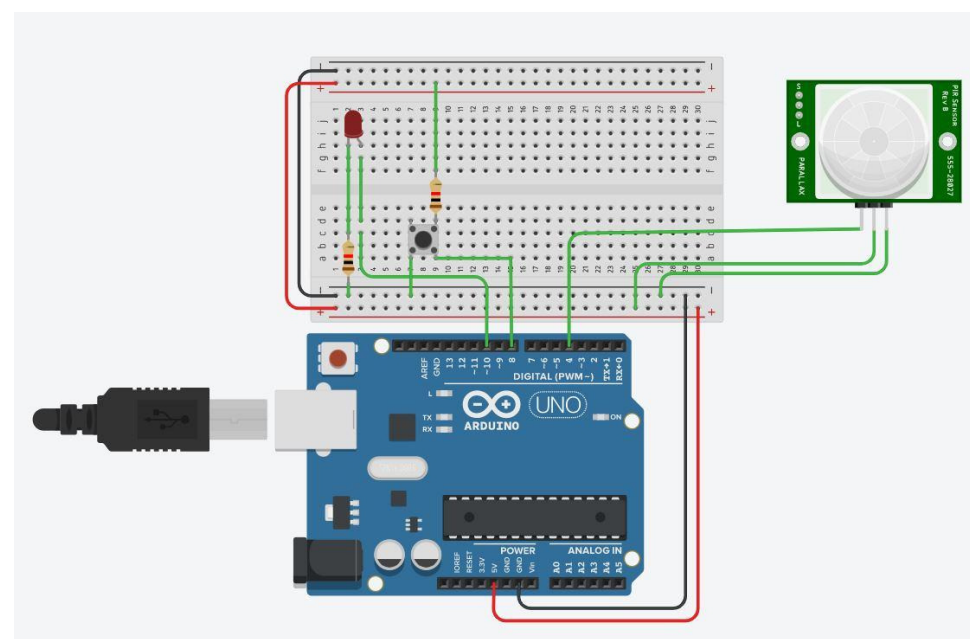


Figura 2.5: Circuito simple de ejemplo.

En Tinkercad, se encuentran disponibles tres tipos de vistas que representan el circuito diseñado: la vista del circuito, la vista esquemática y la lista de componentes. Estas tres vistas se ilustran en la figura 2.6. La lista de componentes se presenta como una tabla que detalla todos los componentes utilizados en el diseño del circuito.

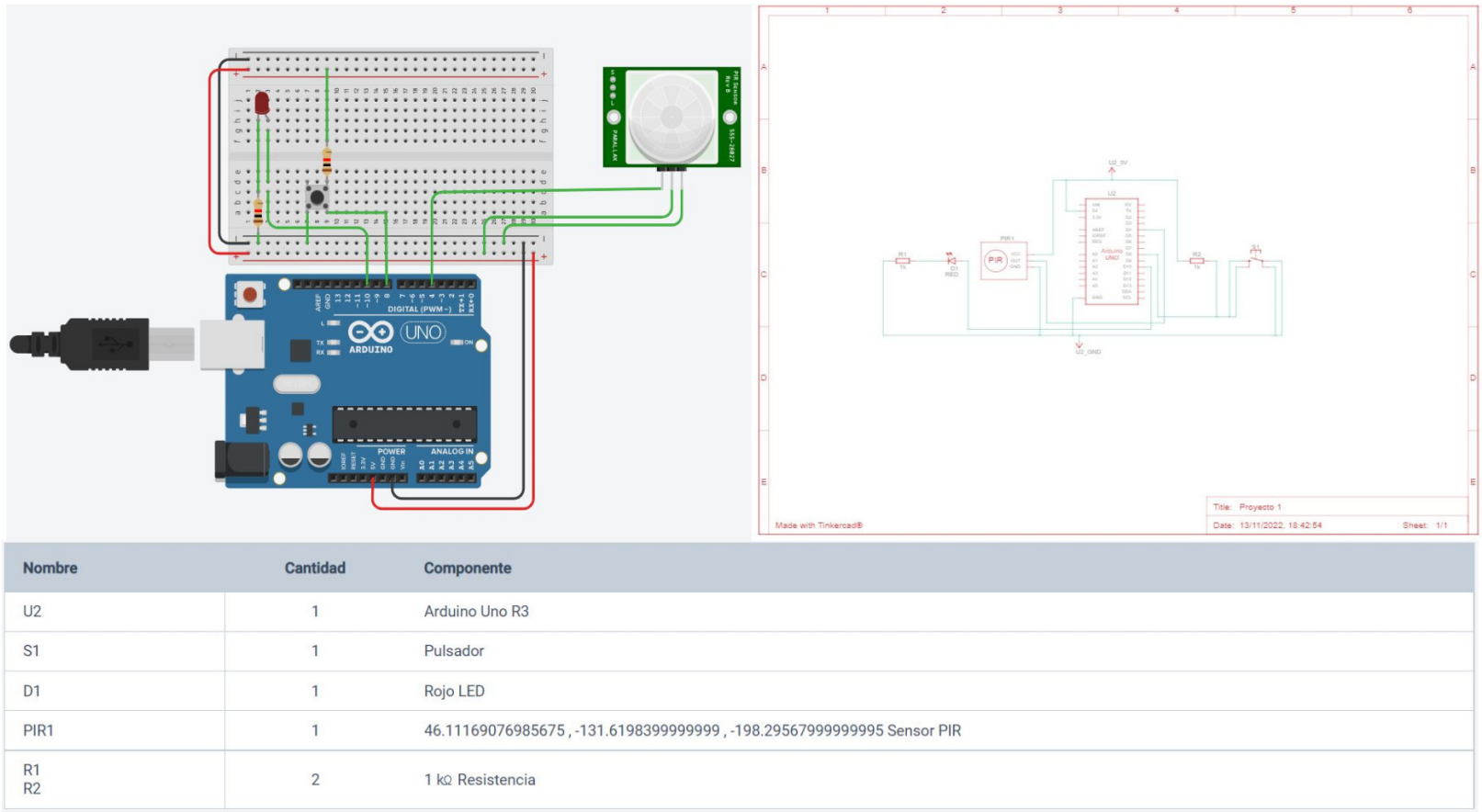


Figura 2.6: Vistas disponibles para el diseño de un circuito.

3. Bloque de código

En la opción de bloques de código de Tinkercad, es posible crear algoritmos utilizando formas volumétricas a través de la opción de diseño en 3D. Los bloques de código disponibles permiten trabajar con las formas en 3D seleccionadas o con variables aleatorias creadas para usos específicos. Por ejemplo, se pueden utilizar bloques de código para mover una figura a una posición diferente en el plano (X, Y, Z) en relación a su posición inicial.

En la figura 2.7 se muestra una vista general de algunos bloques de código disponibles en la biblioteca, los cuales se dividen según su uso en: formas (azul), modificar (morado), controlar (naranja), matemáticas (verde), datos (celeste) y marcar (negro). Por otro lado, la figura 2.8 presenta un algoritmo simple para crear una fila de árboles similar a la de la figura 2.9. Para lograr esto, se utilizan bloques de diferentes colores según su función específica.

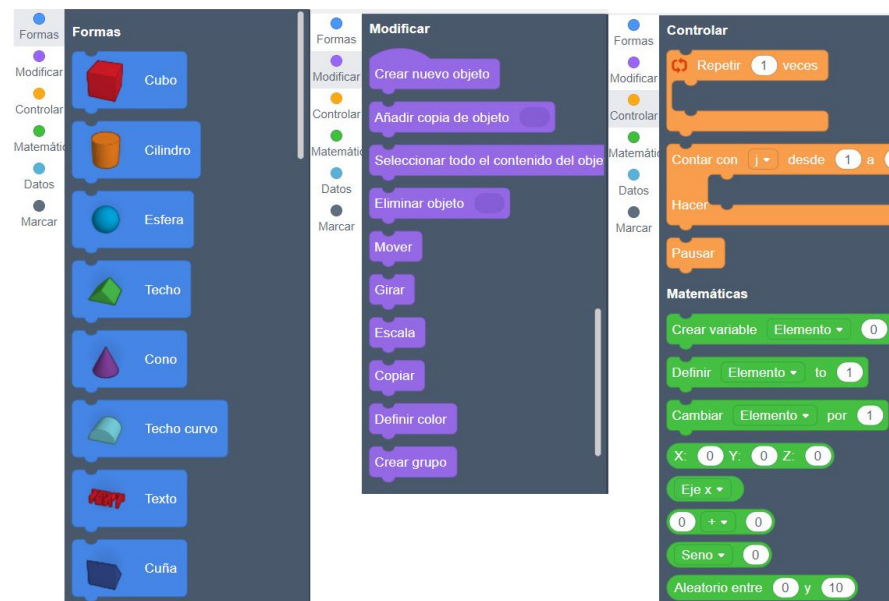


Figura 2.7: Vista de parte de los bloques de código disponibles en la biblioteca.

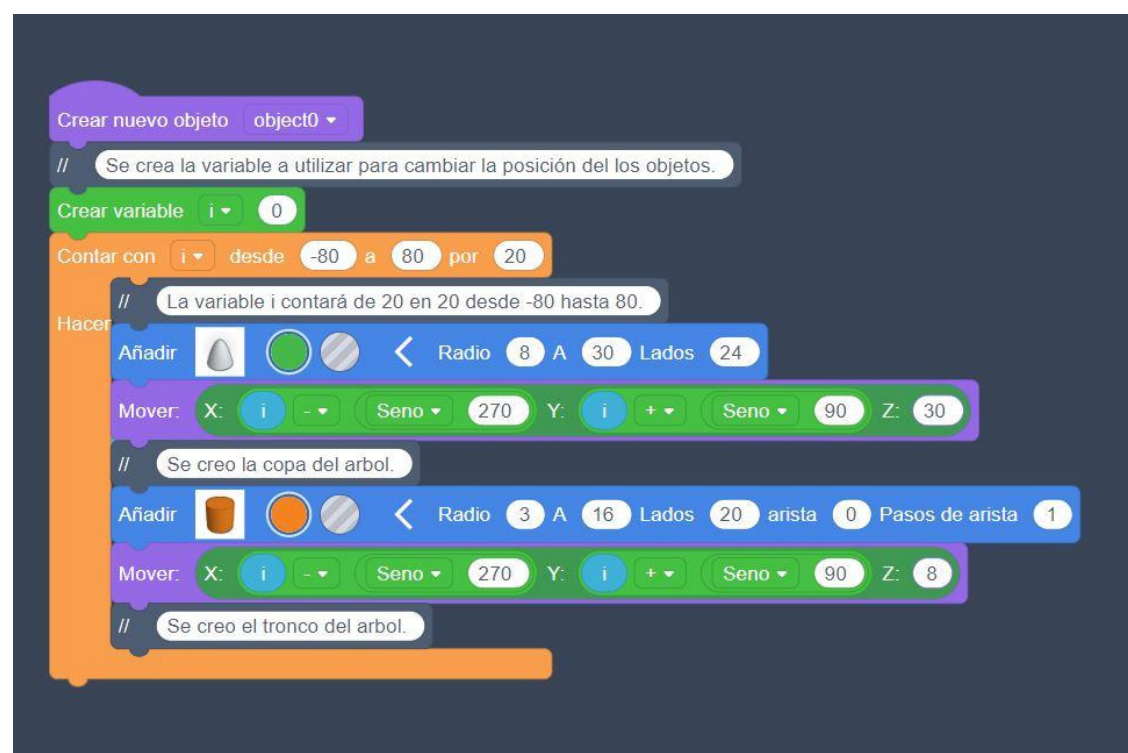


Figura 2.8: Algoritmo con bloques de código.

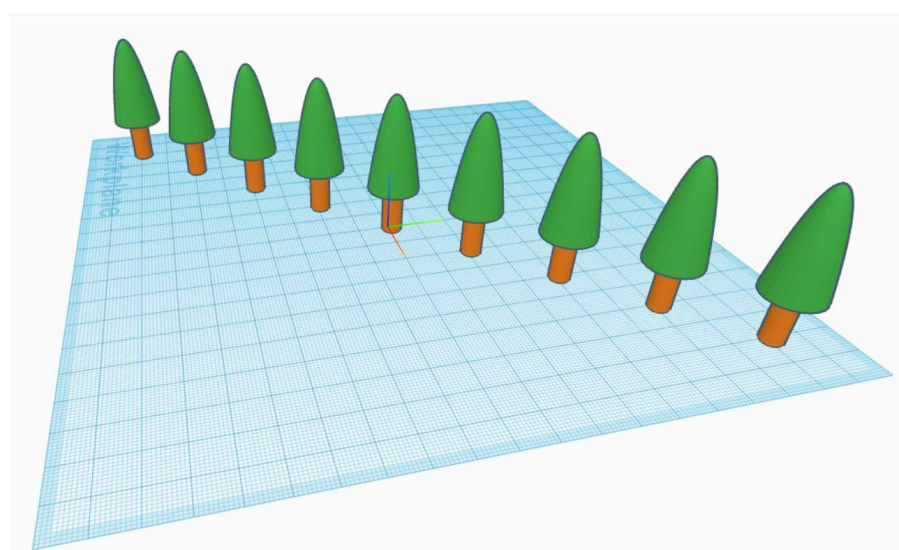


Figura 2.9: Resultado de la simulación del algoritmo anterior desde un ángulo cualquiera.

2.2.2. Errores controlados

A continuación, se desarrollará con más detalle la opción de diseño de circuitos, que está relacionada con las disciplinas de interés para esta memoria: electrónica y programación. Esta opción cuenta con un simulador que se ejecuta al presionar un botón que aparece en la pantalla principal del proyecto en diseño. Al presionar este botón, comienza a correr el tiempo de simulación y se muestra como un cronómetro junto al mismo botón. Para detener la simulación, basta con volver a presionarlo.

Cabe destacar que el botón de simulación no solo funciona cuando se ha programado un microcontrolador, sino también para cualquier experimento realizado en el proyecto, por ejemplo, conectar una pila a un LED.

En Tinkercad, se pueden implementar errores controlados en los circuitos diseñados para simular situaciones específicas. Estos errores controlados permiten evaluar el comportamiento de los circuitos en condiciones adversas y observar cómo reaccionan frente a ellos. No está disponible en la web la cantidad total de advertencias existentes en la plataforma, pero la más común es la sobrecarga de electricidad circulando a través de un LED. Hay dos casos comunes: la advertencia sobre la vida útil del LED y la advertencia sobre la quema del componente debido a una corriente excesiva. Se muestran ejemplos en las figuras 2.10 y 2.11.



Figura 2.10: Símbolo de led quemado por exceso de corriente.

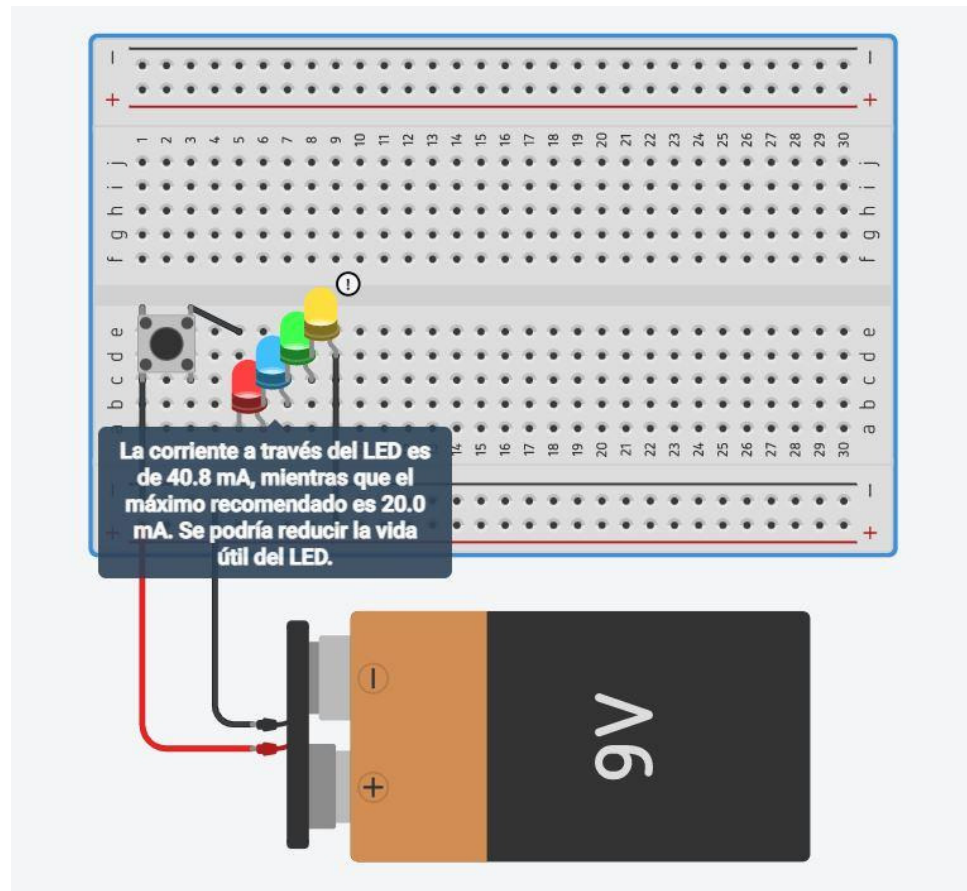


Figura 2.11: Advertencia de disminución de vida útil por sobrepasar corriente recomendada para un led.

Los errores controlados en Tinkercad ofrecen un entorno seguro para el aprendizaje, evitando riesgos en experimentos con componentes reales. Ayudan a los estudiantes a comprender las respuestas de los circuitos ante situaciones no ideales y a desarrollar habilidades de resolución de problemas, fomentando un aprendizaje práctico y seguro.

2.2.3. Uso de Arduino

La elección de Tinkercad como herramienta para las ACATEC permitió seleccionar el microcontrolador Arduino UNO R3 (figura 2.12) como el dispositivo principal para el curso. Este microcontrolador ofrece facilidad de uso y es compatible con las componentes disponibles en la biblioteca de Tinkercad. Además, es posible utilizar múltiples microcontroladores en un mismo proyecto y programarlos de forma simultánea. Esta opción brinda flexibilidad para el diseño y la programación de circuitos, enriqueciendo la experiencia de aprendizaje de los estudiantes y fomentando su capacidad de innovación tecnológica.

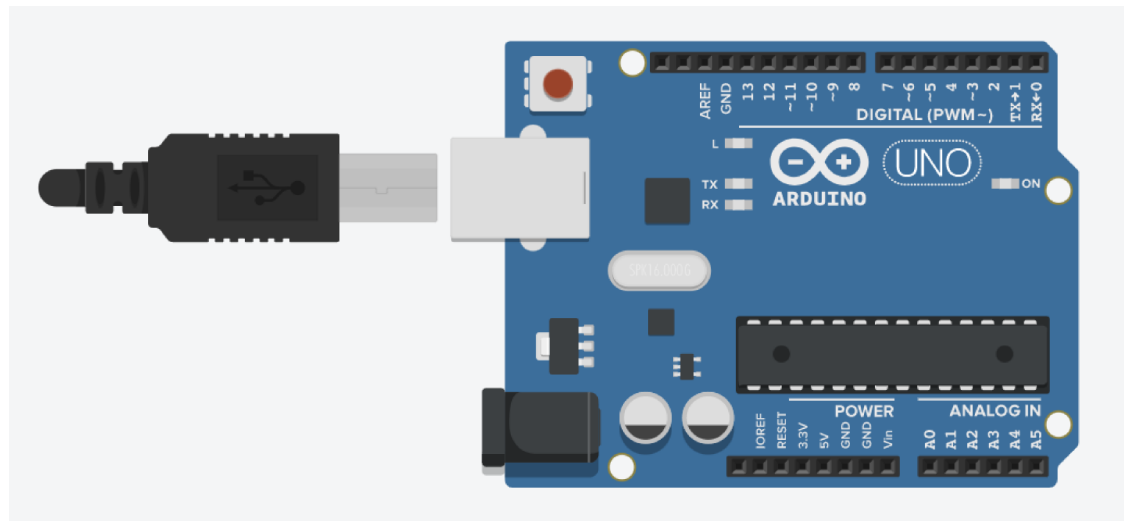


Figura 2.12: Arduino UNO R3.

2.2.4. Uso de código

El editor de código de la figura 2.13 se encuentra al costado derecho del diseño del circuito, se puede acceder a este a través de un botón que dice *código*. En la barra superior del editor se pueden hacer varias elecciones, la primera es qué dispositivo se programará, las opciones disponibles son los microcontroladores utilizados en el diseño, la segunda opción es el tamaño de la fuente con la que se podrá visualizar el código en el editor, la tercera es una opción para incluir bibliotecas específicas para el uso de ciertos componentes en los microcontroladores, la cuarta sirve para descargar el código escrito y finalmente la quinta para escoger de qué forma se programará, si a través de texto, bloques de código o mixto. Existe la forma de transformar bloques de código a texto y viceversa, pero no es recomendable ya que se ha probado y no es precisa la transformación. Se recomienda utilizar uno de los dos modos de programación de forma consistente en lugar de realizar conversiones entre ellos.

Texto



1 (Arduino Uno R3)

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14 }
```

Figura 2.13: Editor de código.

Cabe destacar que el editor de código en Tinkercad presenta una interfaz visual muy similar al entorno de desarrollo integrado (IDE) oficial de Arduino. Esta similitud puede ser de gran ayuda para los usuarios que deseen migrar a la programación de un Arduino real, ya que les resultará familiar y les facilitará la transición.

2.2.5. Función de clase

En el panel principal de Tinkercad, se encuentra la sección *Clases*, donde se puede optar por crear una clase como docente o encargado, o unirse a una clase como estudiante. Como ACATEC se enfoca en crear una clase como encargado o docente, se selecciona esta opción y se crea la clase. Para ello, solo se requiere ingresar el nombre y el alias de cada estudiante, junto con un código proporcionado por Tinkercad, que les permite acceder a la clase. Como docente o encargado (se puede asignar más de uno), se tiene la ventaja de poder realizar actividades y revisar los proyectos de circuitos, modelos 3D y programación en bloques directamente desde esta función (figura 2.15). De esta manera, se tiene acceso organizado a toda la información, lo que facilita realizar modificaciones y corregir los proyectos de forma individual.



Figura 2.14: Vista panel principal Clases.

En la figura 2.15, se puede observar que al acceder a Aula ACATEC se muestra una ventana con la lista de estudiantes, donde se encuentran sus nombres y alias. Además, se encuentra una casilla de búsqueda por nombre y un código de clase. El código de clase, junto con el alias, son utilizados por los estudiantes para acceder a través de la ventana de inicio de sesión de Tinkercad.

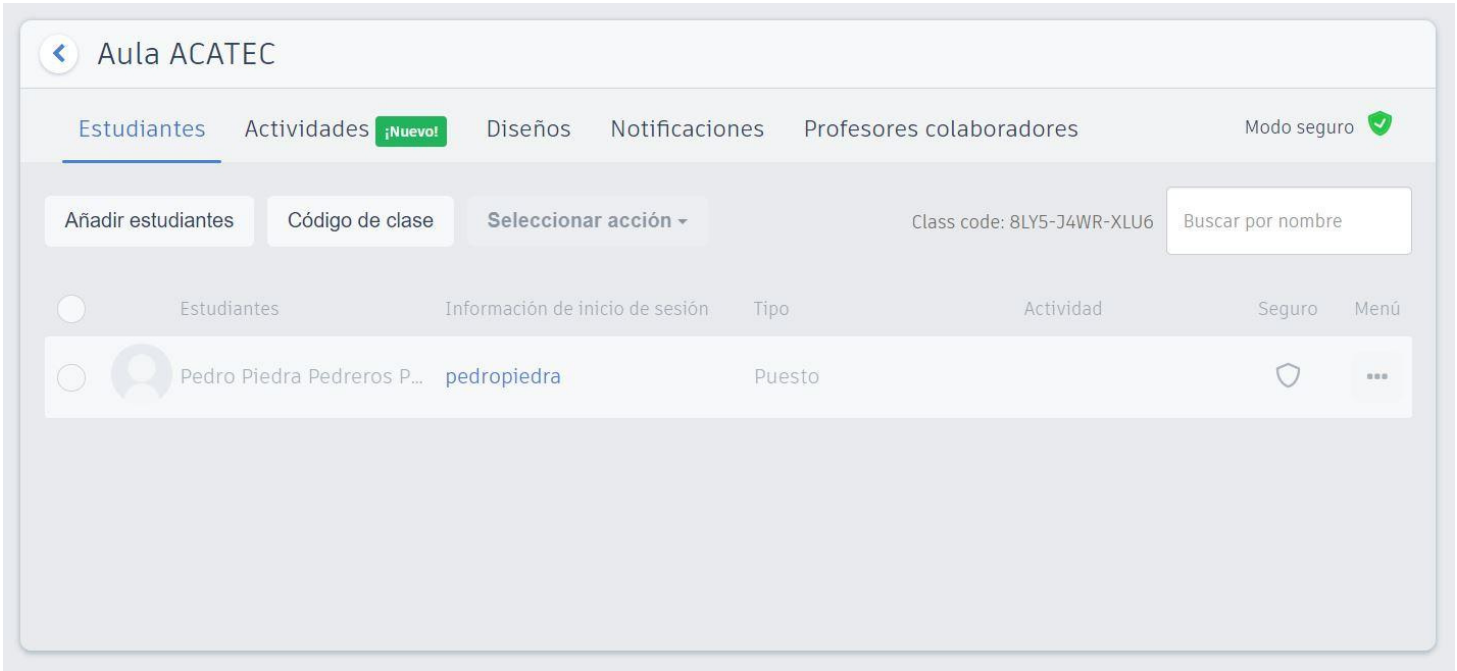


Figura 2.15: Vista luego de acceder a Aula ACATEC.

En la figura 2.16, se ilustra el proceso de inicio de sesión para estudiantes de una clase en Tinkercad. En el flujo de izquierda a derecha, se muestra que los estudiantes solo necesitan proporcionar su alias y el código de clase para acceder, sin necesidad de contar con una cuenta en Google u otro proveedor. Esta característica hace que la plataforma sea más segura y fácil de usar para los estudiantes.



Figura 2.16: Vista pantallas de inicio de sesión como estudiante de una clase.

3 | Diseño de experiencias y métodos de evaluación

Este capítulo se enfoca en las etapas llevadas a cabo en el diseño de las experiencias, así como en el diseño de los instrumentos de evaluación correspondientes. Se describe el flujo de trabajo seguido para cada guía, incluyendo las dificultades encontradas y las decisiones tomadas para alcanzar el objetivo establecido. Mediante este proceso, se logró sistematizar el diseño de las experiencias, lo cual es el objetivo principal de esta memoria. Además, se documenta este proceso como referencia para futuras mejoras, si fuera necesario.

3.1. Descripción del flujo de trabajo de esta memoria

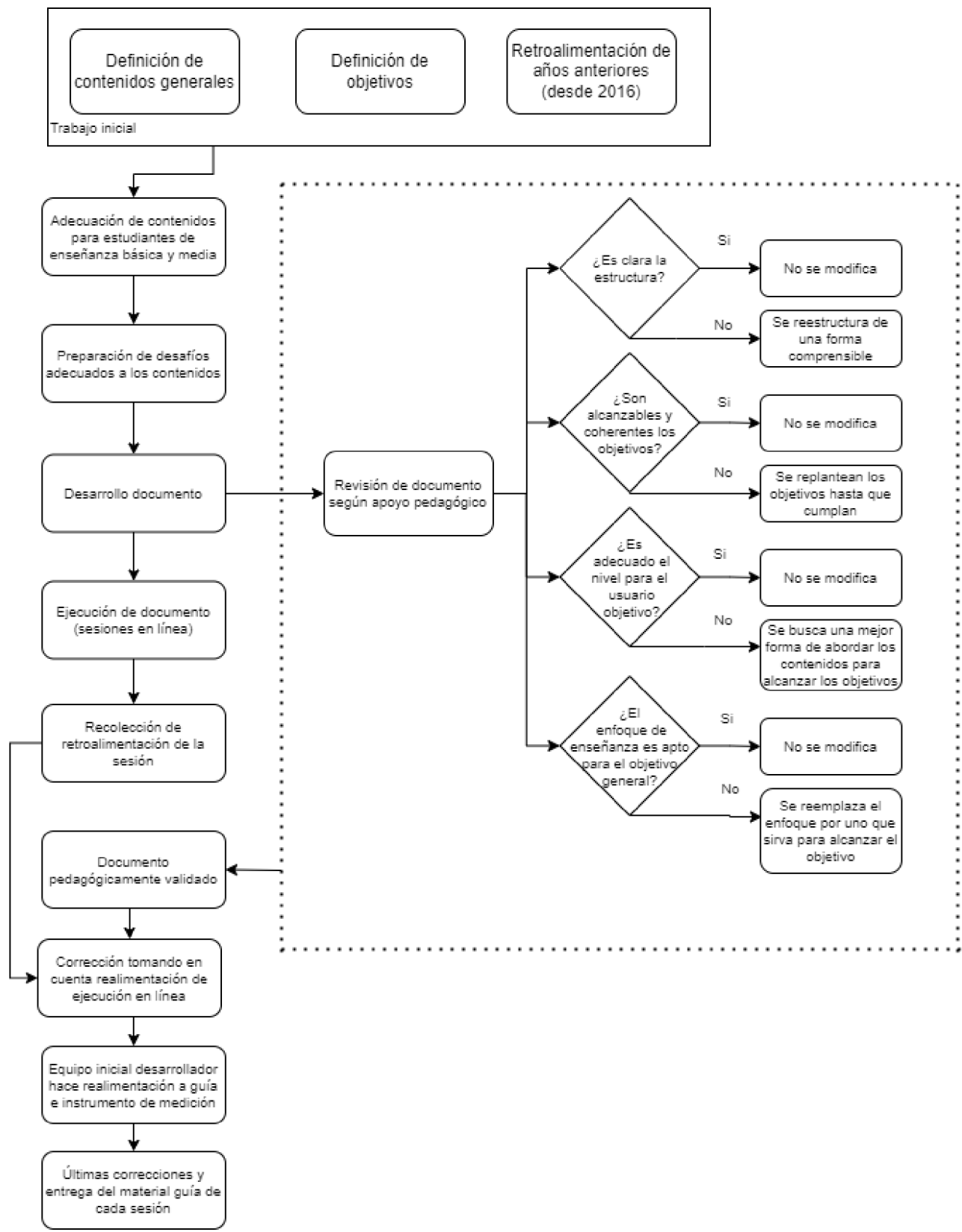


Figura 3.1: Esquema de trabajo utilizado en la generación de experiencias

En esta sección, se explica de manera general cada una de las etapas involucradas en la generación de material para experiencias de electrónica orientado a estudiantes de educación básica y media.

Se presentan los actores clave involucrados en el proceso. El primer actor es un grupo de estudiantes del Departamento de Electrónica, que dentro de las ACATEC toman el rol de desarrollar material para las sesiones a realizar. El siguiente actor involucrado es el docente encargado de las ACATEC, el cual brinda apoyo en conocimientos de electrónica y experiencia respecto a la gestión de las ACATEC. Finalmente, la Asesora Pedagógica, una profesional con Magíster en Evaluación Educacional, forma parte del Area de Vinculación con el Medio de la UTFSM, y su labor es crucial para la validación pedagógica de las experiencias desarrolladas y sus respectivos instrumentos de evaluación.

Las experiencias son diseñadas con el objetivo de entregar conocimiento teórico y práctico en disciplinas como electrónica y programación, de nivel básico ambas, además de habilidades blandas mediante la resolución de actividades y desafíos en forma grupal o individual.

El procedimiento de trabajo se representa como un diagrama de flujo en la Figura 3.1, y el resto del capítulo presenta una breve descripción de las etapas principales involucradas en este proceso.

3.1.1. Trabajo inicial

El trabajo inicial comprendió reuniones en las que participaron, en primera instancia, las ACATEC (todos los equipos que las componen), con el objetivo de recopilar información para realizar un análisis FODA. Esto permitió definir objetivos y áreas de mejora como parte de la iniciativa. Posteriormente, el equipo encargado de generar el material y el docente responsable de las ACATEC se reunieron para llevar a cabo la reestructuración mediante el diseño y mejora de las experiencias. Durante estas sesiones, se abordaron los siguientes temas:

- **Retroalimentación de años anteriores**

Se llevó a cabo una reunión general en la que se reconsideraron los objetivos, la misión y la visión de las ACATEC. Los asistentes, en su mayoría estudiantes de la UTFSM con más de un año de participación en las ACATEC, aportaron su amplia perspectiva sobre el desarrollo de la iniciativa. Además, se realizó una recopilación de información en relación a las disciplinas a impartir, considerando que actualmente existen numerosos talleres que enseñan programación, robótica, entre otros. Fue necesario decidir si la orientación de los contenidos de las ACATEC seguiría centrándose en la electrónica y la programación.

- **Definición de contenidos generales**

Tras la decisión de continuar impartiendo disciplinas como electrónica y programación, se procedió a consultar al equipo de las ACATEC sobre los contenidos clave en un curso de estas temáticas, según su criterio. Luego de definir los contenidos básicos, se llevó a cabo una labor conjunta entre el docente encargado de las ACATEC y el equipo de generación de material para organizarlos de manera coherente, estableciendo un orden progresivo en el avance del curso. Además, se acordó dividir los contenidos en sesiones

de similar complejidad, con una duración aproximadamente equivalente entre sí. En base a esto, se determinaron seis sesiones que abarcarían los contenidos básicos establecidos.

■ **Definición de objetivos**

En el diseño de cada sesión se estableció una progresión lógica de contenidos, de manera que cada uno estuviera construido sobre lo aprendido en la sesión anterior. Se determinó qué herramientas y competencias debían abordarse en cada sesión para garantizar un progreso fluido hacia la siguiente. Con este fin, se definieron objetivos claros para cada sesión, los cuales sirvieron como guía para que el tutor enfocara sus esfuerzos en lograrlos, ya que el éxito de la sesión siguiente dependía de ello. Este proceso de diseño se llevó a cabo en colaboración con el equipo encargado de generar el material.

3.1.2. Adecuación de contenidos para estudiantes de enseñanza básica y media

Se tuvo en cuenta el nivel de conocimiento previo de cada estudiante al abordar los contenidos acordados, reconociendo que no todos tendrían el mismo nivel. Se realizó una adaptación considerando las observaciones de años anteriores. Además, se puso especial énfasis en la forma en que se presentaron las actividades y se redactaron las guías, buscando que fueran simples, intuitivas y entretenidas para los estudiantes.

3.1.3. Preparación de desafíos adecuados a los contenidos

Se diseñaron desafíos no evaluados, incluyendo entre 1 y 3 desafíos por sesión, dependiendo de la extensión de la actividad. Aunque inicialmente no se asignó un bloque específico para la ubicación de los desafíos, este detalle adquiere importancia en las posteriores modificaciones realizadas en colaboración con el apoyo pedagógico.

3.1.4. Desarrollo del documento

Con base en el trabajo previo, se llevó a cabo la redacción y desarrollo de los documentos de cada sesión. Se crearon versiones tanto para los estudiantes como para los tutores, con el objetivo de proporcionar enfoques específicos que fueran útiles para cada usuario. En la guía destinada a los tutores se incluyó información relevante sobre el manejo de herramientas como Zoom, Teams o Tinkercad. Por otro lado, la guía de los estudiantes contenía imágenes de apoyo y recomendaciones para facilitar el proceso de aprendizaje.

Durante esta etapa, se realizaron simulacros para cada sesión, en los cuales participaban todos los miembros del equipo de las ACATEC, simulando ser tutorados y planteando preguntas a los tutores. Estos simulacros permitieron identificar tiempos promedio para cada actividad o sección de la guía, evaluar la complejidad de los contenidos y recibir sugerencias sobre cómo abordarlos de manera más efectiva.

3.1.5. Revisión de documento según apoyo académico [13]

Durante esta etapa, se llevó a cabo una revisión detallada de la primera versión de las sesiones con la asesoría de un profesional en educación. El apoyo pedagógico evaluó varios aspectos clave, incluyendo la estructura de cada sesión, los objetivos tanto individuales como del curso en general, el nivel de los contenidos en relación al público objetivo y el enfoque del curso. Para evaluar estos aspectos, se abordaron las siguientes preguntas relacionadas a cada uno de ellos.

- **¿Es clara la estructura de la sesión?**

Se realizaron modificaciones a la estructura anteriormente establecida. Se incluyó una sección de *Inicio de sesión* en todas las sesiones, la cual tiene como objetivo repasar lo aprendido en la sesión anterior y facilitar la transición hacia los nuevos contenidos. Además, se agregó una sección de cierre al final de cada sesión, en la cual los tutores resumen lo abordado y realizan preguntas a los tutorados para obtener retroalimentación sobre lo aprendido, como qué contenidos les gustaron más o cuáles les resultaron más difíciles de entender. Estas preguntas ayudarán a identificar qué aspectos deben reforzarse en la próxima sesión.

Otra modificación realizada fue la diferenciación entre desafío y actividad. En el primer bloque de cada sesión, además de abordar parte de los contenidos, se incluye una actividad guiada no evaluada que ayudará a comprender los conceptos de manera más clara y brindará la oportunidad de aclarar dudas con los tutores. En el segundo bloque se introducen nuevos conceptos y se plantea un desafío evaluado. Sin embargo, esta evaluación no busca etiquetar a los tutorados según su rendimiento en comparación con el resto de la clase, sino que se utilizará como una herramienta discreta para mejorar la entrega de contenidos y brindar apoyo a aquellos tutorados que puedan necesitarlo.

En cuanto a las guías, se mantuvo la diferenciación entre la guía del tutor y la guía del estudiante. En la guía del tutor se agregaron consejos para abordar los contenidos de manera más entretenida y atractiva, mientras que en la guía del estudiante se incluyó material adicional como glosarios, imágenes o analogías para facilitar la comprensión de términos o conceptos abstractos. Aunque habrá ligeras diferencias entre ambas guías, seguirán la misma estructura, por lo que los nombres de las secciones serán coincidentes tanto para el estudiante como para el tutor.

- **¿Son alcanzables y coherentes los objetivos?**

Se llevó a cabo una definición escrita de los contenidos, estableciendo la profundidad deseada y su importancia en el curso. A partir de esta definición, se procedió a establecer los objetivos principales y secundarios. Los objetivos principales se enfocaron en responder a la pregunta: ¿Qué es lo fundamental que los estudiantes deben aprender en este curso? Estos objetivos se centraron en los contenidos que forman la base sobre la cual se construirá en las sesiones siguientes. Por lo tanto, el logro de estos objetivos principales es crucial para el progreso del curso.

Por otro lado, los objetivos secundarios se plantearon como respuesta a la pregunta:

¿Qué contenidos son complementarios a los objetivos principales? En otras palabras, si los tutorados logran adquirir estos conocimientos adicionales, se considera un valor añadido, aunque no son críticos para avanzar en el curso. Estos objetivos secundarios se reconocieron como importantes, pero no indispensables, y se orientaron a ampliar el conocimiento y las habilidades de los tutorados.

De esta manera, se estableció una estructura clara de objetivos principales y secundarios que orientaron el diseño y la enseñanza del curso, asegurando un enfoque centrado en los contenidos esenciales y proporcionando oportunidades para la adquisición de conocimientos adicionales.

- **¿Es adecuado el nivel para el usuario objetivo?**

El proceso de adaptación de las sesiones consistió en simplificar los conceptos complejos y utilizar ejemplos cotidianos para facilitar la comprensión de los tutorados. Se estandarizaron los contenidos principales para abarcar un rango de nivel educativo específico. Se establecieron objetivos claros para cada sesión, enfocándose en los conocimientos fundamentales. Se diferenciaron desafíos no evaluados y actividades guiadas para reforzar el aprendizaje. Las guías de tutor y estudiante siguieron una estructura similar, pero con enfoques específicos. Se realizaron simulacros y se recopiló retroalimentación para mejorar el diseño de las sesiones. En resumen, se adaptaron los contenidos al nivel de usuario para facilitar el aprendizaje.

- **¿El enfoque de enseñanza es apto para el objetivo general?**

En este enfoque de enseñanza, se decidió introducir los conceptos básicos de circuitos y programación de forma gradual, comenzando por actividades de observación y armado de circuitos simples. Posteriormente, se incorporaron conceptos más complejos de electrónica una vez que los tutorados se sintieran cómodos en el entorno de aprendizaje. Por último, se introdujeron conceptos abstractos de programación sin dificultad adicional. Este enfoque fue validado por el apoyo pedagógico y no requirió modificaciones.

3.1.6. Ejecución de documento

Durante la implementación de las primeras versiones de las guías en las sesiones en línea, se llevó a cabo un trabajo paralelo de mejora en colaboración con el apoyo pedagógico. La evaluación de estas implementaciones se basó principalmente en la retroalimentación y la percepción de los tutores, ya que no se disponía de un instrumento formal de evaluación en ese momento. La información recopilada durante esta etapa se almacenó con el propósito de utilizarla en futuras mejoras del material. Aunque no se realizaron ajustes inmediatos, se planificó aprovechar esta información en momentos posteriores para perfeccionar las guías.

3.1.7. Recolección de retroalimentación de la sesión

Como se hizo mención anteriormente, se solicitó a los tutores realizar una retroalimentación de cada sesión, abarcando aspectos como el flujo de contenidos, el material utilizado, los tiempos asignados a cada sección y otros puntos que pudieran brindar información relevante para mejorar el material. No obstante, es necesario tener en consideración que, en el futuro, al comparar dicha retroalimentación con la información recolectada a través de un instrumento de evaluación, este último posee una mayor solidez, permitiendo realizar modificaciones respaldadas en datos medibles y comprobados.

3.1.8. Documento pedagógicamente validado

Luego de someter cada guía al proceso mencionado anteriormente, que incluyó una revisión por parte del apoyo pedagógico y su posterior mejora basada en dicha revisión, se obtuvo un documento validado pedagógicamente. Este documento cuenta con una estructura definida, objetivos claros, un nivel adecuado y un enfoque específico. Además, se añadió un instrumento de evaluación para cada desafío presente en las sesiones. Sin embargo, es importante destacar que el uso de estos instrumentos es interno de las ACATEC y tiene como objetivo principal apoyar la toma de decisiones y llevar a cabo acciones que promuevan un aprendizaje más efectivo para cada tutorado. Como segundo objetivo, se busca obtener una visión general al finalizar cada año, evaluando la efectividad en el logro de los objetivos planificados, identificando fortalezas y debilidades en el aprendizaje y en la ejecución de las ACATEC.

3.1.9. Corrección tomando en cuenta realimentación de ejecución en línea

El material validado pedagógicamente se enriqueció con la realimentación de las sesiones realizadas en línea por los tutores. Las sugerencias recibidas se centraron principalmente en complementar con más información, mejorar la distribución de los tiempos y perfeccionar la entrega de contenidos. Al integrar estas sugerencias, el material se volvió más completo, sin necesidad de realizar modificaciones significativas en su estructura. Además, algunas de estas observaciones coincidieron con las correcciones realizadas por la pedagoga, lo que evitó la necesidad de hacer ajustes adicionales.

3.1.10. Equipo inicial desarrollador hace realimentación a guía e instrumento evaluativo

En la última etapa, se procedió a entregar el material diseñado al equipo encargado de la generación de material de las ACATEC. Este equipo, al haber estado involucrado desde el inicio del proceso de diseño, tiene un claro entendimiento de los objetivos iniciales con los que se trabajó en las guías y puede proporcionar comentarios críticos sobre el logro de la idea inicial. El material entregado para su evaluación incluyó la guía de tutores, la guía de estudiantes y el instrumento de medición correspondiente a cada sesión.

3.1.11. Últimas correcciones y entrega del material guía de cada sesión

Tras recibir la retroalimentación del equipo de generación de material de las ACATEC, se realizaron correcciones menores y se entregó el material final de cada sesión. Cada sesión cuenta con guías para tutores, guías para estudiantes e instrumentos de medición correspondientes. En total, hay 7 sesiones más una sesión inicial denominada Sesión N°0, que se abordará más adelante. El objetivo original del material se cumplió exitosamente.

4 | Material desarrollado

En este capítulo, se analiza el enfoque del material generado y se detallan los propósitos y objetivos considerados al diseñar las ocho sesiones.

4.1. Guías de trabajo

El conjunto de material diseñado consta de ocho guías y siete rúbricas de evaluación. Las guías se centran en enseñar contenidos de electrónica utilizando la plataforma de simulación 3D y circuitos Tinkercad, aunque el uso de la plataforma no es el foco principal. Cada guía se divide en dos bloques, con actividades no evaluadas en el primer bloque y desafíos evaluados en el segundo bloque. Cada desafío cuenta con una rúbrica diseñada específicamente para evaluar los contenidos y objetivos de la sesión. Además, en este capítulo se proporcionará una explicación más detallada sobre las dimensiones e indicadores de las rúbricas.

4.1.1. Contenidos generales

El dispositivo seleccionado para enseñar las disciplinas deseadas es Arduino UNO. Se planificó incluir contenidos generales, como una introducción a la programación en Arduino, conceptos básicos de electrónica y la construcción de circuitos simples utilizando al menos un sensor comúnmente utilizado. Estos contenidos generales se consideran suficientes para lograr el objetivo final establecido.

4.1.2. Tipos de guías

Para una mejor comprensión del enfoque del curso, se presentarán dos análisis que permitirán entender mejor el diseño de las ocho sesiones. El primer análisis se centrará en la estructura, que se divide en diferentes etapas en el proceso temporal de aprendizaje durante la ejecución de las sesiones. Estas etapas son:

1. Introducción al armado de circuitos y programación en bloques: esta etapa abarca las primeras cuatro sesiones y tiene como objetivo establecer una base sólida en el pensamiento computacional, dando sentido a la programación. Además, se presenta por primera vez un circuito, aunque no se profundiza demasiado en él, centrándose principalmente en su armado y su funcionamiento dependiente de la programación.

2. Introducción a programación en código, resolución de problemas de programación y armado de circuitos: en esta etapa, consta de dos sesiones clave que introducen a los tutorados en la programación en código y la resolución de problemas. Se busca brindar las herramientas necesarias para enfrentar desafíos de programación más complejos y ampliar el uso de componentes en el armado de circuitos. Esto sentará las bases para el aprendizaje posterior en programación y electrónica.
3. Programación en código, comprensión de conceptos de electrónica básica y, uso y conexión de sensores: tercera etapa, que consta de dos sesiones en las que se presentan por primera vez conceptos de electrónica básica, señales y cómo aplicar este conocimiento al uso de sensores.

El segundo análisis se realiza en torno a la función de cada sesión, que se divide en sesiones de reconocimiento, sesiones de preparación base y sesiones objetivo.

1. Reconocimiento: las sesiones N°0 y N°2 fueron pensadas con el objetivo de que el tutorado se familiarice con los contenidos y las herramientas a usar a lo largo del curso. Además, se busca que conozca el sistema de funcionamiento de las ACATEC y las reglas de comportamiento.
2. Preparación base: las sesiones N°3 y N°4 corresponden a sesiones diseñadas para que el tutorado descubra por sí mismo ideas de importancia, a través del desarrollo de actividades simples que permitirán una mejor comprensión de las disciplinas objetivo en el futuro.
3. Objetivo: las sesiones N°5, N°6 y N°7 abarcan los contenidos objetivos, donde se ejercitan de forma más completa las disciplinas deseadas. Cabe destacar que estas sesiones se apoyan plenamente en el aprendizaje alcanzado en las sesiones anteriores, por lo que, si bien las sesiones objetivo representan el propósito del curso, todas tienen importancia en el proceso.

4.1.3. Sesiones

Cada sesión se diseñó con objetivos claros y se tuvo en cuenta la retroalimentación de sesiones anteriores. Se consideraron actividades, mejoras en el contenido y respaldo pedagógico. A continuación, se mencionan las consideraciones surgidas al diseñar cada sesión.

1. Sesión N°0: tiene dos objetivos importantes. El primero es obtener un diagnóstico previo al taller para medir los avances individuales. Se realiza un test abarcando todos los contenidos. El segundo objetivo es establecer una relación sana y colaborativa entre los participantes, a través de una actividad lúdica que fomenta la confianza y el respeto. Esto es crucial para el trabajo en equipo y la expresión de ideas.
2. Sesión N°1: a lo largo del taller anual se utilizan tres herramientas virtuales para desarrollar las sesiones. Estas fueron seleccionadas con el objetivo de facilitar la interacción en el grupo y acercar la experiencia lo más posible a una sesión presencial de las ACATEC. Esta sesión presenta las plataformas y muestra el uso de sus funcionalidades a través de actividades que ayudan a desarrollar el pensamiento lógico-matemático. Las plataformas a utilizar son:

- Classroom: plataforma gratuita que permite tener un depósito de material educativo en línea y compartirlo con todos los tutorados simultáneamente como una especie de repositorio. Además, permite la creación de aulas y encuestas.
- Zoom: plataforma pagada de reunión que posee una versión de prueba gratuita, la cual cuenta con opciones que intentan simular las interacciones posibles en una sala de clases como: levantar la mano, mostrar presentaciones, rayar en pantalla y hacer encuestas.
- Tinkercad: plataforma utilizada para el diseño de figuras 3D, circuitos y programación anteriormente descritos.

El correcto uso de estas plataformas es algo primordial a la hora de ejecutar cualquiera de las sesiones, por lo que debe ser parte de los contenidos entregados. Estos deben ser presentados al inicio con el objetivo de asegurar que el tutorado aprenda a utilizar las plataformas de forma correcta.

3. Sesión N°2: el objetivo principal de esta sesión es el reconocimiento consciente de los procesos mentales que ocurren al resolver un problema, aunque sea básico. Para lograrlo, se utiliza la programación en bloques, que permite separar visualmente las acciones: cada bloque representa una acción y un conjunto de bloques puede descomponer un problema más grande. Además, esta estructura visual permite identificar patrones repetitivos que fomentan la formación de algoritmos mentales de manera consciente. Todo esto prepara al tutorado para abordar en el futuro la programación en código, como una herramienta adicional para resolver problemas.
4. Sesión N°3: esta sesión continúa reforzando los contenidos abordados en la sesión anterior, pero con mayor enfoque en el desarrollo del pensamiento computacional, mediante la definición de algoritmos que posteriormente se convierten en diagramas de flujo. Esto brinda una base sólida a la cual el tutorado puede recurrir cuando se enfrenta a un problema, permitiéndole visualizar de manera objetiva y ordenada una posible solución. Incluso esto le permite al tutorado descomponer cualquier problema en problemas más pequeños, facilitando su resolución. Además, se inicia la introducción a los circuitos a través de actividades en las que se reconocen visualmente los componentes y dispositivos básicos que se utilizarán a lo largo del taller, así como sus funciones fundamentales.
5. Sesión N°4: en esta sesión se comienza con la programación en código y el armado de circuitos. Se establece una conexión específica entre los bloques de código y el código en texto, profundizando en este último. Se utilizan funciones básicas de Arduino como `void loop()`, `void setup()`, `Serial print()`, `delay()`, `digitalWrite()`, `digitalRead()`, `pinMode()` y `Serial.begin()`. Se introduce el monitor serial, el cual es de gran importancia para programar con señales externas y facilitar el seguimiento de los valores de las variables involucradas, permitiendo detectar errores en el código de manera más sencilla. Se consideran las posibles dificultades que los tutorados pueden tener al comprender los términos utilizados, ya sea por su carácter abstracto, su dificultad para imaginarlos o porque son palabras nuevas para ellos. Por ello, el diseño de esta sesión busca simplificar al máximo los términos, añadiendo definiciones sencillas en los pies de página y traduciendo palabras del inglés. En cuanto al armado de circuitos, no se profundiza en la teoría, ya que el objetivo de esta sesión es familiarizar a los tutorados con el uso de Arduino y su

programación.

6. Sesión N°5: en esta sesión se aborda el concepto de condicionales en la programación, utilizando ejemplos de sucesos condicionales de la vida cotidiana. Dado que este concepto puede resultar complejo para estudiantes de educación básica o media, se dedica toda la sesión a la definición y realización de ejercicios que ayuden a comprender la utilidad real de las funciones condicionales en Arduino y en otros lenguajes de programación. Se emplea un ejemplo similar a un semáforo para evitar complicaciones innecesarias y permitir a los estudiantes concentrarse completamente en el objetivo de aprendizaje.
7. Sesión N°6: en esta sesión se abordan los conceptos técnicos y abstractos de energía, voltaje, corriente y resistencia. Luego de explicar estos conceptos básicos, se introducen conceptos menos abstractos pero igualmente complejos, como el cortocircuito y el circuito abierto. Todos estos conceptos se apoyan en términos matemáticos, como las variables y su dependencia o independencia. Por lo tanto, se introducen indirectamente durante la sesión. También se comienza a introducir contenido más avanzado de electrónica, como las señales y el reconocimiento de una señal analógica o digital. Por último, se muestra de manera gráfica algunos de los componentes y sensores más utilizados y sus conexiones con Arduino. No se profundiza mucho en este último aspecto, ya que la sesión es bastante densa en términos de contenido abstracto. Además, esta sesión no incluye un desafío específico, ya que el enfoque principal se centra en la comprensión de los conceptos mencionados anteriormente.
8. Sesión N°7: en esta sesión se repasan nuevamente las funciones de Arduino presentadas en la sesión N°4, pero ahora se incluyen las funciones relacionadas con la lectura y escritura de señales analógicas. Estas se estudian con el objetivo de que los tutorados puedan reconocer fácilmente cuándo utilizar una señal analógica o digital. Esto es posible gracias a la introducción del concepto de señal en la sesión anterior, de modo que los tutorados adquieran el conocimiento necesario para diferenciar entre la lectura y escritura de señales analógicas y digitales. Asimismo, se profundiza en el conocimiento de las funciones utilizadas anteriormente, comprendiendo su fundamento y propósito. Además, se introducen los conceptos de actuadores y sensores, relacionándolos con los componentes vistos en la sesión anterior.

4.1.4. Rúbricas de evaluación [14] [15]

En el primer diseño, existía la necesidad de evaluar los contenidos entregados en las sesiones, pero no se le había dado la urgencia ni la importancia que merecía. Aunque las ACATEC son una iniciativa desarrollada en las aulas, la mayoría de sus integrantes son estudiantes y no tienen conocimientos en pedagogía. Por lo tanto, no se consideró la importancia de la evaluación. Esto también se debía a la idea generalizada entre los integrantes de las ACATEC de que las sesiones debían ser instancias de aprendizaje entretenidas, sin presiones ni evaluaciones que pudieran afectar la motivación individual de los tutorados. Esta idea cambió con el apoyo de la experta en pedagogía y evaluación, quien hizo entender la importancia de medir el logro de objetivos en cada sesión.

Así, en compañía de la experta en pedagogía y evaluación, se diseñaron rúbricas de evaluación para cada desafío presente en las sesiones. Estas rúbricas se centran únicamente

en la evaluación del desafío de cada sesión, ya que fueron diseñados de manera que su resolución abarque todos los contenidos tratados, lo que se considera representativo del alcance de los objetivos. Estas rúbricas se crearon teniendo en cuenta la idea de no convertir las ACATEC en una clase común de una institución educativa, donde se realicen pruebas individuales que puedan desmotivar a los tutorados. Por lo tanto, se optó por evaluar discretamente a los tutorados, diseñando las rúbricas para que sean completadas por el tutor durante la sesión. Es importante destacar que cada tutorado estará plenamente consciente de que está siendo evaluado, pero los resultados solo se entregarán si los solicita.

Cada rúbrica cuenta con **dimensiones**, que se definen como las habilidades que se espera que el tutorado aprenda. Algunos ejemplos podrían ser el pensamiento lógico, la expresión oral y el conexionado de componentes electrónicos. Dentro de cada dimensión se encuentran los **indicadores**, que se pueden definir como lo que visualmente hace el tutorado y que demuestra que ha logrado aprender, total o parcialmente, la habilidad definida en la dimensión. Cada dimensión puede tener varios indicadores. Por ejemplo, en la dimensión de expresión oral, algunos indicadores podrían ser el uso de un lenguaje formal y culto al presentar su proyecto a sus compañeros, la utilización de un tono de voz audible y una modulación que permita entender las ideas presentadas, entre otros.

Cada indicador puede ser evaluado en una escala de 0 a 3, en donde:

- 0: No logrado
- 1: Medianamente logrado
- 2: Logrado
- 3: Destacado

Esta escala de evaluación se utilizó pensando en simplificar el proceso para los tutores encargados de llevarla a cabo. Se busca evitar que esto tome demasiado tiempo y haga que el tutor deje de atender los requerimientos de enseñanza de la sesión. Sin embargo, al igual que todo lo planteado en esta memoria, debe ser probada y modificada si es necesario, basándose en retroalimentación documentada.

5 | Conclusiones y recomendaciones

5.1. Conclusiones

En resumen, esta memoria ha logrado obtener una solución concreta al problema principal planteado. El proceso comenzó con la recopilación de información dentro de la iniciativa, involucrando a los tutores y futuros usuarios del material y la documentación, para validar el enfoque como un trabajo en equipo que considerara sus experiencias e ideas. Tras una investigación y evaluación, se seleccionó Tinkercad como herramienta de apoyo para el diseño de las sesiones, cumpliendo con las expectativas al ofrecer las funcionalidades necesarias para el diseño de circuitos y programación. El diseño de las sesiones consideró los contenidos definidos, su profundidad, la estrategia de enseñanza, la distribución temática y los objetivos específicos de cada sesión. Además, se realizaron mejoras en el material a partir de la retroalimentación obtenida durante el proceso. Una de las dificultades encontradas fue la distribución de contenidos, ya que era necesario mantenerse dentro del tiempo establecido y evitar dejar temas pendientes de una sesión a otra, por lo que se tuvo que calcular cuidadosamente el tiempo requerido para abordarlos en su totalidad.

5.2. Recomendaciones para trabajo futuro

Esta memoria forma parte de un proceso amplio y ambicioso que busca transformar las ACATEC en una iniciativa que no solo se base en una evaluación cualitativa de su impacto, como se hacía hasta el año 2020, sino también en una evaluación cuantitativa. El trabajo presentado aquí es solo un ejemplo de los procesos que las ACATEC deben llevar a cabo de ahora en adelante. Seguramente se desarrollarán nuevos materiales en el futuro, se establecerán nuevos objetivos y se ampliará la cantidad de contenidos abordados en el taller, pero el principio fundamental siempre será el mismo: trabajar en pos de la mejora de la iniciativa, basándose en la evaluación cuantitativa del impacto en las personas.

A continuación, se enumeran algunas recomendaciones para dar continuidad al trabajo presentado en esta memoria.

- Se recomienda renovar, rediseñar o ajustar las rúbricas de evaluación de tal forma que sean cómodas de abordar por los tutores, facilitando así su uso.
- Se recomienda realizar mejoras en las sesiones antes de ejecutarlas, siguiendo las retroalimentaciones del año anterior. Esto permitirá mejorar el proceso en comparación con

lo mostrado en esta memoria, donde se realizaron modificaciones a las guías durante la ejecución de las sesiones anteriores, lo cual generó una extensión del proceso.

- Si bien las herramientas investigadas en esta memoria fueron todas orientadas a un trabajo remoto, se espera poder mantener la investigación de nuevas herramientas relacionadas con la electrónica y la programación, ya sea de uso en línea o presencial, con el objetivo de aprovechar al máximo los recursos disponibles y orientarlos hacia la mejora del aprendizaje en las ACATEC. Por ejemplo, el uso de robots para la enseñanza de electrónica y programación en formato presencial.
- Una de las dificultades a las que se enfrentarán las ACATEC es la implementación de las evaluaciones. Requerirá constancia y dedicación para que funcione según lo deseado. Se recomienda realizar capacitaciones a los tutores sobre la importancia de la evaluación en el ciclo de aprendizaje, ya que esta tiene la misma relevancia que la ejecución de la sesión.
- Se recomienda agregar tiempos promedio a cada actividad, basándose en la experiencia de los establecimientos donde se ejecuta la sesión.

Bibliografía

- [1] Leonidas Mustakis, 2022. Fundación Mustakis. Enlace: <https://www.fundacionmustakis.org/>. Ingreso: 06/01/2023.
- [2] Universidad de Talca, 2019. Taller de Robótica Universidad de Talca. Enlace: <https://www.utalca.cl/noticias/talleres-de-robotica-educativa-para-jovenes-en-la-utalca/>. Ingreso: 06/01/2023.
- [3] MCI Electronics, 2019. MCI Capacitaciones. Enlace: <https://cursos.mcielectronics.cl/electronica-basica/>. Ingreso: 06/01/2023.
- [4] Kodland, 2023. Talleres Kodland. Enlace: <https://www.kodland.org/>. Ingreso: 06/01/2023.
- [5] Semillero Innovación. Talleres Semillero Innovación, 2023. Enlace: <https://semilleroinnovacion.com/curso-introduccion-automatizacion-y-robotica/>. Ingreso: 06/01/2023.
- [6] USM, 2019. Iniciativa Sysmic Robotics. Enlace: <https://vra.usm.cl/fondos-externos/fondo-de-desarrollo-institucional-linea-emprendimiento-estudiantil/proyecto-fdi-fsm-1903/>. Ingreso: 06/01/2023.
- [7] Innovación y Robótica Estudiantil USM, 2016. Facebook Innovación y Robótica Estudiantil USM. Enlace: https://www.facebook.com/innovacionyrobotica/?locale=es_LA. Ingreso: 06/01/2023.
- [8] Carla E. Förster. El poder de la evaluación en el aula: mejores decisiones para promover aprendizajes, 2018. Editado por Carla E. Förster, publicado por Ediciones UC.
- [9] Analog Devices, 2021. Página oficial de LTspice en donde se puede descargar de forma gratuita (en inglés). Enlace: <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>. Ingreso: 06/01/2023.
- [10] MuseMaze, 2012. Página oficial donde descargar Every Circuit (en inglés). Enlace: <https://everycircuit.com/>. Ingreso: 06/01/2023.
- [11] Autodesk, 2011. Página oficial de Tinkercad en español. Enlace: <https://www.tinkercad.com/>. Ingreso: 06/01/2023.
- [12] MuseMaze, 2012. Vista de prueba de Every Circuit. Enlace: <https://everycircuit.com/app>. Ingreso: 06/01/2023.

- [13] J. Gimeno Sacristán y A.I. Pérez Gómez, 2008. Comprender y transformar la enseñanza. Ediciones Morata, publicado por Morata.
- [14] N. Elola, N. Zanelli, A. Oliva y L. Toranzos, 2010. La evaluación educativa: fundamentos teóricos y orientaciones prácticas. Editado por Elena Luchetti, publicado por Aique Educación.
- [15] Antoni Sans Martín, 2008. La evaluación de los aprendizajes: construcción de instrumentos. Ediciones Octaedro, publicado por Octaedro.

A | Anexo

Este anexo contiene en orden numérico las sesiones desarrolladas para el uso de tutorados, desde la N°0 a la N°7.



Academias de Tecnología Online

Sesión N° 0

7 de julio de 2023

Sesión N°0: Bienvenidos y bienvenidas a la aventura ACATEC.

Objetivo principal: Evaluar/ diagnosticar conocimientos respecto a los contenidos a abordar durante este año en las academias.

Objetivo secundario: Generar un ambiente de confianza, permitiendo compartir y conocerse.

Bienvenida

¡Hola! ésta es nuestra primera sesión, esperamos la disfrutes y aprendas junto a nosotros de la magia de la Electrónica. Pero primero:

- Debes procurar quedar presente en la clase.
- No olvides escuchar atentamente instrucciones generales para una adecuada convivencia. Recuerda que siempre se debe mantener un clima de respeto en las sesiones, cualquier crítica y opinión debe ser hecha con respeto hacia tus compañeros y compañeras o tutores. Caer en conductas que entorpezcan el desarrollo de la clase de forma reiterada, será interpretada como una falta de interés de seguir participando, por lo que si estas faltas son recurrentes podrían causar la expulsión de quien no haya cumplido las normas de buena convivencia.

Diagnóstico

A continuación deberás desarrollar una pequeña prueba de diagnóstico que nos permitirá medir el nivel de conocimiento que posees respecto a los contenidos que serán abordados durante el año.

En base a los resultados del test podrás ser destinado a los cursos básicos o avanzados, con el objetivo que los contenidos a ver sean abordados de forma adecuada al o la estudiante y lograr un aprendizaje efectivo.

Para esta actividad cuentan con *30 minutos*.

Enlace al test de diagnóstico:

<https://forms.gle/NwGo6A1EamoQRqKU6>

Descanso de 15 minutos

Dinámica rompe hielo: ¡Sálvese quien pueda!

Instrucciones

Espera atento las instrucciones de tu tutor o tutora.

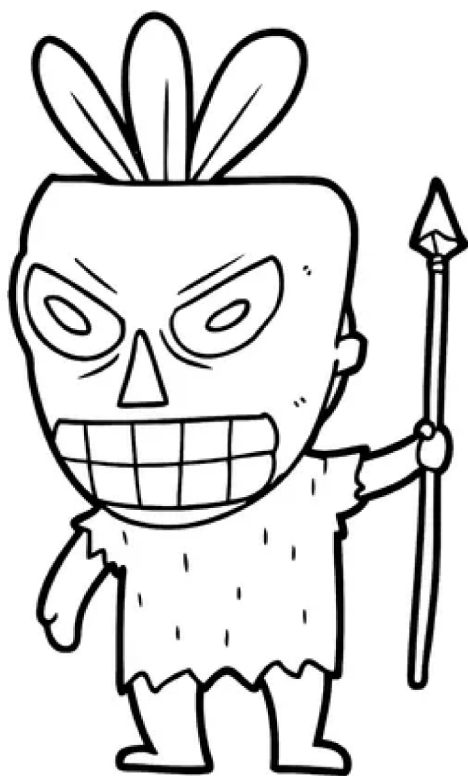
Caso: *Tribu de caníbales*

En un viaje camino a la gran competencia internacional ACATEC, familiares de los y las estudiantes sufren un accidente quedando varados en medio del Amazonas donde son capturados por una tribu caníbal (se alimentan de carne humana).

Con la carne del piloto, la tribu tiene para alimentarse una semana, por lo tanto, irán matando cada semana a una persona. El resto será cuidado para las siguientes semanas. Los ocho (u otro número) deben debatir y llegar a un acuerdo para establecer el orden en el que irán a la “olla” y además, tienen que buscar una forma para escapar con vida y llegar a la civilización.

Personajes:

- Una cantante de opera obesa que sabe medicina.
- Un chef delincuente joven y fuerte.
- Una profesora con dificultad para caminar que sabe técnicas de caza.
- Un anciano que conoce la zona.
- Un cerrajero que no puede ver.
- Un biólogo enfermo.
- Un encantador de serpientes sordo.
- Una interprete que conoce el idioma de la tribu.





Academias de Tecnología Online

Sesión N° 1

7 de julio de 2023

Sesión N°1: Herramientas de trabajo.

Antes de la sesión:

Para recibir y contar con toda la información importante para ésta y las siguientes sesiones, debes asegurarte de haber facilitado tu correo de contacto (a tu establecimiento o al correo acatec@usm.cl) con anticipación y haber aceptado la invitación vía correo para ingresar a Classroom.

Con el fin de optimizar el tiempo de la sesión, te pediremos que revises en internet una presentación de la plataforma de reuniones Zoom y sus funcionalidades más importantes. En esta sesión haremos uso de estas funcionalidades, por lo que es importante que tengas un conocimiento previo.

Objetivo principal: Conocer las herramientas a emplear durante el curso del año y el objetivo de las Academias de Tecnología.

Objetivo secundario: Fortalecer el pensamiento computacional mediante actividades lúdicas e interactivas.

Muy importante:

En las sesiones siempre se debe mantener un clima de respeto y tolerancia. Todos somos responsables de crear un ambiente acogedor, para sentirse en confianza de aprender y desarrollarse. Ser empático con el o la compañera cada vez que se lleven a cabo presentaciones de proyectos o actividades grupales, cualquier crítica u opinión debe ser hecha con respeto hacia sus compañeros y compañeras.

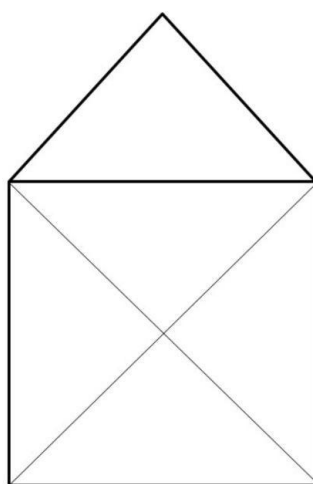
Iniciando

Zoom

Zoom como habrás investigado en internet es una plataforma para reuniones. Esta será nuestra sala de clases, y como probaremos más adelante podremos interactuar con varias funciones para hacer esta experiencia lo más parecida a como sería en una sala de clase. Zoom proporciona herramientas como el chat, audio, video, levantar la mano, escritura en una pizarra, compartir pantalla y también la separación de la reunión central en “grupos pequeños” con lo que trabajaremos bastante. Desde estos grupos pequeños se pueden llamar a los tutores y tutoras si es necesario. El cómo acceder a ellas, será mostrada por su tutor o tutora.

Actividad 1: ¡Buscando el camino!

Cuando se te indique, utilizando la herramienta de anotación, dibuja una casa como la que se presentará, sin levantar el lápiz y sin dibujar las aristas más de una vez. Si te equivocas recuerda que puedes borrar con la goma que está en la barra superior que aparece al seleccionar *Anotar* en tu pantalla.



Herramientas a utilizar

También se utilizarán otras plataformas para el desarrollo de las clases, como lo son Google Classroom, que es un aula virtual, donde nos comunicaremos fuera de clases y en caso de que se requiera, se les subirán distintos materiales y por otra parte se utilizará Tinkercad que es un espacio para realizar distintos proyectos, similar a un laboratorio virtual, en donde trabajaremos las habilidades que buscan desarrollar las Academias de Tecnología.

Classroom

Ingreso a la plataforma

Debes ser registrado a partir de tu correo electrónico, siendo habilitado por tu tutor o tutora y quedando listo para su participación en las clases.

Recorrido por la plataforma

Haz un recorrido por lo que será el sitio oficial de informaciones y actividades de las academias durante toda su duración, con la ayuda de tu tutor o tutora.

Probando el pensamiento lógico

Ingresa al apartado de *QUIZ 1* que se presenta en la pestaña de *Trabajo en clase* de Classroom, también puedes ingresar directamente desde el “tablón” en el cuadro con la

leyenda *Quiz número 1*. Debes responder este quiz con el fin de conocer las variadas formas posibles de usar esta plataforma. Además, podrás apreciar cuánto está desarrollado en ti el pensamiento lógico.

El pensamiento lógico se puede ejercitar, por lo tanto, el resultado que obtengas ahora no refleja el máximo potencial que puedes llegar a alcanzar ¡ánimo!

Descanso de 15 minutos

Tinkercad

Presentación de la plataforma

Te presentamos la plataforma de trabajo a utilizar a lo largo del año. Aquí podrás crear modelos 3D, circuitos (nuestro interés) y bloques de código (pronto a trabajar).

Equipos de trabajo

Se te agrupará con un compañero o compañera de trabajo al azar ¡oh no!, pero no te preocupes, todo tiene un propósito y el fin de esto es aprender a relacionarse y conocer a distintos tipos de personas, compartir experiencias y propiciar el trabajo en equipo, porque sí, somos un equipo. Se les asignará un alias de grupo, trata de recordarlo.

Registro en la plataforma

Se les solicitará ingresar al enlace enviado por el tutor o tutora e introducir como alias el nombre de tu grupo, sin espacio ni mayúsculas, es decir, si eres *Grupo 1* ingrésalo como *grupo1*. Los grupos son semanales y elegidos al azar en cada clase, sin embargo, habrá algunas actividades que deberán realizarse individualmente, debido a limitaciones de la plataforma. **Al finalizar la sesión puedes personalizar tu perfil a gusto.**

Desafío

Interfaz 3D

Te presentamos la herramienta de diseño de objetos tridimensionales en Tinkercad, aquí aprenderás:

- Dominio de espacio tridimensional.
- Escalamiento y desplazamiento de objetos.
- Configuración de objetos.

- Agrupación y creación de nuevos objetos.

No olvides consultar tus dudas las veces necesarias si es que algo no te queda claro o si necesitas un ejemplo para comprender de mejor forma, hazlo con confianza.

Proyecto: *Casa*

Junto a tu grupo, deben diseñar una casa con las nuevas funcionalidades aprendidas. Serás enviado junto a tu compañero o compañera a una sala de grupos pequeños de Zoom para hacer más clara la comunicación al trabajar y además que conozcas a tu par. Al finalizar el proyecto deben presentarlo, por lo que es necesario que a medida que trabajen analicen qué, cómo y por qué lo hicieron de esa forma, les será de ayuda al momento de presentar.

Presentación del proyecto

En los últimos minutos de la sesión deberás presentar junto a tu compañero o compañera el proyecto de casa y comentar cómo la construiste, por qué la construiste así y qué fue lo que más te costó.

Cierre

Se hará un recuento de las actividades que se ejecutaron. Además, todos juntos deberán responder algunas preguntas como:

- ¿Qué fue lo que más te gustó de la sesión?
- ¿Qué fue lo que más te costó entender?
- ¿Qué aprendiste?

No te preocupes si no alcanzaste a mostrar tu proyecto, lo podrás mostrar la siguiente sesión. Además, no olvides anotar el horario de ésta.



Academias de Tecnología Online

Sesión N° 2

7 de julio de 2023

Sesión N°2: Introducción a la programación.

Antes de la sesión:

Para esta sesión tendrás que haber desarrollado la tarea solicitada en Classroom. Dibujar el boceto en papel de una casa y de un avatar personal utilizando figuras geométricas básicas (esferas, cubos, cilindros, estrellas, etcétera). Como prueba de su desarrollo a tiempo, deberán subir una fotografía de su boceto a la plataforma Classroom.

Objetivo principal: Realizar un acercamiento a la programación por medio del uso de bloques de código.

Objetivo secundario: Conocer nuevas funcionalidades de la plataforma Tinkercad manipulando objetos 3D.

Muy importante:

Recuerda mantener el clima de respeto y las reglas de buena convivencia.

Iniciando

Revisión de tarea

Debes preocuparte de quedar presente en la sesión.

En la sesión de hoy se espera comprendan que los computadores suelen procesar una instrucción a la vez y que ocurren de manera secuencial. Para ello, se utilizará una herramienta visual e interactiva que corresponde a la programación en bloque.

Las actividades se realizarán de manera individual, por ello las dudas sobre estas serán libres de realizarlas frente a todos y todas o bien solicitar a algún tutor o tutora que habilite una sala especial de consultas en plataforma Zoom. De todas formas, durante la clase sus tutores estarán supervisando el avance y aclarando las dudas que surjan.

Presentación bloques de código

Todo computador requiere de instrucciones para funcionar, una de las formas más sencillas de programar es mediante bloques de código. En esta sesión debes utilizar los bloques para volver realidad la casa diseñada la semana pasada. Este tipo de programación consiste en utilizar bloques que están asociados a una instrucción, donde también es posible enlazar un bloque dentro de otro para formar grupos de instrucciones, los que pueden ser tratados como si fuesen una sola sentencia.

Programación en bloques I

Ingreso a la plataforma

Ingresa a la plataforma Tinkercad usando el alias otorgado. Luego en el panel de la izquierda ingresa a la sección *Bloques de código* y selecciona *Crear nuevo Bloque de código*. Ya que las actividades se realizarán de manera individual, una vez creado el proyecto, debes asignarle un nuevo nombre (tu nombre es una opción).

Figuras y formas

Debes arrastrar el bloque *Crear nuevo objeto* (ubicado en la sección *Modificar*) hacia el plano de trabajo. El nombre del objeto puede ser modificado seleccionando *Renombrar variable*. A continuación, puedes elegir alguna figura disponible en la sección *Formas* y arrastrarla hacia el código de tal manera que este nuevo bloque quede ensamblado al anterior.

El color y las dimensiones de la figura pueden ser modificadas en el mismo bloque. Para recortar una figura, puedes añadir otra figura al interior de ésta (con la forma que se desea eliminar), marcar la opción *Hueco* y añadir un bloque púrpura llamado *Crear grupo*.

Simulación

Para observar el progreso en el plano 3D, debes presionar el botón presente en la barra superior derecha con forma de **play**. También se puede ajustar la velocidad de simulación y avanzar paso a paso.

Movimientos

Las figuras y objetos se pueden desplazar en todas las direcciones, girar en todos los sentidos y escalar para volverlos más grandes o pequeños. Lo mejor para que puedas comprender estos bloques es ponerlos a prueba en la misma plataforma y observar sus movimientos.

Actividad N°1: ¡Nos mudamos de casa!

Al igual que la sesión pasada, debes diseñar una casa en la plataforma Tinkercad, aunque ahora con la dificultad agregada de utilizar solamente programación en bloque.

Actividad N°2: *El nacimiento de un avatar*

Ya que dominas la programación en bloque (si no es así y aún tienes dudas, es el momento ideal para aclararlas con tus tutores). Debes crear tu propio Avatar y presentarlo a la clase.

Descanso de 15 minutos

Programación en bloques II

Bloques matemáticos

Usando el bloque *Crear variable* puedes simplificar el cambio de algún valor que se utiliza varias veces en el código.

Luego, con el bloque *Definir* o *Cambiar* puedes asignar un nuevo valor a la variable creada.

Para realizar operaciones matemáticas se presenta el bloque de suma, este puede ajustarse para hacer restas, multiplicaciones o divisiones entre dos números (para operaciones entre más números se debe incluir un bloque dentro de otro).

Bloques de control

Los bloques anidados permiten repetir un grupo de sentencias, como el bloque *Repetir x Tiempos* y el bloque *Contar con j desde 1 a 10*, que realizan acciones similares según su configuración.

Te proponemos desplazar una figura en repetidas ocasiones (el bloque *Añadir* está antes del bloque *Anidado* y el bloque desplazar esta anidado). Así se podrá generar múltiples figuras (ocurre cuando el bloque añadir o copiar está anidado).

Desafío

La Aldea Konoha/Kage Bunshin no Jutsu

Con ayuda de los bloques *Crear nuevo objeto* y *Añadir copia de objeto*, debes crear una pequeña aldea basada en la casa que creaste anteriormente. Además, deben programar un recorrido que tu avatar pueda realizar de forma cíclica alrededor de la aldea.

Revisión

Finalizando el desafío, en la sala principal de Zoom, se compartirán los diseños y simulaciones de todos los proyectos, donde deberás explicar lo que buscaban realizar y la lógica que utilizaste para resolver los desafíos de esta sesión.

Cierre

No olvides comentarle a tu tutor o tutora aquellas actividades que más te gustaron y aquellas que te costó seguir o ejecutar, para que puedan ser reforzadas en las sesiones siguientes.



Academias de Tecnología Online

Sesión N° 3

7 de julio de 2023

Sesión N°3: Programación de Arduino mediante Bloques de Código.

Objetivo principal: Implementar la programación de Arduino haciendo uso de bloques de Código.

Objetivo secundario: Comprender la importancia y la utilidad de los diagramas de flujo al generar un algoritmo.

Muy importante:

Recuerda mantener el clima de respeto y las reglas de buena convivencia.

Iniciando

Siempre asegúrate de quedar presente en el control de asistencia.

Concepto de código

En la clase anterior mediante bloques de código de Tinkercad, se logró agrupar instrucciones (bloques) para realizar algo más complejo como una villa.

A diario realizamos muchas tareas en forma secuencial de manera inconsciente, pero para los computadores aquellas tareas no resultan tan fáciles.

Para que un computador realice una acción se debe conocer cuáles son los pasos y la secuencia u orden en que se realizan, por lo que se debe entregar información e instrucciones claras y precisas para que realice la acción que se le especifica.

Más adelante aprenderás cómo utilizar una herramienta que sirve como intermediaria entre el ser pensante (tú) y el ser inerte que llevará a cabo las tareas específicas que le asignes (un microcontrolador), con la que podrás escribir tus propios programas y cargarlos dentro de tu microcontrolador para que este lo ejecute.

Actividad N°1: mi avatar tiene sed.

Debes explicarle a un aldeano virtual de la Villa ACATEC como debe servirse un vaso de agua; para ello debes en papel o en un lugar donde puedas compartirla, elaborar una lista de al menos cinco pasos, siguiendo un orden cronológico, enumerando qué debe hacer (podrías comenzar con la instrucción *Ir a la cocina*). Propone modificaciones a los diagramas presentados por el tutor o tutora a través de la pantalla de Zoom para que queden más completos.

Algoritmos

¿Qué son y para qué sirven?

Felicidades, acabas de crear un Algoritmo, esto significa que se han creado un conjunto de instrucciones definidas y no-ambiguas, ordenadas y finitas que permite realizar una tarea.

En general, los Algoritmos sirven para solucionar problemas, realizar un cómputo, procesar datos y llevar a cabo distintas actividades. La idea de esto es que se escriba una solución a un problema o se encuentre una salida a partir de un estado inicial y una entrada, determinando los pasos mínimos que debe realizar un sistema para lograr esta.

¿Te has preguntado alguna vez cómo resolver un cubo Rubik? Es más simple de lo que imaginas. Sí, es un algoritmo, solo debes seguirlo y tendrás tu cubo Rubik armado.

Diagramas de flujo

El Diagrama de Flujo es la representación gráfica de un Algoritmo, este debe presentar el algoritmo en forma simplificada

Para construir uno debes marcar el inicio (Quiero un vaso de agua) y el final (Tomar agua), en el medio se deben incluir las acciones intermedias (Ir a la cocina, Buscar un vaso) para lograr cumplir la tarea final.

Debes plantearte preguntas para definir cuál será siguiente paso (¿El vaso está limpio?), y según la respuesta se pueden tomar dos caminos (Sí: Abrir llave y llenar el vaso con agua. No: Lavar el vaso).

Para evitar confusiones debes incluir flechas que indiquen cual es el siguiente paso (el camino o dirección en el que se ejecutan las instrucciones).

A continuación, se presentan dos ejemplos de Diagramas de Flujo. Se propone buscar una forma de mejorarlos, ¿falta algo o hay un error en ellos?

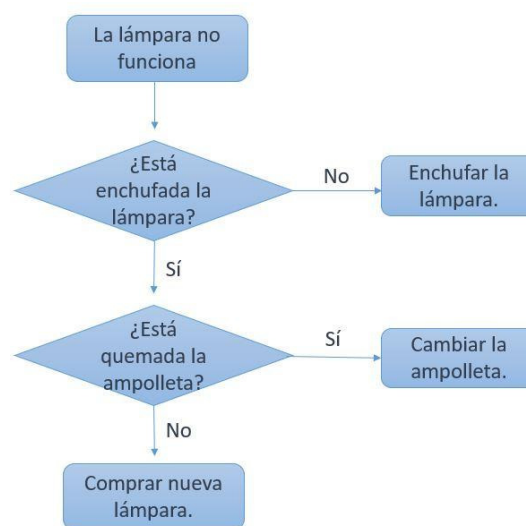


Figura A.1: Mi lámpara no enciende.



Figura A.2: Quiero una taza de café.

Actividad N°2: mi primer diagrama de flujo

Recordando la casa que realizaste la clase pasada, dibuja un diagrama de flujo lo suficientemente detallado para que cualquier persona entienda lo que debe hacer para replicarla. Recuerda hacer un diagrama ordenado, preciso y lo más acotado posible. ¡Adelante!

Circuitos I

Creación de un circuito en Tinkercad

Ingresa a Tinkercad usando el alias otorgado por tu guía en la sesión 1. Luego en el panel de la izquierda ingresa a la sección *Circuitos* y selecciona *Crear nuevo circuito*. Una vez creado el proyecto, debes asignar un nombre al proyecto ya que se trabajará de forma individual.

Circuitos en Tinkercad

El espacio de trabajo cuenta con una hoja en blanco donde puedes armar circuitos de forma personalizada. Las componentes electrónicas están disponibles en la barra de la derecha y pueden ser arrastradas hacia el espacio de trabajo.

Luego para simular tus circuitos debes presionar en *Iniciar simulación* y para detener o realizar modificaciones presiona en *Detener simulación*. Más adelante te invitamos a descubrir las componentes disponibles, conectarlas, dibujar cables y modificar sus formas o colores. También puedes utilizar los circuitos básicos que ya están armados y ver qué hacen.

Encendiendo un LED

En la actualidad no hay casi quién no conozca las luces LED, en la electrónica son una herramienta muy útil para corroborar o monitorear el estado de ciertas señales o acciones que se desean ejecutar. Además, uno de los circuitos más básicos para armar y programar corresponde al de una batería y un LED/ampolleta.

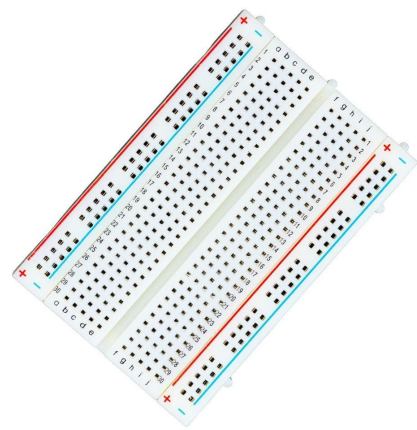
Juntos encenderemos un LED y responderemos preguntas interesantes sobre su funcionamiento. Ahora debes armar un circuito sencillo que corresponde al de una batería y un LED guiado por tu tutor o tutora.

Debes conectarlos, simular y responder las siguientes preguntas en conjunto con la clase. Hacer preguntas a tu guía para aclarar dudas. Preguntas a responder:

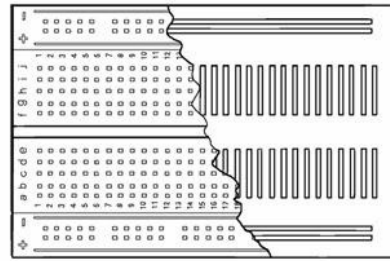
1. ¿Se enciende tu LED?
2. ¿Aparece una advertencia?
3. ¿Qué pasa si intercambias los cables?
4. Probar conectar una resistencia entre la batería y el LED ¿qué cambia?

Protoboard: ¿Qué es y para qué sirve?

Para armar circuitos muchas veces se parte por hacer un prototipo (primer modelo) de lo que buscamos como resultado final; una herramienta muy útil para el prototipado es la protoboard, la cual permite armar, desarmar o modificar a libertad el circuito en el que se trabaja. La protoboard consiste en un tablero con orificios que se encuentran conectados eléctricamente entre sí, estas conexiones se encuentran en forma de líneas en el tablero.



(a) Protoboard.



(b) Conexión interna protoboard.

Arduino

Nos concentraremos en el microcontrolador (circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria) Arduino, que se conforma por una CPU (unidad de procesamiento central), memoria y pines (patitas o puntillas) de entrada/salida, los que permiten interactuar con otras componentes electrónicas, como distintos tipos de sensores, así como actuadores (componente que emplea la energía recibida para activar un proceso).



Figura A.4: Arduino UNO.

Descanso de 15 minutos

Circuitos II

Códigos de Bloques para Arduino

A continuación, estudiaremos los distintos bloques disponibles en Tinkercad para lograr programar el microcontrolador Arduino. La plataforma dispone de bloques de salida, entrada, control, matemáticas, notación y variables. Analizaremos además los bloques de salida y control que nos permitirán programar un semáforo.

Al igual que en las clases pasados existen distintos tipos de bloques con diferentes colores, para el caso de los circuitos con Arduino son los siguientes:

- Salida en Azul.
- Control en Naranja.
- Entrada en Morado.
- Matemáticas en Verde.
- Notación en Gris.
- Variables en Rosado.

Bloques de salida

Corresponden a los bloques que le dan órdenes a todos los sensores y/o actuadores conectados como salida en el Arduino, a partir de estos se pueden manipular los pines digitales según sea necesario ya sea para LED's, servomotores, altavoces y monitor serie.

Bloques de control

Mediante los bloques de control es posible definir el tiempo que dura cada instrucción, la cantidad de veces que se ejecuta, en qué condiciones ocurrirá determinada acción, entre otras cosas. Corresponde al análogo de las condicionales if, while y for en el lenguaje de Arduino.

Cabe destacar que la relación entre ambos bloques es directa, mientras uno indica qué hacer al microcontrolador Arduino, el otro se preocupa de cómo hacerlo. Ej: En el encendido de un led el bloque de salida indicará si el led debe encenderse o apagarse y el bloque de control permitirá ajustar cuanto tiempo este permanecerá encendido.

NOTA: Para las siguientes actividades se utilizará el bloque de salida *Definir pasador x en ALTA/BAJA* que permite cambiar el voltaje a 5[v](alto) o a 0[v] (bajo) del pin digital **x** del Arduino. Además, es importante conocer el bloque de Control *Esperar x Segundos* que pausa la simulación durante **x** tiempo, lo cual será útil en la actividad siguiente para controlar el tiempo de encendido de un LED.

Actividad N°3: semáforo

Debes simular un semáforo utilizando LED's y un Arduino, donde se deben generar las transiciones entre el encendido de los LED's verde, amarillo y rojo. Para ello se recomienda dibujar un Diagrama de Flujos que ayudará a programar el encendido del semáforo usando Códigos de Bloques.

Luego de forma individual, crear el circuito y simular el semáforo.

Enlace del circuito a simular:

<https://www.tinkercad.com/things/231mt15c9Lc>

Lo que aprendiste en esta Actividad te ayudará mucho para el siguiente Desafío, por lo que cualquier duda con la que hayas quedado, es importantísimo que la aclares con tus guías o tus pares, antes de pasar al siguiente punto.

Desafío: esquina de semáforos

Ya que se construyó un primer semáforo, ahora te proponemos programar los semáforos de una intersección considerando los semáforos peatonales también.

¿Cuántos semáforos programar? Al igual que en la actividad anterior se recomienda dibujar el Diagrama de Flujo del problema antes de comenzar a construir.

Enlace de ejemplo de circuito:

<https://www.tinkercad.com/things/aELk4keT80m>

La metodología será la siguiente:

La clase será dividida en parejas, en donde ambos integrantes trabajarán de forma individual en un proyecto creado en Tinkercad. ¿Cuál es la idea? Que se apoyen y que las dudas que surjan las resuelvan primero entre ustedes; si ambos no saben cómo solucionar el problema, se puede pedir ayuda al tutor o tutora.

Al finalizar debes presentar a la clase el circuito de tu compañero o compañera de trabajo, explicando las diferencias y semejanzas entre ambos. Por lo que debes contemplar dentro de los plazos dados por tus guías, tiempo para interiorizarse mutuamente sobre el trabajo de tu equipo.

Cierre

Coméntale a tus guías aquellas actividades que más te gustaron y aquellas que te costó seguir o ejecutar, para que puedan ser reforzadas en las sesiones siguientes.



Academias de Tecnología Online

Sesión N° 4

7 de julio de 2023

Sesión N°4: Programación de Arduino mediante código.

Objetivo principal: Aprender a describir estructuras fundamentales de algoritmos utilizando lenguaje de programación.

Objetivo secundario: Desarrollar el pensamiento computacional y ser capaces de implementar diferentes ideas en un código.

Muy importante:

Recuerda mantener el clima de respeto y las reglas de buena convivencia.

Iniciando

Siempre asegúrate de quedar presente en el control de asistencia.

Introduciendo un lenguaje de código

Trabajar con bloques de códigos está bien para comenzar el camino hacia la programación, sin embargo, si queremos entregar instrucciones más complejas a nuestro microcontrolador, los bloques no serán suficientes. Cada uno de los bloques que hemos utilizado, de forma implícita son equivalentes a una o más líneas escritas de instrucciones para nuestro controlador.

En esta clase se les enseñará a programar mediante líneas escritas, en las cuales podremos identificar instrucciones de igual forma que en la programación de bloques.

Así como en el mundo existen muchos idiomas: español, inglés, italiano, etc. que utilizan los mismos símbolos para escribirlos, sin ser todos iguales, cada microcontrolador tiene un “idioma” en el cual debe ser programado para que este entienda nuestras instrucciones, a este le llamaremos lenguaje de programación. Es por lo anterior que si queremos que Arduino nos entienda las instrucciones que le damos, debemos dárselas en su lenguaje de programación, que llamaremos Lenguaje Arduino.

El siguiente cuadro muestra tres ejemplos de cómo los bloques son equivalentes a líneas de código:

Observación: Si bien de ahora en adelante aprenderemos un lenguaje específico, eso no quiere decir que si tienen la oportunidad de ver un código escrito en otro lenguaje, no serán capaces de entender nada. De forma conveniente existen muchos lenguajes de programación similares en estructura, incluso se suele decir que si se aprende a programar en un lenguaje, los siguientes lenguajes que se aprendan, será más rápido el proceso de aprendizaje.

Bloque	Código
	<pre>void setup() {</pre>
	<pre>} void loop() {</pre>
	<pre>} digitalWrite(3,HIGH);</pre>

Arquitectura de Arduino

void loop y void setup:*¿Para qué sirven y cómo utilizar estas funciones?*

Al ejecutar la aplicación de Arduino (previo descarga desde la página) se abre un nuevo sketch ¹ en el cual están escritas las dos funciones esenciales de Arduino.

Por otro lado, en Tinkercad, se debe ir a la pestaña de *Código* y seleccionar la alternativa de *Texto* para obtener el mismo resultado que en el párrafo anterior. En la ventana desplegada, ya sea en la aplicación Arduino o en la plataforma que estamos utilizando Tinkercad, aparecen dos funciones escritas:

- `void setup ()`
- `void loop()`

En el `void setup()` se escriben las instrucciones que queremos que se realicen una sola vez al encender el Arduino, sirve para inicializar variables², definir entradas y salidas en la comunicación entre el Arduino y el computador.

El `void loop()` como lo dice su nombre, es un bucle³ de instrucciones, por ello en esta sección del código se escriben todas las instrucciones que queremos se repitan **una y otra vez** hasta apagar el dispositivo.

Tener en consideración el uso de `//` para realizar comentarios en los códigos. Todo lo escrito siguiente a los símbolos `//` en una línea, Arduino lo ignorará como instrucción, solo servirá y podrá ser leído por el programador o programadora. Existe igualmente una forma de escribir comentarios más extensos o comentar múltiples líneas, dicha función puede ayudar en casos que voluntariamente necesitemos hacer que no se ejecuten ciertas líneas de código: **`/*código que se quiere omitir*/`**.

Ejemplos de ambos casos se muestran en la imagen a continuación:

¹Ventana donde se podrá escribir el borrador de un código

²Espacio designado en la memoria que almacenará lo que deseemos bajo un nombre, este valor podrá cambiar.

³Proceso que se repite infinitamente

```

void loop()
{
  ROJO = digitalRead(8);
  AMARILLO = digitalRead(9);
  VERDE = digitalRead(10);

  /*digitalWrite(2,ROJO);
  digitalWrite(3,AMARILLO);
  digitalWrite(4,VERDE);*/

  Serial.print(" ROJO: ");
  Serial.print(ROJO);
  Serial.print(" AMARILLO: ");
  Serial.print(AMARILLO);
  Serial.print(" VERDE: ");
  Serial.println(VERDE);

  delay(1000); //Se espera 1 segundo
}

```

La idea es generar un código ordenado con la finalidad de facilitar la corrección y detección de posibles fallas y que sus pares lo entienda también.

Traducción de palabras en inglés que podrían ser de ayuda:

Palabra	Traducción
Void	Vacío o Vacía
Loop	Bucle
Setup	Configuración
Delay	Demora o retardo
Mode	Modo
Write	Escribir
High	Alto o arriba (posición)
Low	Bajo o abajo (posición)
Input	Entrada
Output	Salida

Otras funciones útiles

Luego de hablar de las dos funciones esenciales de Arduino, conocerás otras funciones que llamaremos funciones básicas, las cuales son:

- **pinMode(pin, mode):** Corresponde a la función que define si los pines⁴ se comportan como entrada o salida. En la casilla pin ingresamos el número del pin digital⁵ o

⁴Clavija, patita o punta metálica de un conector.

⁵Pin con valores binarios, puede estar solo en dos estados: alto o bajo, 0[V] o 5[V]

análogo⁶. En *mode* ingresamos **INPUT** o **OUTPUT** según sea el caso.

- **digitalRead(pin)**: Corresponde a la función que lee el valor de un pin digital específico, ya sea HIGH o LOW, lo que entenderemos como ENCENDIDO o APAGADO. Estos estados son “leídos” como un 1 o un 0.
- **digitalWrite(pin, estado)**: Mediante esta función accionaremos los pines según su modalidad. En caso de que el pin sea configurado como salida, al activarlo (HIGH) se establece un voltaje de 5V en el pin. En caso de ser configurado como entrada, estaremos activando (HIGH) o desactivando (LOW) las resistencias pull-up⁷ internas en el pin.
- **delay(valor)**: Corresponde a una función de tiempo, está programada en milisegundos, de esta manera al ingresar en *valor* el valor 1000 será igual a 1 segundo.

Cabe destacar que en Arduino (y en los lenguajes de programación en general) las funciones y nombres de ellas son **case sensitive**, es decir, se distingue el uso de mayúsculas y minúsculas. De esta manera, es correcto escribir la función `pinMode(pin,mode)`, no así `pinmode(pin,mode)`.

Actividad: simulación de semáforo haciendo uso de código

Por ahora se mantendrán en terreno conocido y se repetirá la actividad de la clase pasada.

Una de las tareas repetitivas más conocidas es la de un semáforo.

Deben programar y montar el circuito de un semáforo basándose en el diagrama realizado en la sesión pasada.

El código debe ser escrito en texto, iniciando las salidas del Arduino dentro de la función `Void Setup()` y el funcionamiento del semáforo dentro de la función `Void Loop()`.

Descanso de 15 minutos

Funciones y tipos de variables de interés

Comunicación Serial

Es de interés tener alguna herramienta que permita observar desde el computador la información que procesa Arduino en su interior. Para ello utilizaremos una pestaña que es capaz de mostrarnos qué es lo que está viendo Arduino en el pin deseado.

⁶Pin capaz de leer valores reales, con decimales, los valores que nos da la naturaleza, por ejemplo: la temperatura, la altura, etc

⁷Resistencias parte de un circuito interno de Arduino que permiten fijar el voltaje en el pin en alto, en este caso 5[V], independiente si se utiliza o no este pin

Para entender lo que está pasando entre el Arduino y el computador el monitor debe hablar por el mismo “medio”, es lo mismo que para dos personas comunicarse por celular, ambas deberán tener uno, o si queremos conversar persona a persona, se debe estar presente. Así es como se define este **medio** como **comunicación serial de 9600 baudios**, si configuramos el monitor serial de esta forma, seremos capaces de ver la información exacta que traspasa el Arduino al computador. Para ello, se puede establecer una “conversación” con Arduino a través del cable mientras éste trabaja. Debes saber que existen muchos tipos de comunicación, pero en ese caso se utilizará la comunicación serial.

Para utilizar el monitor serial primero es necesario activarlo, para ello se utilizará la sentencia:

```
Serial.begin (9600) ;
```

Responde Junto a tu tutor o tutora:

¿Dónde debe ir dicho comando (o línea)?

¿Dentro de Void setup(), Void loop() o afuera?

¿Por qué tiene un número específico?

¿Qué significa el numero 9600?

La información que se recibe (Arduino → PC) se puede observar en la parte inferior del código de Arduino en *Monitor serie*.

¿Qué se observa en el monitor serie?

Cada vez que se quiera visualizar información en el monitor serie se debe utilizar las funciones:

- `Serial.print (...);`
- `Serial.println(...);`

Ambas funciones enviarán variables o palabras desde Arduino hacia el monitor del computador.

¿Cuál es la diferencia entre ambas entonces?

Un buen hábito es utilizar `Serial.print()` para hacer un seguimiento a las variables dentro de un código y ver que todo está funcionando bien (o mal) o para visualizar mediciones de distinta índole.

Variables

Se define una variable como un símbolo (se suelen utilizar ya sea una letra o una palabra) que es utilizado para designar un valor que suele cambiar dentro de un conjunto específico de números (Ejemplo: conjunto de los números enteros). En este lenguaje de programación se pueden describir distintos tipos de variables, entre ellas están:

- **char:** Se utiliza para almacenar un caracter (Letras).
- **int:** Será uno de los más usados, se utiliza para variables que tendrán números enteros (esta variable podrá tomar el valor de cualquier número entero dependiendo de lo que programemos).

- **float**: Se utiliza para variables donde se trabajará con números decimales.
- **boolean**: Son variables binarias, que como tales solo pueden tomar dos valores: **TRUE** o **FALSE**, ó en su defecto **0** o **1** respectivamente.

En el caso de definir variables en la parte superior del código, es decir, antes de la función `void setup()` estaremos definiendo una variables global lo que quiere decir que sera visible en todas las funciones, en cambio, al definir variables dentro de las funciones `void setup()` o `void loop()` se definirán como locales, es decir, será visible solo dentro de cada función.

Desafío: mecanismo de emergencia

Se requiere, como objetivo principal, diseñar un mecanismo de emergencia manual para tu semáforo de la primera parte. Para ello se deberá crear un nuevo código que permita utilizar **Señales externas** (Interruptores) para encender los LED's manualmente.

Para ello dispones de las variables **VERDE**, **AMARILLO** y **ROJO** que deberán almacenar la información proveniente del “exterior” y asignarse como estado a cada LED.

Como objetivo adicional se propone implementar un sistema de seguimiento que permita observar el valor de las variables en el monitor serial.

El staff presentará un circuito previo con la finalidad de aclarar dudas respecto a la actividad. Se hará entrega del circuito previo y del circuito con el nuevo conexionado.

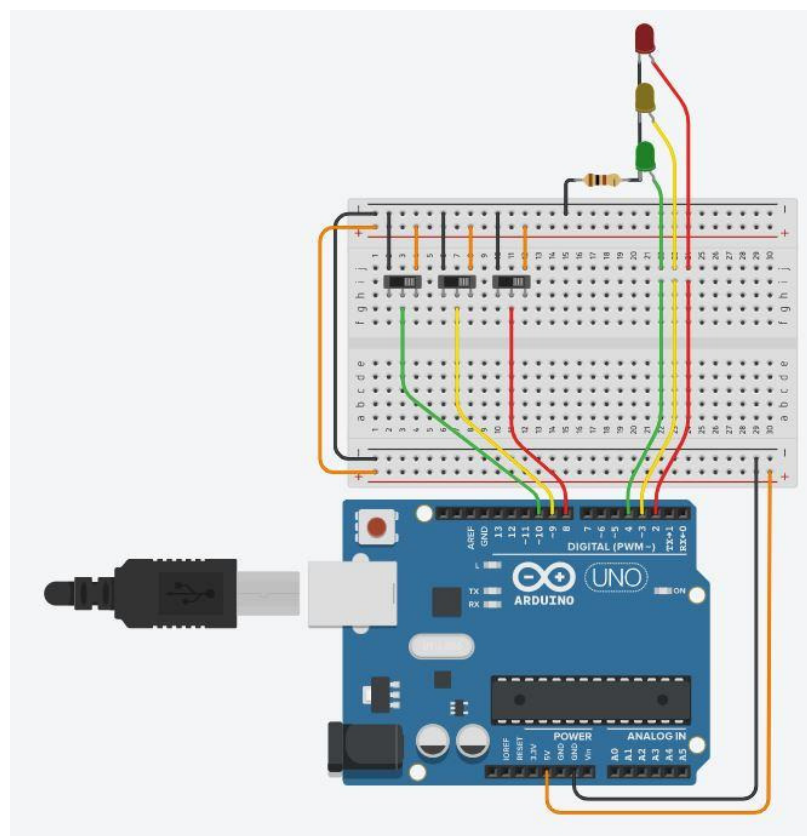


Figura A.5: Circuito Desafío Sesión N°4

- Pre Desafío: <https://www.tinkercad.com/things/7dRjntRvKtS>
- Desafío sin código: <https://www.tinkercad.com/things/0Ipjea3jh3T>

- Desafío con código: <https://www.tinkercad.com/things/1YTJmSzaNXz>

Cierre

Coméntale a tus guías aquellas actividades que más te gustaron y aquellas que te costó seguir o ejecutar, para que puedan ser reforzadas en las sesiones siguientes.



Academias de Tecnología Online

Sesión N° 5

7 de julio de 2023

Sesión N°5: Programación de Arduino mediante código.

Objetivo principal: Comprender la importancia y utilidad de las sentencias condicionales y bucles en la programación.

Objetivo secundario: Reconocer la aplicación de la programación en ejemplos de la vida cotidiana.

Muy importante:

Recuerda mantener el clima de respeto y las reglas de buena convivencia.

Iniciando

Siempre asegúrate de quedar presente en el control de asistencia.

Valores de verdad y comparadores

Podemos llamar valores de verdad a las posibles respuestas acerca de la existencia de un suceso, situación, algo observable de alguna forma. Un ejemplo sería: ¿Has desayunado? Si tengo conocimiento sobre esto, puedo responder si o no, no hay respuestas intermedias, o incluso si ésta fuera una afirmación, podría ser verdadera o falsa. Así cada suceso conocido por ti o que este dentro de nuestras posibilidades en el momento, podemos convertirlo en una afirmación o negación con un valor de verdad, este será **VERDADERO** o será **FALSO**.

Si nos apoyamos en lo anterior, podemos asegurar que los valores de verdad solo son dos, los cuales conoceremos como VERDADERO y FALSO, que a veces son representados por pares como (1,0), en donde 1 corresponde a verdadero y 0 a falso.

True(1)

False(0)

En programación estos valores de verdad se dan por lo general como resultado de una comparación de este tipo:

$x == y$ ¿es x igual a y?

$x \geq y$ ¿es x mayor o igual a y?

$x > y$ ¿es x mayor a y?

$x \leq y$ ¿es x menor o igual a y?

$x < y$ ¿es x menor a y?

$x \neq y$ ¿es x distinto de y?

Intenta definir valores de verdad para las siguientes sentencias, con la ayuda de tu tutor o tutora:

- $(1 == 1)$ ¿es el número 1 igual al número 1?
- $(4 \geq 1)$ ¿es el número 4 mayor o igual que 1?
- $(6 > 7)$ ¿es el número 6 mayor que 7?

Operadores Lógicos

En relación con lo anterior, a veces es necesario que se cumpla mas de una condición al mismo tiempo para que ocurra un suceso, o bien, de que 2 condiciones, puedan llevar al mismo resultado, y pueden ocurrir de forma separada. A esto se le llama una operación lógica, a continuación se muestra cómo se representan en Arduino:

`&&` Conjunción lógica, representa el “y”.

`||` Disyunción lógica, representa el “o”.

Lo anterior, significa, en el caso del primer comando, que entregará un valor de *true* (VERDADERO), si se cumplen todas las condiciones al mismo tiempo en la sentencia. Esto se puede leer como: Se cumple *condición 1 Y condición 2*.

Análogamente, en la segunda línea, que entregará un valor de *true* (VERDADERO), si se cumple alguna de las condiciones en la sentencia o bien las dos. Esto se puede leer como: Se cumple *condición 1 O condición 2*.

Un ejemplo de ambas operaciones lógicas sería:

- *Condición 1: Esta lloviendo*
Condición 2: Estas fuera de tu casa

Por lo que podríamos afirmar que necesitarás un paraguas si se cumple la condición 1 y también la 2. Deben suceder ambas condiciones para que se cumpla nuestra afirmación, cualquiera de ellas por si sola no podrá afirmar que necesitarás un paraguas.

- *Condición 1: Me hacen cosquillas*
Condición 2: Me cuentan un chiste

Por lo que podríamos afirmar que te reirás si se cumple la condición 1 o la 2. Cualquiera de las dos desencadenará nuestra afirmación.

Condicionales If y else

Aunque no lo hayas notado nuestras decisiones diarias dependen de distintos sucesos o situaciones, hagamos un pequeño recuento de nuestro día a día, es muy probable que comer en tu día dependa de si tienes o no tienes hambre y lo mismo el dormir, dependerá de si tienes o no tienes sueño. Los anteriores ejemplos son los más simples, pero son infinitos y tan complejos como quieras, como el avanzar de curso en el colegio depende de qué tan buenas fueron tus calificaciones y así, podrías pensar en infinitas dependencias.

Basándose en estas dependencias o condiciones aparecen las estructuras condicionales, las cuales nos entregan mayor versatilidad a la hora de programar una tarea que dependa de la ocurrencia de otra tarea o situación, en este caso la función *if* nos permite definir acciones a realizar en caso de que se cumpla la condición establecida, por ejemplo: si hoy esta soleado, saldré a caminar, en este caso es fácil determinar que la condición a analizar es **si hoy esta soleado**, si la condición es verdadera, la acción a realizar sería **saldré a caminar**, como se muestra en el siguiente código de ejemplo:

```
If (soleado){  
    saldre_a_caminar();  
}
```

Luego, es posible complementar la función condicional utilizando la expresión *else* que permite realizar una acción distinta en caso que la afirmación sea falsa, es decir, si hoy no esta soleado, me quedaré en casa. Es importante notar que en el caso del *else*, no es necesario expresar de manera explícita la condición, ya que, el procesador analizará por defecto la condición planteada en el *if*, como se muestra a continuación:

```
If (soleado){  
    saldre_a_caminar();  
}  
else{  
    quedarme_en_casa();  
}
```

Nota: Se debe tener claro que el uso de **else** es ideal cuando se quiere especificar la acción complementaria a la condición del **if**. En el caso en que solo se quiera especificar la acción que se realizará si se cumple la condición, solo necesitarás **if**.

Actividad N°1: ¡Alto ahí!

Últimamente los robos a una tienda de salchichas se han visto aumentados y resulta imposible reconocer a los salchi-ladrones debido a la falta de luz, es por ello que necesitamos tu ayuda profesional para programar un sistema de salchi-seguridad, el cual esta compuesto por un foco (LED) el cual se encenderá cada vez que alguien se acerque a la salchi-tienda, para ello un

técnico ha hecho el conexionado de un LED junto a un sensor de movimiento PIR (Sensor Infrarrojo Pasivo), solo falta tu parte del salchi-trabajo ¡Ánimo!

PIR es el nombre de un sensor comunmente utilizado en alarmas de movimiento. Usa un LED infrarrojo que mide la luz infrarroja (IR) radiada de los objetos situados en su campo de visión.

En el siguiente enlace podrás encontrar el circuito base para esta actividad:

<https://www.tinkercad.com/things/5LtmkQiUnXS>

Descanso de 15 minutos

Condicional Switch-case

Otro tipo de condicionales presentes en la programación de Arduino son los **switch-case**, estos, al igual que los condicionales **if/else**, permiten controlar una serie de tareas a ejecutar, dependiendo de una o varias condiciones que se deben cumplir. En este caso, el **switch-case**, compara el valor de una variable, con una serie de valores específicos, los cuales son declarados en la(s) sentencia(s) **case**, cuando el valor de la variable coincide con el caso a analizar, se ejecutará la acción declarada para ese caso.

Una consideración que se debe tomar en cuenta en la programación de los **switch-case** es el uso de la función **break**, que permite detener la ejecución de un código y que en este caso se utilizará para finalizar la ejecución de la tarea asociada a la sentencia **case**, y por otro lado, la sentencia **default**, que define una tarea a ejecutar, en caso de que la variable no cumpla ninguno de los casos planteados en las sentencias **case**.

Con estas consideraciones, se plantea el código general de programación de un **switch-case** como se observa a continuación, donde nuestra variable a analizar esta determinada por *var* y los casos posibles son *label1*, *label2* y **default**.

```
switch (var){
    case label1:
        //tarea2 a ejecutar
        break;

    case label2:
        //tarea 2 a ejecutar
        break;

    default:
        //tarea a ejecutar por defecto
        break;
}
```

Un ejemplo que podría ayudarte a entenderlo mejor podría ser: Si *var* fuera la hora del día y que cada *label*, sea la hora a la cual comes una comida del día: a las 9:00 desayuno, 13:00

almuerzo y 18:00 cena. Si var es igual a 9:00 entonces desayunamos, si var es igual a 13:00 almorzamos, si var es igual a 18:00 entonces cenamos y finalmente si no es ninguna de esas horas específicas (uso de `default`), no comemos (uso del `break`).

Desafío N°1: el menú

Como una forma de organizar los pedidos de comida que llegan a la cocina del restaurante ACATEC, un grupo de chefs del mismo propone diseñar un sistema de notificación de colores, según el platillo solicitado, para gestionar de mejor forma los platillos a preparar, es por esto que plantearon un circuito utilizando Arduino, en el cual se le asigna un color específico de LED a cada plato de comida preparado en el restaurante.

Por problemas de gestión, el grupo a cargo de la implementación del sistema solo logro diseñar el circuito pero no lograron generar la programación necesaria para implementar el sistema y solicitan tu ayuda para finalizarlo. Es por esto que deberás probar el sistema de notificaciones.

Utilizando lo aprendido en clases sobre `switch-case` y considerando el circuito descrito en el enlace al final de esta sección, debes programar las partes faltantes del código de forma que cambien las notificaciones de cada platillo cada 1 segundo. A medida que va corriendo el tiempo, se debe encender la luz correspondiente a cada platillo en cada segundo y apagar las anteriores. Además, se deberá ver en el monitor serial qué platillo esta siendo pedido. Para esto cuentas con el siguiente código base de ejemplo.

```
int rojo=13; //platillo A
int blanco=10; //platillo B
int azul=7; //platillo C
int naranja=4; //platillo D
int verde=2; //platillo E
```

```
void setup()
{
    Serial.begin(9600);

    pinMode(rojo, OUTPUT);

    //.....//
    //complete código aquí//
    //.....//
}
```

```
void loop()
{
    int i;
    for(i = 0; i<5; i++){
```

```

switch(i){
  case 0:
    digitalWrite(rojo, HIGH);
    digitalWrite(blanco, LOW);
    digitalWrite(azul, LOW);
    digitalWrite(naranja, LOW);
    digitalWrite(verde, LOW);
    break;
    //.....//
//complete código aquí//
//.....//
}

delay(1000);

}

}

```

Para ser más prácticos, se utilizará la función **for(i = 0; i < 5; i++)**, la cual se verá en las sesiones siguientes. Solo basta entender que lo que hace dicha función es: cada un segundo el valor de **i** va aumentando(desde 0 hasta 4, y se reinicia la cuenta después de llegar a 4, vuelve a 0), y bajo esta variable los platillos deben ir cambiando.

El siguiente enlace contiene el circuito y código base para este desafío:

<https://www.tinkercad.com/things/4pzSPsk21i0/>

Cierre

Coméntale a tus guías aquellas actividades que más te gustaron y aquellas que te costó seguir o ejecutar, para que puedan ser reforzadas en las sesiones siguientes.



Academias de Tecnología Online

Sesión N°6

7 de julio de 2023

Sesión N°6: Conceptos de electrónica básica y componentes de uso común.

Objetivo principal: Reconocer conceptos de electrónica básica, incluyendo señales análogas y digitales.

Objetivo secundario: Introducir componentes como medios de comunicación con el exterior.

Muy importante:

Recuerda mantener el clima de respeto y las reglas de buena convivencia.

Iniciando

Siempre asegúrate de quedar presente en el control de asistencia.

Voltaje y corriente

Introducción

Todo elemento electrónico requiere de energía para funcionar, específicamente necesita de la energía eléctrica por lo tanto es de nuestro interés analizar y comprender el comportamiento de la electricidad con el fin de manipularla de forma controlada.

La energía es la responsable de casi todos los fenómenos que observamos a nuestro alrededor, movimiento, luz, calor, etc. Podemos ver este proceso más claramente por ejemplo cuando comemos para poder vivir y movernos como queremos, si nosotros no comemos nuestra energía baja de forma considerable, o también si pensamos en la energía solar, tendremos más energía solar en un lugar donde hay más horas de sol. Existen distintos tipos de energía, algunos de ellos son:

- Energía Solar.
- Energía Atómica.
- Energía Mareo motriz.

¿Qué es la electricidad?

La forma más fácil para entender la electricidad es verla como el agua que corre por una manguera o cañería de tu casa. Se identifican 3 factores involucrados en su funcionamiento: la

fuerza que empuja el agua (Voltaje), las imperfecciones o lugares más angostos en la manguera que impiden el paso del agua (Resistencia) y finalmente se tiene el caudal (éste es definido como cantidad de fluido que pasa por la cañería en cierta cantidad de tiempo, podría medirse como $[m^3/s]$) de agua que logra salir de la manguera (Corriente).

Estas 3 variables son dependientes (Voltaje – Resistencia - Corriente) en sí y son las bases de la electrónica. **¿Qué quiere decir que son dependientes?**

Recuerda que las variables son elementos, datos que normalmente son expresados a través de símbolos en matemáticas que son libres de tomar cualquier valor dentro del conjunto de números donde están definidos. Un ejemplo podría ser: la cantidad de veces que estornudamos en un día, la cantidad de veces que ladra un perro en un día, etc. Ejemplo: Si la variable fuera **Ladridos del perro Lomito en un día** ésta todos los días cambiaría su valor, o sea variaría. Ahora, ¿y si Lomito solo ladra cuando ve pasar el camión de Lipigas por afuera de la casa? Podríamos decir que: el número de veces que Lomito ladre estará relacionado con la cantidad de **Veces que pase el camión del gas afuera de la casa en un día** (nueva variable), por lo que podríamos concluir que SON VARIABLES DEPENDIENTES una de otra.

En contraste, si aparece la variable **veces que pasa el camión del gas por una calle lejana a la casa de Lomito**, Lomito no sabrá, por lo que no ladrará y la variable **Ladridos de Lomito en un día** NO estará relacionada a ésta, por lo que se concluye que ambas son VARIABLES INDEPENDIENTES una de otra.

Relaciones entre las 3 variables Voltaje- Resistencia- Corriente:

- Mientras mayor sea la variable **voltaje**, mayor es la (variable) **corriente**, la cual fluirá a través del cable o alambre.
- Mientras mayor sea la **resistencia** (variable), menor es la **corriente** que fluirá por el cable o alambre.
- Si queremos que fluya o circule menos **corriente**, podemos: aumentar la **resistencia**, reducir el **voltaje** o ambos juntos.



Figura A.6: Analogía relación voltaje- corriente- resistencia.

Ya en la práctica, para que exista una corriente, debe existir una fuente de voltaje (pilas, red eléctrica domiciliaria, batería, etc.) y un camino que guíe a la corriente (cables o alambres); dicho camino debe siempre ser cerrado para que pueda producirse dicho flujo.

Tierra → Voltaje positivo → componente → Tierra

¡OJO! si no hay un componente intermedio se produce un cortocircuito, que es equivalente a conectar el voltaje positivo con Tierra lo que puede con alta probabilidad quemar el microcontrolador y o algún otro componente conectado. Lo anterior si estas trabajando con voltajes pequeños no representa peligro además de perder tus equipos (pérdida económica), pero a voltajes más altos, puede costar vidas, es por esto que se debe ser cauteloso a la hora de conectar bien tus componentes y no correr riesgos.

¿Qué es un cortocircuito?

Imagina que el fondo de un sartén sea una pila, el mango sea la componente o resistencia y finalmente nuestra mano tocando el mango de esta como Tierra o el polo negativo de una pila (batería). Si comenzamos a acortar cada vez más el largo del mango del sartén, llegará un momento en el que nuestra mano quedará tan cercana al fondo del sartén que comenzaremos a quemarnos y finalmente cuando el mango desaparece tendremos que tomar el sartén directamente, sintiendo la totalidad del calor en nuestras manos. Lo mismo sucede con la corriente, si no existe componente intermedia que ponga resistencia a esta, se transmitirá su totalidad a tierra o al voltaje opuesto, ocasionando lo que es llamado un **corto circuito**. Un corto circuito en una red domiciliaria podría fácilmente ocasionar una explosión y luego un incendio. Si bien los aparatos electrónicos en su interior no suelen tener altas corrientes, como para dañar a una persona, si se produce un corto circuito podría fácilmente quemarse nuestro microcontrolador, como Arduino, u otras componentes de elevado valor.

Además, se debe estar muy atento a **cerrar el circuito** o como se suele decir también, cerrar el camino de la corriente. Sabemos que la corriente podemos asociarla a agua fluyendo por una manguera (cable) esta fluye desde el polo positivo al negativo (o tierra) de la batería u otro objeto que cumpla el fin de energizar, si esta manguera en algún punto es cortada, el flujo de agua ya no llegará a su destino y es lo que llamaremos **circuito abierto**.

¿Qué es un circuito abierto?

En cada circuito que se arme, nuestro objetivo es que la corriente siga el camino previsto, fluyendo desde el polo positivo al negativo sin ser interrumpida en ningún punto intermedio. En una carrera de relevos se busca algo similar al hacer correr entre los participantes el testigo (barra) completando un camino para llegar a la meta propuesta.

Imagina que el testigo es la corriente, y el circuito que recorrerá solo se podrá energizar si se termina esta carrera de relevos, si en cualquier lugar de la carrera el testigo no se transfiere, entonces el circuito no tendrá energía y no funcionará, por lo tanto, se debe ser capaz de seguir con la vista este camino hasta llegar a tierra (o polo negativo), solo de esta forma habrás energizado el circuito deseado de forma correcta.



Figura A.7: Analogía circuito abierto.

Actividad N°1: *¡el camino si importa!*

Tu tutor o tutora planteará configuraciones de circuitos en donde deberás poder reconocer si el circuito presenta en algún lugar un **corto circuito** o **circuito abierto**. Debes reconocer el lugar puntual dentro del circuito en donde ocurre (no se transfiere el testigo), además de indicar qué camino toma la corriente antes de llegar a este punto.

Descanso de 15 minutos

Señales digitales y Analógicas

Una señal eléctrica corresponde a una señal que nace de la interacción entre las variables antes nombradas voltaje, corriente y resistencia.

Existen dos tipos de señales eléctricas, las digitales y las analógicas o análogas:

- **Señal Digital:** Es una señal no continua que suele tomar solo dos valores posibles. Particularmente en Arduino una señal digital toma valores de 0 y 5[V], voltajes que se pueden interpretar como un 0 o un 1, como un falso o un verdadero, como un apagado o un encendido, respectivamente. Recuerda no confundir una señal binaria con una digital, una señal binaria es digital, en cambio una señal digital no es necesariamente binaria, pero sí tendrá dos valores posibles.
- **Señal Analógica:** Es una señal continua que tiene la libertad de tomar cualquier valor dentro de algún rango específico. Particularmente en Arduino, este rango suele ser entre 0 y 5[V], por lo que la señal puede tomar valores intermedios (ej: 3[V], 4.21[V], etc).

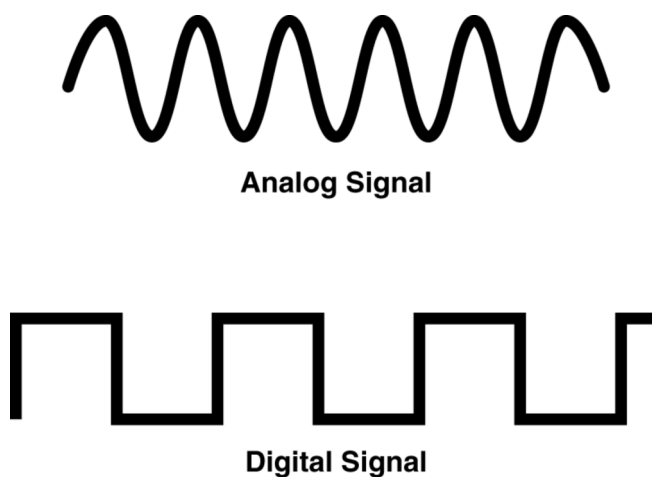


Figura A.8: Ejemplo señal análoga y digital.

Actividad N°2: *nuestro mundo digital y análogo*

Identifica si las siguientes señales son digitales o analógicas

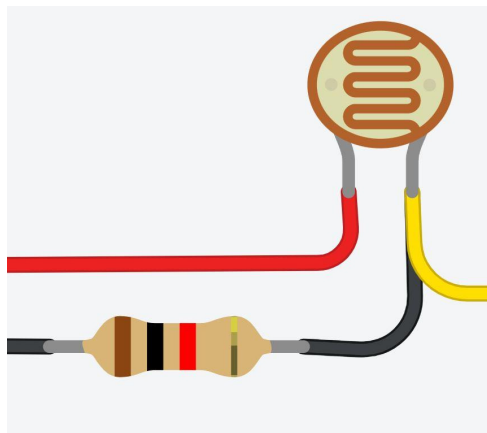
1. La temperatura.
2. La Hora en tu celular.
3. La Información almacenada en un Pendrive.
4. Tu altura/peso.
5. La edad de una persona.
6. La distancia entre nosotros.
7. Eventos como la lluvia, un temblor, etc.(suceden o no suceden)
8. El interruptor de luz.

Conexionado de componentes

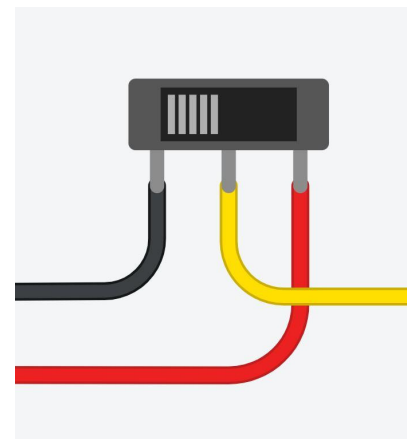
Dado que la base de la electrónica es guiar y controlar la cantidad de corriente que fluye por nuestros componentes, es de vital importancia saber cómo conectarlos correctamente. Hasta ahora solo lo hemos asumido o se han entregado imágenes de circuitos ya conectados, por lo que hoy vamos a formalizarlas.

La mayoría de las conexiones (unión de dos elementos, en este caso: conductores) son de 2-3 terminales (lugares de los componentes destinados a la conexión- comunicación), dos de ellos comúnmente son para ser conectados a +5V y GND (recordar que GND es ground, tierra en inglés) pertenecientes al camino principal y un tercero para una señal de información que sea entregada (sensores) o recibida (leds, motores, etc) por el componente, algunas conexiones requieren de resistencias adicionales para evitar realizar cortocircuitos o moderar corriente circulante por el circuito.

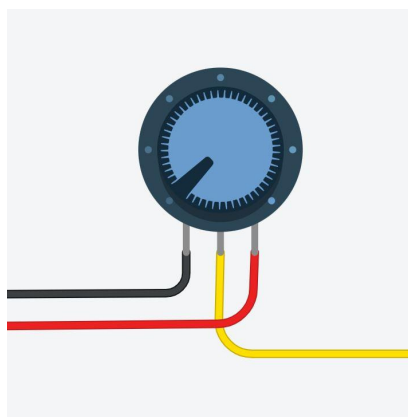
A continuación, se muestran algunas conexiones de los sensores más útiles para este curso, se utilizará por convención +5V, GND y Señal (recibida o enviada por el componente).



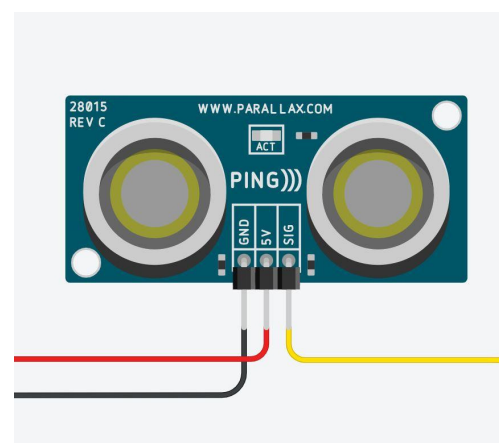
(a) LDR - La resistencia evita un cortocircuito y permite que en la señal se lea un valor de voltaje entre +5V y GND según la cantidad de luz.



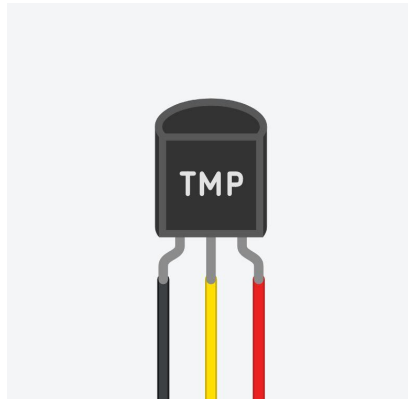
(b) Interruptor - Da el paso de corriente directa en un sentido u otro (no necesita resistencia) En la señal se lee +5V o GND según su posición (es similar al interruptor de una lámpara).



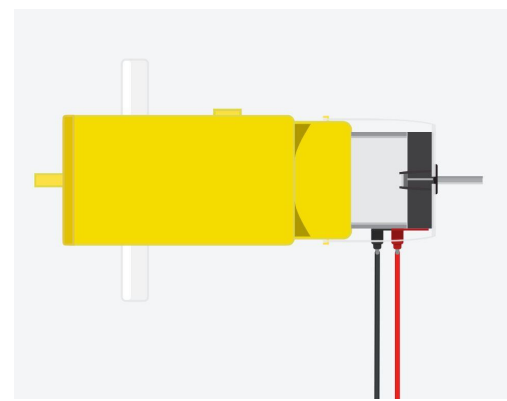
(c) Potenciómetro- Al ser ya una resistencia en sí, no se necesita otra adicional. En la señal se lee un valor entre +5V y GND según su posición.



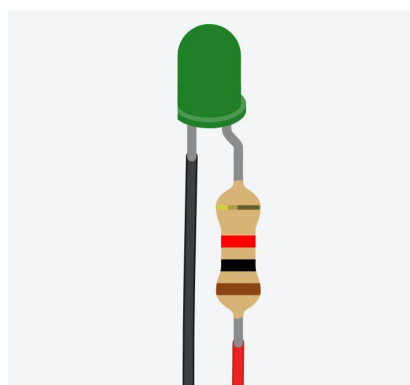
(d) Ultrasonido - Se usa para estimar la distancia a partir del valor de señal, se energiza con +5V y GND.



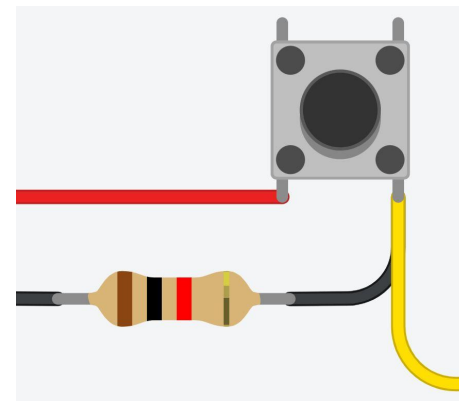
(e) Sensor temperatura - Se utiliza para estimar la temperatura a partir del valor de señal leída en este, se energiza con +5V y GND.



(f) Motor de corriente continua. Se conecta a +5V y GND para que gire en un sentido. Si se conecta al revés, el motor gira en sentido contrario.



(g) LED - Es necesario regular (disminuir) corriente con una resistencia.



(h) Botón - La resistencia permite que en la señal se lea +5V al presionarse y GND al soltarse (pull up o pull down).

Figura A.12: Conexión de componentes básicas para el curso.

Circuitos funcionales en Tinkercad

En Tinkercad se hallan muchos componentes interesantes con los que se pueden armar diversos circuitos, pero para comenzar hay circuitos funcionales base que se pueden armar y analizar, obteniendo aprendizajes que pueden ser útiles a futuro, algunos de estos son:

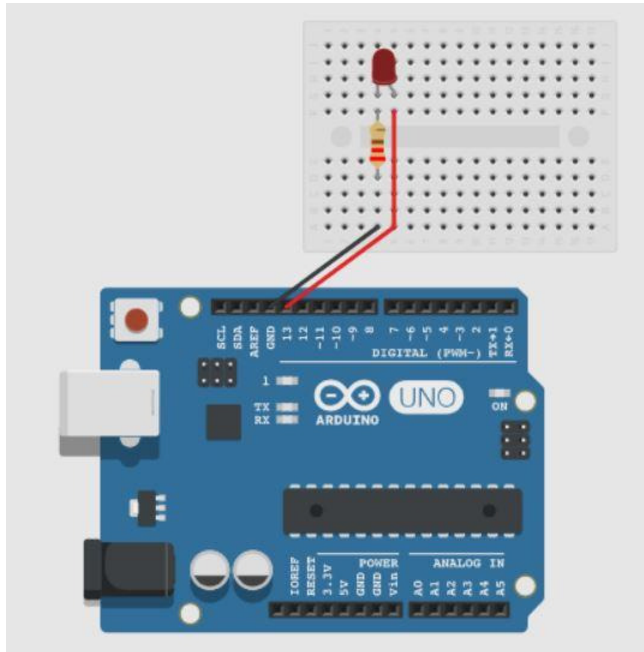


Figura A.13: Circuito básico led.

El anterior circuito se trata de una conexión simple para encender un led, como se observa va conectado a una resistencia con el objetivo de reducir la corriente que circula por el led, evitando este se queme y deje de funcionar. En este caso el led esta conectado a un pin digital, no existe un pin digital específico para esto, por lo que cualquiera que tenga esta característica puede ser utilizado, aunque debes tener en cuenta cual es al momento de programar.

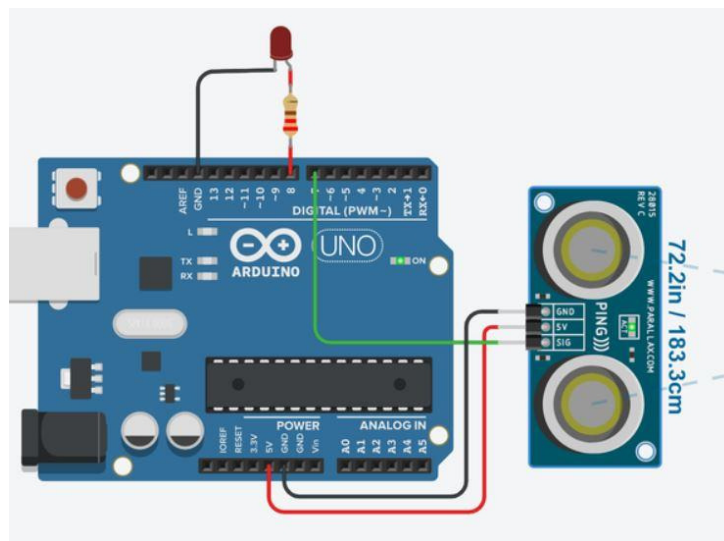


Figura A.14: Alarma de proximidad con led.

Este circuito puede ser programado como una alarma de proximidad, esto quiere decir que cualquier objeto o ser que se acerque y sobre pase a cierta distancia del sensor de ultra sonido, hará encender el led rojo automáticamente. Se observa nuevamente la resistencia conectada con el led, para regular corriente. Además, se puede observar la conexión de un sensor de ultrasonido.

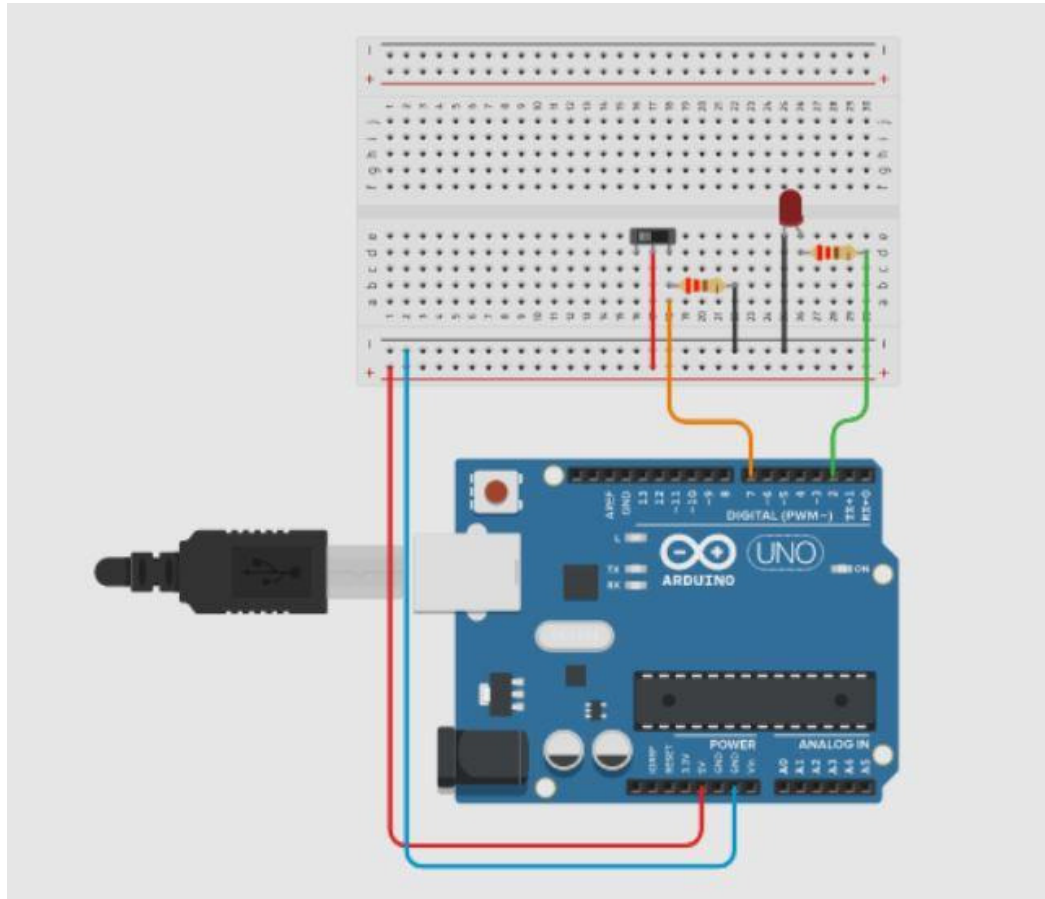


Figura A.15: Led con interruptor de encendido.

Finalmente este circuito funciona con un led que se enciende utilizando el interruptor, ambos van conectados a una resistencia, el led por temas de regulación de corriente y el interruptor para obtener un valor de tierra cuando el interruptor se “abra” y uno de Vcc cuando el interruptor se “cierre” (para más dudas, no dudes en consultarle a tu tutor o tutora).

Cierre

Coméntale a tus guías aquellas actividades que más te gustaron y aquellas que te costó seguir o ejecutar, para que puedan ser reforzadas en las sesiones siguientes.



Academias de Tecnología Online

Sesión N°7

7 de julio de 2023

Sesión N°7: Funciones de Arduino: señales, sensores y actuadores..

Objetivo principal: Conocer y definir funciones básicas de Arduino que permitan trabajar con señales.

Objetivo secundario: Introducir componentes sensores y actuadores como medios de comunicación con el exterior.

Muy importante:

Recuerda mantener el clima de respeto y las reglas de buena convivencia.

Iniciando

Siempre asegúrate de quedar presente en el control de asistencia.

Funciones para señales leídas por Arduino

Si bien las siguientes funciones ya se han usado anteriormente, ahora se verán en detalle:

- **digitalRead(pin):** Permite leer el valor digital del pin especificado, la función leída puede tomar el valor 0 o 1. Esta lectura se perderá si no es guardada en alguna variable. Ejemplo: Si le pidiéramos a un amigo que lea nuestro número de teléfono desde un papel, lo hará sin problemas, pero si le pedimos lo repita de memoria un minuto más tarde, no podrá, ya que solo fue lectura, no fue almacenado en su memoria, lo mismo sucede con esta función, es solo lectura en el instante en el que se ejecuta la línea de código.
- **digitalWrite(pin,Estado):** Permite escribir el estado especificado (HIGH o LOW), como un valor digital (5[V] o 0[V] respectivamente), en el pin indicado.
- **analogRead(pin):** Esta función permite leer el valor de un pin analógico especificado en su entrada, para esto reconoce el rango de voltajes de 0 y 5 [V] y los transforma en una variable de rango entre 0 y 1023 (1024 niveles en que se puede encontrar la variable, ya que se cuenta el cero) que es la que arroja esta función al ejecutarla.
- **analogWrite(pin,Valor):** Esta función permite escribir en un **pin analógico** valores intermedios entre 0 y 5[V]. Los pines analógicos corresponden a los números 3, 5, 6, 9, 10,

y 11 (pines acompañados por un ~). A la variable valor se le pueden asignar valores entre 0 y 255, lo que permite generar 256 voltajes intermedios entre 0 y 5[V]. Véase las Figuras 1 y 2 para una mejor comprensión:

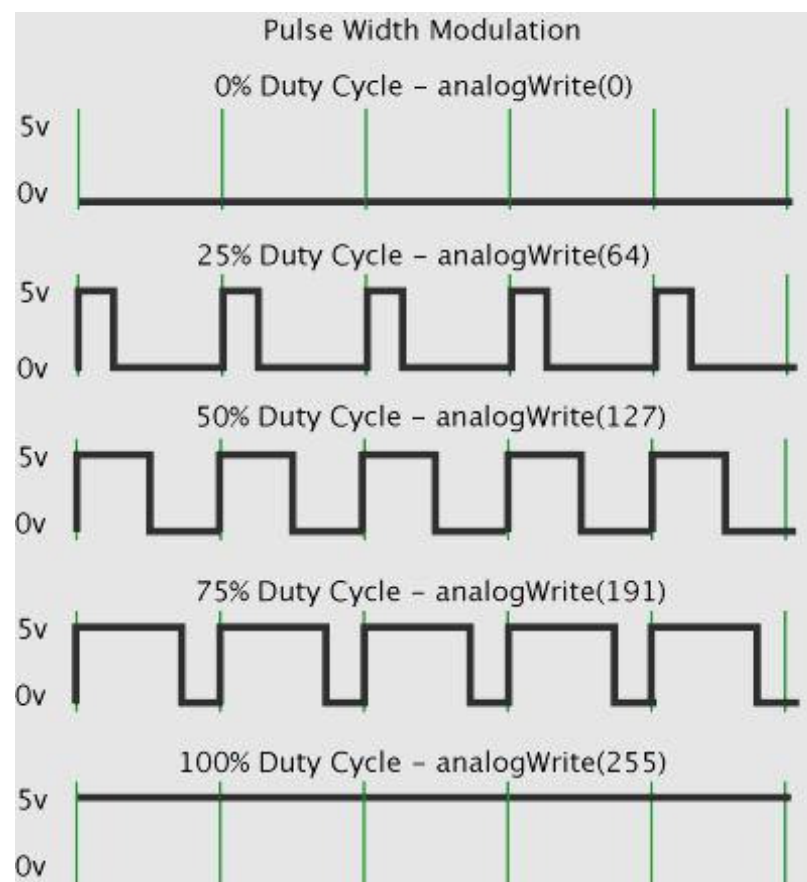


Figura A.16: Ancho de pulsos de la función `analogWrite(Valor)`, según variación de la variable **Valor**.

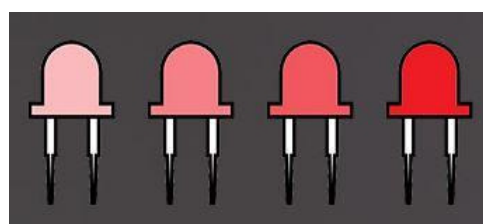


Figura A.17: Efecto del cambio en anchos de pulso en un led alimentado por estos.

- `pinMode(pin, mode)`: Corresponde a la función que define si los pines se comportarán como entradas o salidas. En la casilla pin ingresamos el número del pin digital o análogo. En mode ingresamos INPUT y OUTPUT según sea el caso.

Actividad N°1: *y así se hizo la luz.* (20 minutos)

Uno de los aldeanos de tu ciudad necesita implementar un sistema de iluminación para una nueva habitación de su casa que utilizará como estudio.

El estudio necesitará dos tipos de iluminación:

- Una lámpara capaz de encender solo si se mantiene presionado un botón.

- Una lámpara que sea capaz de regular su intensidad según lo que necesite el usuario por medio de una perilla (potenciómetro)

Deberás realizar el correcto conexionado (conexión de cables) y programación del microcontrolador Arduino para el manejo de los componentes solicitados de acuerdo a los requisitos planteados anteriormente.

Ayuda: Para convertir valores en el rango de 0-1023 a valores entre 0-255, una buena aproximación se logra con multiplicar el valor que se tiene entre 0-1023 por 0.25, siendo así el resultado el valor entre 0-255 equivalente.

O se puede hacer uso de la función `map()`, ésta devuelve como resultado el mapeo (se halla el valor correspondiente en otra escala) del número deseado en un rango distinto al actual:

```
map(a,b,c,d,e);
```

En donde,

- a: Valor de entrada a convertir o mapear.
- b: Valor inicial del rango de entrada.
- c: Valor final del rango de entrada.
- d: Valor inicial del rango de salida.
- e: Valor final del rango de salida.

En este caso se utilizaría de la siguiente forma:

```
valor_convertido=map(valor_a_convertir,0,1023,0,255);
```

La anterior línea guardará el valor que necesitamos en `valor_convertido` para su utilización.

Link circuito base para la actividad:

<https://www.tinkercad.com/things/hcBFnztmUaD>

Descanso de 15 minutos

Sensores y Actuadores

¿Cómo conocer mis componentes?

Si deseas saber las especificaciones de un modelo específico de componente, cómo se conecta, para qué se utiliza, nombre técnico, debes revisar su hoja de datos (se denomina en inglés: datasheet).

Reconoceremos por ahora dos tipos de componentes: Actuadores y sensores. Llamaremos actuador a cualquier componente que recibe una señal eléctrica del sistema en este caso del microcontrolador (Arduino) y transforma ésta en energía de uso común, puede ser mecánica o energía eléctrica para alimentación y llamaremos sensor a cualquier componente que reciba

una señal eléctrica de alimentación (sin información) y entregue una señal con información, datos, etc. hacia el sistema (Arduino).

Actividad 1: *Reconocer actuadores y sensores*

Tomando en cuenta lo aprendido, ahora debes reconocer si cada componente electrónica que se muestre en la siguiente lista es un actuador o sensor, y debes dar una explicación de por qué consideras lo es:

- Pantalla (display).
- LM35
- Botón.
- Bocina/parlante.
- Potenciometro.
- Led
- Fotorresistencia.
- FC-28.
- HC-SR04.
- PIR HC-SR501.

Desafío: *¡Con mi queso no!*

El ratón Jean Paul, amigo francés de Condensín vino de intercambio a estudiar a las Academias de Tecnología y luego de asistir a todas las sesiones decidió hacer un súper proyecto: un sistema seguridad para su queso máspreciado, un queso francés con 10 años de maduración y hecho con leche de vacas rosadas y que pastan en los jardines de la Torre Eiffel.

El sistema de seguridad que él necesita debe cumplir las siguientes funciones:

- Debe encender un led cuando haya movimiento al rededor de su lujoso queso.
- Debe enviar un mensaje a través del monitor serial que diga: **¡Policía, se llevan mi queso!**, cada vez que Jean Paul apriete un pulsador.

Para estos requisitos el ratón Jean sabe más o menos qué necesita, así que en su kit para el proyecto tiene:

- Un sensor PIR
- Un led rojo
- Un pulsador

- Dos resistencias
- Una protoboard
- Un Arduino
- Y por su puesto un trozo de queso de colación, porque de seguro nos dará hambre trabajando.

Con toda la información anterior ayuda al ratón Jean Paul a hacer una alarma para su queso súper lujoso.

Cierre

Coméntale a tus guías aquellas actividades que más te gustaron y aquellas que te costó seguir o ejecutar, para que puedan ser reforzadas en las sesiones siguientes.

B | Anexo

Este anexo contiene en orden numérico las sesiones desarrolladas para uso de tutores, desde la N°0 a la N°7.



Academias de Tecnología Online

Sesión N° 0

7 de julio de 2023

Sesión N°0: Bienvenidos y bienvenidas a la aventura ACATEC.

Objetivo principal: Evaluar/ diagnosticar conocimientos respecto a los contenidos a abordar durante este año en las academias.

Objetivo secundario: Generar un ambiente de confianza, permitiendo compartir y conocerse.

Bienvenida (5 minutos)

Presentación personal de los tutores y tutoras. Para ésta se recomienda que los tutores o tutoras a cargo activen sus cámaras, para que los y las estudiantes puedan identificar quienes serán sus guías el resto del año.

El objetivo del tutor o tutora en esta sesión será, además de cumplir los objetivos estipulados, motivar en la medida de lo posible a los y las estudiantes para que estos acudan regularmente a las sesiones. Se debe tratar de interactuar con estos para mantener su atención y participación.

Se debe promover siempre un clima de respeto en las sesiones, si bien uno de nuestros objetivos es crear un ambiente acogedor para el o la estudiante y que se sientan en confianza de preguntar y mostrar sus capacidades, no debemos dejar de lado el recalcar antes de hacer presentaciones de proyectos o actividades grupales, que cualquier crítica y opinión debe ser hecha con respeto hacia sus compañeros y compañeras. Caer en conductas que entorpezcan el desarrollo de la clase de forma reiterada, faculta al tutor o tutora para dejar al estudiante en sala de espera y de continuar este tipo de actitudes, la expulsión del estudiante, previo aviso al establecimiento (labor llevada a cabo por Coordinación ACATEC informada previamente por los tutores encargados del establecimiento).

Diagnóstico (7 minutos)

Tras la entrada a la plataforma Zoom, se realiza una introducción a lo que son las Academias de Tecnología, con los objetivos que tiene y el desarrollo de las habilidades que busca lograr en sus estudiantes.

“ACATEC o las Academias de Tecnología es una iniciativa cuyo propósito es llevar a escolares la curiosidad y el trabajo colaborativo desarrollando paralelamente el

pensamiento crítico y reflexivo. Buscamos transferir el conocimiento y la tecnología a través del desarrollo de actividades prácticas y didácticas, guiadas por estudiantes del Departamento de Electrónica de la UTFSM”

Los y las estudiantes deberán desarrollar una pequeña prueba de diagnóstico que permitirá establecer el nivel de conocimiento que poseen. En base a esto podrán ser ubicados en los niveles básico o avanzado según corresponda, para esta actividad cuentan con **30 minutos**, se debe Compartir el enlace en el chat de Zoom.

Enlace al test de diagnóstico:

<https://forms.gle/NwGo6A1EamoQRqKU6>

Descanso de 15 minutos

Dinámica rompe hielo: ¡Sálvese quien pueda!

Instrucciones

Se dividirán los y las estudiantes en salas, de forma que en cada sala hayan máximo 8 participantes, incluyendo al tutor o tutora. Se leerá el contexto del caso *Tribu de caníbales* para luego realizar un sorteo en la plataforma <https://echaloasuerte.com/groups> para definir el personaje que adoptará cada estudiante. Cada grupo deberá solucionar el caso, en conjunto

A partir de este momento dispondrán de 20 minutos para darle solución al caso. La respuesta será expuesta por un representante que el grupo elija.

El evaluador registra las conductas que cada integrante evidencia durante la fase de debate grupal.

Caso: *Tribu de caníbales*

En un viaje camino a la gran competencia internacional ACATEC, familiares de los y las estudiantes sufren un accidente quedando varados en medio del Amazonas donde son capturados por una tribu caníbal (se alimentan de carne humana).

Con la carne del piloto, la tribu tiene para alimentarse una semana, por lo tanto, irán matando cada semana a una persona. El resto será cuidado para las siguientes semanas. Los ocho (u otro número) deben debatir y llegar a un acuerdo para establecer el orden en el que irán a la “olla” y además, tienen que buscar una forma para escapar con vida y llegar a la civilización.

Quienes participan deberán adoptar el personaje de acuerdo a la asignación del tutor o tutora (ayudado por una página de selección al azar) y, desde la perspectiva del personaje, defender su vida, negociando para ser los últimos en ser comidos:

Personajes:

- Una cantante de opera obesa que sabe medicina.

- Un chef delincuente joven y fuerte.
- Una profesora con dificultad para caminar que sabe técnicas de caza.
- Un anciano que conoce la zona.
- Un cerrajero que no puede ver.
- Un biólogo enfermo.
- Un encantador de serpientes sordo.
- Una interprete que conoce el idioma de la tribu.



NOTA: Antes de la Sesión N°1 se debe agregar a Classroom a todos los y las participantes utilizando sus respectivos correos y nombres. Puede hallar más información en la guía de la sesión correspondiente.



Academias de Tecnología Online

Sesión N° 1

7 de julio de 2023

Sesión N°1: Herramientas de trabajo.

A los tutores y tutoras:

Se recuerda que para la creación de los equipos de trabajo sesión a sesión, se pueden agrupar a criterio del tutor, tutora o al azar. Aplicación anteriormente usada:

<https://echaloasuerte.com/groups>.

Se debe agregar a Classroom a todos los y las participantes utilizando sus respectivos correos y nombres. Con la ayuda de los establecimientos en la obtención de los correos de los participantes, **este proceso debe llevarse a cabo antes de la Sesión N°1**, esto ahorrará mucho tiempo.

Debe de asignarles nombres genéricos de grupo. También debe añadir cada grupo a las salas (classroom) de Tinkercad de la siguiente forma: Nombre: Integrantes — Alias: *grupo1*.

Objetivo principal: Conocer las herramientas a emplear durante el curso del año y el objetivo de las Academias de Tecnología.

Objetivo secundario: Fortalecer el pensamiento computacional mediante actividades lúdicas e interactivas.

Muy importante:

Recordar a los y las estudiantes mantener el clima de respeto y las reglas de buena convivencia. Tomar asistencia.

Iniciando

Zoom (7 minutos)

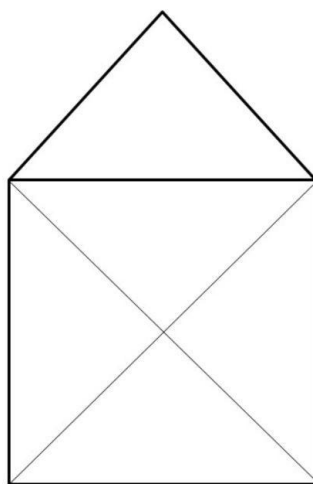
Tras la entrada a la plataforma Zoom, se realiza una introducción a lo que son las Academias de Tecnología, con los objetivos que tiene y el desarrollo de las habilidades que busca lograr en sus estudiantes.

Ya que los y las estudiantes tenían el deber de revisar en internet información y las herramientas más utilizadas de Zoom, se debe intentar hacer esta sección en forma de repaso y como demostración, además de aclarar dudas. **Funcionalidades de Zoom (Demostración rápida):**

- Compartir pantallas: Mostrar la pantalla a todos los integrantes de la sala, también se puede compartir audio.
- El Chat: Usado para hacer preguntas, compartir enlaces y archivos.
- Anotación: ¡A dibujar! Invitar a los y las estudiantes a dibujar algo en la pantalla. Recordar a los y las estudiantes que sean responsables al dibujar, para que no ocurran situaciones no deseadas.
- Grupos pequeños: A probar más adelante.
- Votaciones: Ojalá crear una pregunta pequeña de prueba para que los y las estudiantes puedan responder.

Actividad 1: *¡Buscando el camino!* (10 minutos)

Se les presenta la actividad en donde utilizando anotación deben dibujar una casa como la que se presentará (imagen a compartir en material de apoyo, como PPT), sin levantar el lápiz y sin dibujar las aristas más de una vez.



Herramientas a utilizar (5 minutos)

Se presentan las Plataformas Tinkercad y Google Classroom, las cuales son las herramientas que se utilizarán a lo largo de las academias y se desarrollarán con mayor detalle en este documento.

Classroom

Ingreso a la plataforma (5 minutos)

Enviar la invitación a los participantes de las academias para que se registren y queden habilitados para su participación en las clases.

Recorrido por la plataforma (3 minutos)

Se debe comenzar confirmando que todos los asistentes han aceptado la invitación a Classroom y que todos pueden acceder a ella. Luego, hacer un recorrido por lo que será el sitio oficial de informaciones y actividades de las academias durante toda su duración.

Probando el pensamiento lógico (10 minutos)

Los y las estudiantes deberán ingresar al apartado de *QUIZ 1* que se les presentará en la pestaña de *Trabajo en clase* de Classroom. También puede hacer ingreso directamente desde el “tablón” en el cuadro con la leyenda “*Quiz número 1*”. El objetivo de este quiz es conocer mejor las posibilidades de la plataforma y obtener una idea de cuán desarrollado tienen su pensamiento lógico. Hacer hincapié en que éste puede ser ejercitado y que el resultado que obtengan ahora, no refleja el máximo potencial que pueden alcanzar.

Descanso de 15 minutos

Tinkercad

Presentación de la plataforma (5 minutos)

Se presenta la plataforma de trabajo a modo de introducción, mostrando las funcionalidades disponibles, como:

- Personalización del perfil a utilizar.
- Áreas de modelado 3D.
- Áreas de diseño de circuitos.
- Áreas de programación por bloques de código

Equipos de trabajo (3 minutos)

Se les presenta a los y las estudiantes los equipos de trabajo designados; dichos equipos deben tener nombres genéricos (para el posterior ingreso) como se muestra aquí:

Grupo 1	Pepito Palote Juanita Juanete Suzana Horia
Grupo 2	Armando Casas Marcia Ana Elsa Pato

Registro en la plataforma (5 minutos)

Se les solicita a los y las estudiantes ingresar al enlace de la clase brindado por el tutor o tutora (clase creada previamente). Una vez ingresado al enlace deberán ingresar con los nombres de su grupo de la siguiente forma, Ej: El alias del grupo 1 es *Grupo1*. Se les pide que posteriormente junto a su pareja decidan una palabra clave para cambiar *Grupo1* por otra clave para así otros compañeros no tengan acceso a sus archivos y evitar problemas que pudieran surgir. Posteriormente, tienen la opción de personalizar sus perfiles.

Desafío

Interfaz 3D (10 minutos)

Se presenta la interfaz de diseño 3D de Tinkercad. Entre los temas a tratar se debe hacer especial énfasis en:

- **Dominio de espacio tridimensional:** Se presenta el plano de trabajo, con sus ejes coordenados. Se deben mover figuras a través del plano y mostrar sus dimensiones, esto ayudará a comprender mejor las características del espacio.
- **Escala y desplazamiento:** Se debe mostrar que, al seleccionar una figura, ésta puede ser escalada, tanto de sus aristas como sus vértices. Además, puede ser desplazada arrastrándola con el puntero en el plano X-Y, mientras que para ser desplazada en el plano Z debe hacerse con la flecha en la parte superior.
- **Colores:** Puede escogerse entre una gama de colores para cada objeto seleccionándolo y pinchando en *Sólido*. Por otro lado, la opción *Hueco* indica que la figura será usada para recorte.
- **Recorte y grupos:** Cuando se intersecan dos figuras es posible fusionarlas con la opción *Agrupar* o apretando Ctrl+G. De intersecar y agrupar una figura sólida con una hueca hará que la figura hueca “recorte” un trozo de la figura sólida, posibilitando la creación de nuevos tipos de figura.

Proyecto: *Casa* (25 minutos)

Se les solicita a los y las estudiantes que, junto a su pareja, diseñen una casa usando la funcionalidad 3D de Tinkercad. Usando la función de grupos pequeños en Zoom se divide a

los estudiantes según su pareja y se les entrega el tiempo suficiente para trabajar.

Presentación del proyecto (8 minutos)

En los últimos minutos de la sesión se les envía a todos y todas a la sala principal de Zoom y se comparte a todos los diseños de sus proyectos. Dejando unos minutos para el cierre.

Cierre (5 minutos)

- Hacer un recuento de las actividades que se ejecutaron. Solicitar a los y las estudiantes que señalen qué aprendieron, qué les gustó más y qué les resultó más complicado. El tutor o tutora, debe considerar estos comentarios para las próximas sesiones.
- Verificar asistencia nuevamente.
- Comentar que si algún o alguna estudiante no alcanzó a mostrar su proyecto, lo podrá mostrar la siguiente sesión.
- Despedirse y recordar horario de próxima reunión.



Academias de Tecnología Online

Sesión N° 2

7 de julio de 2023

Sesión N°2: Introducción a la programación.

A los tutores y tutoras:

Una semana previo a esta actividad se debe agregar a Google Classroom la tarea de dibujar el boceto en papel de una casa y un avatar personal utilizando figuras geométricas básicas (esferas, cubos, cilindros, estrellas, etcétera). Además, se debe verificar la entrega de los bocetos a través de la misma plataforma.

Objetivo principal: Realizar un acercamiento a la programación por medio del uso de bloques de código.

Objetivo secundario: Conocer nuevas funcionalidades de la plataforma Tinkercad manipulando objetos 3D.

Muy importante:

Recordar a los presentes mantener el clima de respeto y las reglas de buena convivencia.

Iniciando

Revisión de tarea (7 minutos)

Es importante dar comienzo a esta sesión revisando la tarea asignada durante la semana, destacando aquellas que cumplen bien con los requerimientos solicitados por el tutor o tutora en la actividad.

Explicar los objetivos e importancia de la sesión. Principalmente lo que se busca es que los y las estudiantes sean capaces de comprender que los computadores suelen recibir una instrucción a la vez y que son ejecutadas de manera secuencial, para ello se utilizará la programación en bloque que es una herramienta visual e interactiva.

En esta sesión las actividades se realizarán de manera individual, por ello, se habilitará una sala especial de consultas en la plataforma Zoom. Por otro lado, los presentes serán libres de realizar sus preguntas en la sala principal o ir a esta sala de consultas. De todas formas, durante la clase se deberá consultar a cada estudiante sus avances en esta sala.

Presentación bloques de código (10 minutos)

Todo computador requiere de instrucciones para funcionar, una de las formas más sencillas de programar es mediante bloques de código. En esta clase se utilizarán los bloques para volver realidad la casa que diseñaron en la tarea dada la semana pasada.

Esta programación consiste en utilizar bloques que están asociados a cada instrucción, donde también es posible enlazar un bloque dentro de otro para formar grupos de instrucciones, los que pueden ser tratados como si fuesen una sola sentencia.

Se debe ejecutar un ejemplo simple de lo enseñado frente a la clase.

Programación en bloques I

Ingreso a la plataforma (5 minutos)

El siguiente paso a paso debe ser mostrado a los asistentes, con el objetivo de agilizar la creación del proyecto y que nadie quede atrasado.

Cada estudiante debe ingresar a la plataforma Tinkercad usando el alias otorgado por su tutor o tutora. Luego debe ir a la sección *Bloques de código* y seleccionar *Crear nuevo Bloque de código*. Ya que las actividades se realizarán de manera individual, una vez creado el proyecto, debe asignarle un nuevo nombre.

Figuras y formas (3 minutos)

Arrastrar el bloque *Crear nuevo objeto* (ubicado en la sección *Modificar*) hacia el plano de trabajo. El nombre del objeto puede ser modificado seleccionando *Renombrar variable*. A continuación, se elige alguna figura disponible en la sección *Formas* y se arrastra hacia el código de tal manera que este nuevo bloque quede ensamblado al anterior. El color y las dimensiones de la figura se modifican en el mismo bloque.

Para recortar una figura, se debe añadir otra figura al interior de esta (con la forma que se desea eliminar) y marcar la opción *Hueco* y añadir un bloque púrpura llamado *Crear grupo*.

Se debe mostrar un ejemplo del resultado final que se pretende obtener. Además, mostrar el proceso de una figura básica.

Simulación (5 minutos)

Es necesario indicar que para observar el progreso en el plano 3D, se debe presionar el botón presente en la barra superior derecha. También se puede ajustar la velocidad de simulación y avanzar paso a paso. Puede ser ejecutado un ejemplo simple para mayor comprensión.

Movimientos (5 minutos)

Se presentan las funciones del tipo **Modificar**, específicamente, realizar un ejemplo con dos estrellas (u otra figura a gusto) de los comandos *Desplazar*, *Girar*, *Escala*, *Crear grupo* y *Crear nuevo objeto*.

Actividad N°1: ¡Nos mudamos de casa!(15 minutos)

Al igual que la semana pasada, se le solicita a la clase diseñar una casa nueva, con la dificultad agregada de usar solamente programación en bloques.

Actividad N°2: *El nacimiento de un avatar* (15 minutos)

Utilizando programación en bloques deberán crear su propio avatar. Se comparten los trabajos entre los y las asistentes.

Descanso de 15 minutos

Programación en bloques II

Bloques matemáticos (5 minutos)

Usando el bloque *Crear variable* se simplifica el cambio de algún valor que se utiliza varias veces. A modo de ejemplo, cambiar la configuración de una figura al reemplazar todas sus dimensiones por un dato. Luego, con el bloque *Definir* o *Cambiar* se puede asignar un nuevo valor a la variable creada. Para realizar operaciones matemáticas se presenta el bloque de suma, que puede ajustarse para hacer restas, multiplicaciones o divisiones entre dos números (para operaciones entre más números se debe incluir un bloque dentro de otro). Mostrar con ejemplos.

Bloques de control (5 minutos)

Los bloques anidados permiten repetir un grupo de sentencias, como el bloque *Repetir x Tiempos* y el bloque *Contar con j desde 1 a 10*, que realizan acciones similares según su configuración.

Se propone desplazar una figura en repetidas ocasiones (el bloque *Añadir* está antes del bloque *Anidado* y el bloque *Desplazar* está anidado) para generar múltiples figuras (ocurre cuando el bloque *Añadir* o *Copiar* está anidado).

Desafío

La Aldea Konoha/Kage Bunshin no Jutsu (15 minutos)

Con ayuda de los bloques *Crear nuevo objeto* y *Añadir copia de objeto*, se solicita a la clase crear una pequeña aldea basada en la casa que crearon anteriormente, además, pedirles que programen un recorrido que su avatar pueda realizar de forma cíclica alrededor de la aldea.

Revisión (5 minutos)

Se solicita retornar a la sala principal de Zoom, se compartirán los diseños y simulaciones de sus proyectos donde cada estudiante debe explicar lo que buscaban realizar y la lógica que utilizaron para resolver los desafíos.

Para evaluar la actividad, aplicar pauta de evaluación disponible.

Cierre (10 minutos)

- Recordar las actividades que ejecutaron. Solicitar a la clase que señalen qué aprendieron, qué les gustó más y qué les resultó más complicado. Se deben considerar estos comentarios para las próximas sesiones.
- Verificar asistencia nuevamente.
- Despedirse y recordar horario de próxima reunión.
- Informaciones varias.



Academias de Tecnología Online

Sesión N° 3

7 de julio de 2023

Sesión N°3: Programación de Arduino mediante bloques de código.

Objetivo principal: Implementar la programación de Arduino haciendo uso de bloques de Código.

Objetivo secundario: Comprender la importancia y la utilidad de los diagramas de flujo al generar un algoritmo.

Muy importante:

Recordar a los y las estudiantes mantener el clima de respeto y las reglas de buena convivencia.

Iniciando (5 minutos)

Se saluda a la clase mientras se verifica asistencia.

Concepto de código (5 minutos)

En la clase anterior, mediante la herramienta de **Bloques de Código** de Tinkercad, se vió cómo con un conjunto de instrucciones (bloques) junto a una máquina como el computador, pueden realizar algo bastante complejo como un diseño 3D de una villa, solo juntando instrucciones aisladas. Con ese ejemplo, se les puede dar a entender a los estudiantes de lo torpe que son los computadores, o si lo quieren ver de otra forma, lo complejas que son las decisiones que se toman cada día.

En nuestro día a día no nos molestamos en reconocer cada paso que damos o decisión que tomamos, estas son tomadas casi inconscientemente. Sin embargo, para ejecutar una acción a través de código toma importancia cada decisión o paso y el orden en que estos son ejecutados.

Cada acción que se requiere realice el código, deberá ser dada de forma explícita, con la información que requiera para hacerlo. No olvidar recalcar esta información a la clase.

Para esto se usará una herramienta, la que se detalla a continuación en esta guía, ésta servirá como intermediaria física entre el ser pensante (estudiante) y el ser inerte que ejecuta acciones (un microcontrolador, por ejemplo), el idle de Arduino.

Actividad N°1: mi avatar tiene sed. (7 minutos)

Se le pide a la clase que individualmente le explique a un aldeano virtual de la Villa ACATEC cómo debe tomar un vaso de agua, para ello debe elaborar una lista de al menos cinco pasos, en un lugar visible para luego compartirlo, enumerando qué debe hacer para llevar a cabo esta tarea el aldeano (se sugiere comenzar con la instrucción *Ir a la cocina*).

Finalizando la actividad, debe ser reforzado el proceso realizado a través de ejemplos de diagramas de flujos: se solicitará a la clase rayar en la pantalla, utilizando la funcionalidad de Zoom, con alguna modificación que le harían a los diagramas para que estos sean más completos. (Ejemplo: preparación de algún platillo simple: puré de papas). Idealmente dar la posibilidad para presentar a la clase algún diagrama hecho por algún o alguna estudiante, propiciando la instancia de realimentación para los presentes.

Algoritmos

¿Qué son y para qué sirven? (1 minutos)

Se debe explicar que lo que acaban de crear es un algoritmo.

Se entrega una definición de algoritmo como “**un conjunto de instrucciones definidas y no-ambiguas, ordenadas y finitas que permite realizar una tarea**”. Se puede reforzar con algún ejemplo simple, o solicitar que los alumnos o alumnas los planteen de forma oral.

La finalidad de los algoritmos es que se escriba una solución a un problema o se obtenga una salida determinada, a partir de un estado inicial y una entrada, hallando los pasos mínimos que debe realizar el sistema. Se puede dar como ejemplo la resolución de los cubos Rubik, los cuales son posible de solucionarse solo siguiendo una serie de pasos ordenados, recalcar que esta no es magia ni una habilidad sobrehumana.

Diagramas de flujo (3 minutos)

Utilidad de los diagramas de flujo como la representación gráfica de un algoritmo:

- Para construir uno se debe identificar o determinar el inicio: **quiero un vaso de agua**, y el final: **tomar agua**.
- Entre los determinados inicio y final se deben incluir las acciones como: ir a la cocina, buscar un vaso, etc. Se pueden realizar preguntas para definir cuál sería siguiente paso: **¿el vaso está limpio?**, y según la respuesta entregada, se pueden tomar dos caminos: **Sí → Abrir llave y llenar el vaso con agua** o **No → Lavar el vaso**.
- No olvidar, que para evitar confusiones se deben incluir flechas que indiquen cuál es el siguiente paso.

A continuación, se presentarán dos ejemplos de Diagramas de Flujo. Los estudiantes deben proponer formas de mejorarlos. Se puede utilizar la siguiente pregunta guía ¿falta algo o hay un error en ellos?. Si hay poca motivación de los estudiantes por responder, se recuerda que puede focalizar la pregunta en un alumno o alumna en particular.

Actividad N°2: mi primer diagrama de flujo (7 minutos)

Deberán dibujar un Diagrama de Flujo lo suficientemente detallado para que cualquier persona entienda lo que debe hacer para seguirlo. El trabajo será individual con separación previa por Zoom en “n grupos según los n tutores disponibles”.

Se propone compartir con toda la clase un enlace de documento de Power Point, en donde cada slide le pertenecerá al diagrama de flujo de un grupo en especial. Power Point tiene las herramientas de formas que pueden ayudar a la creación del diagrama y además al trabajo grupal.

Finalizando la actividad se deben mostrar los diagramas y realizar un feedback a los y las estudiantes.

Circuitos I

Creación (5 minutos)

Se indica a la clase que ingresen a la plataforma Tinkercad haciendo uso del alias que les fue otorgado sesiones atrás (Sesión N°1). Luego en el panel de la izquierda deben ingresar a la sección *Circuitos* y selecciona *Crear nuevo circuito*. Una vez creado el proyecto, cada uno deben asignarle su propio nombre al proyecto ya que se ejecutará de forma individual.

Circuitos en Tinkercad (5 minutos)

Con el proyecto recién creado en pantalla, se deben presentar las diferentes funcionalidades de Tinkercad para circuitos a la clase, de forma que sea entendible su uso básico.

Se recomienda presentar rápidamente la plataforma de Circuitos de Tinkercad, y luego mostrar los bloques principales que se van a utilizar (definir pasador(), y esperar()), para luego pasar a las actividades que van a tomar más tiempo. Lo primero que se recomienda mostrar es el panel a la derecha que contiene las diferentes componentes desde LED's hasta circuitos básicos ya armados. Se debe mostrar cómo poner las componentes, conectar cables, inicio de simulación, etc.

Encendiendo un LED(7 minutos)

En la actualidad es difícil no conocer las luces LED. En la electrónica son una herramienta muy útil para corroborar o monitorear el estado de ciertas señales o acciones que se desean ejecutar (recordar que la palabra LED es una abreviación de Light Emitting Diode). Además, uno de los circuitos más básicos para armar y programar corresponde al de una batería y un LED/ampolleta (dibujar un circuito en forma de ejemplo). Proceder al armado del circuito frente a la clase de forma que puedan repetirlo en sus proyectos de forma individual. Terminando de armar el circuito, realizar de forma guiada las siguientes preguntas a la clase utilizando un circuito preparado:

1. ¿Se enciende el LED?(Poner el LED en la forma en que no circula corriente).
2. ¿Aparece una advertencia?(Probar conectar sin resistencia).
3. ¿Qué pasa si intercambias los cables?
4. Probar conectar una resistencia entre la batería y el LED ¿qué cambia?

Se reflexionan en conjunto las respuestas.

Si hay poca motivación de los y las estudiantes por responder las preguntas, intenta aludir con la pregunta a una persona en particular.

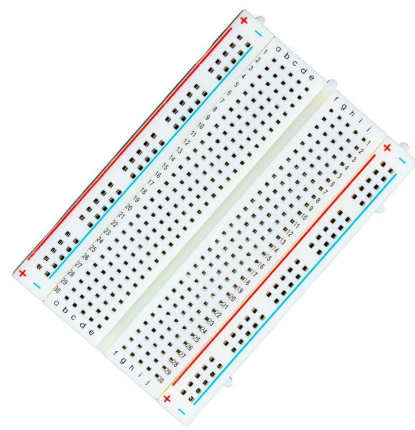
Teniendo en consideración la ley de Ohm ($V = i \cdot R$, esta debe ser explicada de forma breve para que los estudiantes la conozcan y entiendan a grandes rasgos), una batería de un voltaje muy elevado puede dañar al LED, por ello es necesario recalcar que se debe colocar una resistencia que disminuya la corriente que pasa por el circuito y por consiguiente el LED, mencionar que en Tinkercad se muestra una alarma mostrando esta problemática.

NOTA: No es necesario estancarse en dar grandes explicaciones sobre conceptos tan complejos como lo son la corriente o el voltaje, traten de ser puntuales y simples, ya que en las siguientes sesiones se dedicará mas tiempo a estas explicaciones, por el momento se busca que toda la clase sepa la parte técnica de cómo se debe y cómo no se debe conectar un LED.

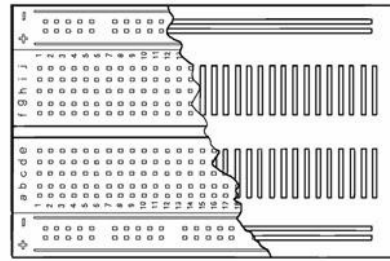
Protoboard: ¿Qué es y para qué sirve? (3 minutos)

Para diseñar circuitos se puede iniciar por crear un prototipo, y una herramienta muy útil para esto es la protoboard, la cual permite armar, desarmar o modificar a libertad el circuito trabajado. La protoboard consiste en un tablero con orificios que se encuentran conectados eléctricamente entre sí en forma de líneas. Se debe mostrar una protoboard real y la forma de conectarse que tiene por dentro.

NOTA: Es de suma importancia el aprendizaje de cómo se usa o funciona la Protoboard, tratar de dar una explicación clara e interactuar con la clase en busca de dudas que puedan haber surgido.



(a) Protoboard.



(b) Conexión interna protoboard.

Arduino (3 minutos)

Los microcontroladores (μC) son circuitos integrados que poseen una unidad de procesamiento central (CPU) y memoria, además de varios pines de entrada y salida que permiten interactuar con otras componentes electrónicas como distintos tipos de sensores, así como actuadores.

En particular este curso se centrará en el micro controlador Arduino, que tiene como principal característica la simplicidad en su implementación debido al lenguaje de programación propio que posee.

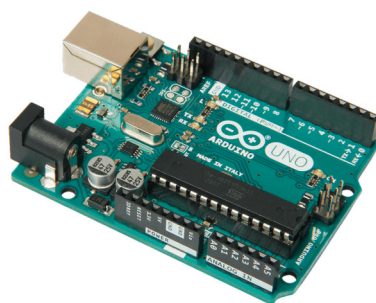


Figura B.2: Arduino UNO.

Se debe mostrar cómo buscar y colocar el Arduino en el proyecto creado. Presentar la interacción del microprocesador con el exterior mediante los pines digitales (0 y 1), analógicos (0 a 1023), y los pines de energía (Tierra y Vcc), procurar no extenderse más de lo necesario, es solo una presentación de Arduino.

Se debe ser muy cuidadoso de que cada palabra técnica que se utilice, sea presentado también su sinónimo o definición entendible para los y las estudiantes.

Descanso de 15 minutos

Circuito II

Códigos de Bloques para Arduino (3 minutos)

La plataforma Tinkercad/Circuitos dispone de bloques de salida, entrada, control, matemáticas, notación y variables con los que puede ser programado el microcontrolador Arduino.

Como ya se ha visto en las sesiones pasadas existen distintos tipos de bloques con diferentes colores, para el caso de los circuitos con Arduino son los siguientes:

- Salida en Azul.
- Control en Naranja.
- Entrada en Morado.
- Matemáticas en Verde.
- Notación en Gris.
- Variables en Rosado.

Bloques de salida (3 minutos)

Se procede a explicar de forma didáctica la función que cumple una salida en Arduino (Ej: Si imaginamos nuestro cerebro como Arduino, todas las ordenes que damos o acciones que llevamos a cabo son salidas y las entradas sería cualquier estímulo externo, como ver algo, oír algo, tocar algo).

Los bloques de salida son bloques que le dan órdenes a todos los sensores y/o actuadores conectados como salida en el Arduino (deben ser definidos de esta forma en el código), a partir de estos se pueden manipular los pines digitales según sea necesario ya sea para LED's, servomotores, altavoces y monitor serie.

Bloques de control (3 minutos)

Se procede a explicar de forma didáctica la función que cumple un bloque de control en Arduino con el cual es posible definir el tiempo en que dure cada instrucción, la cantidad de veces que se ejecuten, en qué condiciones ocurrirán determinadas acciones, entre otras cosas. Corresponde al análogo de las condicionales if, while y for en el lenguaje de Arduino.

Definir de forma breve qué es un if, while y for, un contexto y ejemplo, en la vida cotidiana, aún no ahondar en el código.

Actividad N°3: semáforo (15 minutos)

En esta actividad los y las estudiantes deben simular un semáforo utilizando LED's y un Arduino, donde se irán generando las transiciones entre el encendido de los LED's verde, amarillo y rojo con tiempos similares a los comúnmente utilizados. Se recomienda establecer tiempos fijos para ayudar a unificar resultados y mostrar ejemplo de resultado final.

Se debe ejecutar de forma individual.

Enlace del circuito a simular:

<https://www.tinkercad.com/things/231mt15c9Lc>

Se debe explicar la actividad y compartir el enlace del ejemplo para realizar el trabajo más expedito.

Previo a esta actividad se debe mostrar el circuito para el correcto encendido de un led, con código incluido y aclarar la mayor cantidad de dudas posibles.

Es muy importante que esta actividad cumpla a cabalidad los objetivos ya que si no, será casi imposible continuar con el desafío siguiente. Se debe instar a la clase a aclarar dudas y generar nuevas preguntas para ser respondidas, para esto el apoyo del staff será importante.

Desafío: esquina de semáforos (15 minutos)

Tomando en cuenta que los integrantes de la clase ya debiesen manejarse bien con los bloques de código y dado a que ya programaron su primer semáforo, ahora se les propone la actividad de programar los semáforos de una esquina como desafío. Para ello, al igual que en la actividad anterior deberán realizar el diagrama de flujo del problema, pero antes, deberán definir la cantidad de semáforos necesarios o que piensan utilizar tomando en cuenta los semáforos peatonales.

Enlace de ejemplo de circuito:

<https://www.tinkercad.com/things/aELk4keT80m>

La metodología será la siguiente:

Serán divididos en parejas, pero cada integrante realizará su proyecto de forma individual, debido a que los cambios efectuados en el programa no son de rápida visualización para un segundo integrante.

Si bien trabajarán de forma individual, la idea es que resuelvan dudas o consultas entre sí. Recordarles que si ambos no logran solucionar las dudas que surgen, pueden solicitar ayuda al tutor o tutora disponible.

Al finalizar deben presentar a la clase el circuito del compañero de trabajo, explicando las diferencias y semejanzas entre ambos, por lo que se debe contemplar dentro de los plazos asignados al desafío tiempo para familiarizarse mutuamente sobre el trabajo de su par.

Para evaluar la actividad, utilizar pauta de evaluación de la Sesión N°3.

NOTA: Se debe comentar la utilidad de reciclar o reutilizar líneas de código, bloques en este caso, pudiendo ser utilizadas para extender las funciones del código o lograr aplicaciones a mayor escala, como la programación de 2 o 3 esquinas de semáforo.

Cierre (5 minutos)

- Recordar las actividades que ejecutaron. Solicitar a la clase que señalen qué aprendieron, qué les gustó más y qué les resultó más complicado. Se deben considerar estos comentarios para las próximas sesiones.
- Verificar asistencia nuevamente.
- Despedirse y recordar horario de próxima reunión.
- Informaciones varias.



Academias de Tecnología Online

Sesión N° 4

7 de julio de 2023

Sesión N°4: Programación de Arduino mediante código.

Objetivo principal: Aprender a describir estructuras fundamentales de algoritmos utilizando lenguaje de programación.

Objetivo secundario: Desarrollar el pensamiento computacional y ser capaces de implementar diferentes ideas en un código.

Muy importante:

Recordar a los y las estudiantes mantener el clima de respeto y las reglas de buena convivencia.

Iniciando (5 minutos)

Hacer un breve recordatorio de lo que se vio la sesión anterior. Explicar los objetivos e importancia de la clase de hoy. Chequear asistencia de presentes.

Introduciendo un lenguaje de código (5 minutos)

Trabajar con bloques de códigos es una buena herramienta para comenzar a aprender de código, sin embargo, si queremos entregar instrucciones más complejas a nuestro microcontrolador, los bloques no serán suficientes.

Se debe ser capaz de enseñar a los asistentes que cada uno de los bloques que hemos utilizado, de forma implícita son equivalentes a una o más líneas escritas de instrucciones para nuestro microcontrolador.

Se debe mostrar cómo sustituir los bloques de código por escritura de código puro, pero sin dejar de comentar que el lenguaje Arduino que se utilizará solo es uno de los lenguajes existentes, no el único. Se debe destacar que la forma de programar mas usada es de código escrito, no bloques, que los bloques suelen utilizarse para enseñar en las etapas tempranas a programar, por lo que ahora subirán de nivel.

Así como en el mundo existen muchos idiomas: español, inglés, italiano, etc. que utilizan los mismos símbolos (letras), sin hablarse ni leerse todos iguales, cada microcontrolador tiene un “idioma” en el cual debe ser programado para que éste entienda nuestras instrucciones, a este le llamaremos lenguaje de programación. Es por lo anterior que si deseamos que Arduino nos

entienda las instrucciones que le damos, debemos dárselas en su lenguaje de programación, que casualmente se llama Lenguaje Arduino.

Observación: Se les debe recalcar que programar no es de una dificultad inalcanzable, pero que les surgirán muchas dudas, y es lo correcto preguntar. Debido a que se utilizará este lenguaje a lo largo de las siguientes sesiones, es importante que no solo las actividades las realice bien, si no que aprenda de tal modo que pueda utilizar el conocimiento para solucionar otros desafíos.

Arquitectura de Arduino

void loop y void setup: *¿Para qué sirven y cómo utilizar estas funciones?* (15 minutos)

Al ejecutar la aplicación de Arduino, un nuevo sketch (ventana) en blanco se abre en el cual están escritas las dos funciones base de Arduino. Se debe compartirlas pantalla mostrándoles en forma real cómo se ve una ventana de programación de Arduino y cuales son las dos funciones que se muestran en ésta.

Por otro lado para mostrar la misma ventana que aparece en la aplicación Arduino en Tinkercad, se debe ir a la pestaña de *código* y seleccionar la alternativa de Texto.

En la ventana desplegada, ya sea en la aplicación Arduino o la plataforma que estamos utilizando, Tinkercad, aparecen dos funciones escritas:

- `void setup ()`
- `void loop ()`

Dentro de la función **void setup()** se escriben las instrucciones que queremos que se realicen una sola vez al prender (inicializar) el Arduino, sirve para inicializar variables, definir entradas y salidas en la comunicación entre el Arduino y el computador.

La función **void loop()** como lo dice su nombre, es un bucle de instrucciones, por ello en esta sección del código se escriben todas las instrucciones que se van a repetir una y otra vez hasta apagar el dispositivo.

Para que se entiendan mejor estas funciones, se recomienda utilizar el siguiente ejemplo: Imaginemos que nosotros somos un Arduino, en nuestro void setup() se define nuestro nacimiento, nombre, color de piel, color de pelo, básicamente nuestro ADN, por otro lado, en el Void Loop() se deben escribir todas las acciones que realizaremos a lo largo de la vida por ejemplo, despertar, vivir, dormir y estas acciones deberán repetirse todos los días hasta que se apague la vida (muerte).

Se les debe enseñar y recalcar el tener en consideración el uso de `//` para realizar comentarios en los códigos como un buen hábito al programar, para facilitar la lectura y comprensión de nuestro código por otros o incluso nosotros mismos. Mostrar un ejemplo simple de cómo usar `//`.

```
DigitalWrite(led,HIGH);// Se enciende el led
Delay(1000);//Se mantiene encendido durante 1 segundo
DigitalWrite(led, LOW); //Se apaga el led
Delay(1000);//Se mantiene apagado 1
```

Existe igualmente una forma de escribir comentarios más extensos o comentar múltiples líneas, dicha función puede ayudar en casos que voluntariamente se necesite hacer que no se ejecuten ciertas líneas de código: */*código que se quiere omitir*/*.

La idea es generar un código ordenado con la finalidad de facilitar la corrección de posibles fallas y que sus pares lo entiendan también. Se propone mostrar una tabla que contraste dos códigos, uno ordenado y otro desordenado.

Otras funciones útiles (10 minutos)

Pero antes: Entradas y salidas ¿cómo funcionan?:

Se propone hacer esta analogía para la mejor comprensión de lo que es entradas y salidas, antes de hablar de funciones de Arduino.

Supongamos somos un o una chef de un destacado restaurant de sopas de pokemon, en donde los platos estrella son consomé de Pikachu y crema de Bulbasaur. Para preparar cada una de estas sopas necesitamos ingredientes, y estos son traídos desde el mercado de pueblo paleta.

Como es un restaurante de prestigio, ninguno de los platos debe salir de la cocina sin tener todos sus ingredientes en perfecto estado.

Tomando ese ejemplo, asumiremos que todo lo que entre a la cocina, como los ingredientes, los llamaremos **entradas** y todos los platos listos que salgan serán **salidas**, esto ayudará en un futuro a la asignación de pines de Arduino, o para entender cómo trabaja una función, al igual que esta cocina, recibe ingredientes y entrega platos.

Además de lo anterior, no estaría de más hacer preguntas a la clase, del tipo: ¿Qué objetos tecnológicos tienen entradas y salidas? De los respondidos, ¿Cuáles serían entradas y cuáles salidas según su criterio?.

Entendiendo lo anterior se presentan las siguientes funciones:

- **pinMode(pin, mode):** Corresponde a la función que define si los pines se comportan como **entrada** o **salida**. En la casilla *pin* ingresamos el número del pin digital o análogo y en *mode* ingresamos INPUT y OUTPUT según sea el caso.
- **digitalRead(pin):** Corresponde a la función que lee el valor de un pin digital específico, ya sea HIGH o LOW, lo que entenderemos como ENCENDIDO o APAGADO. Estos estados son “leídos” como un 1 o un 0.
- **digitalWrite (pin, estado):** Mediante esta función accionaremos los pines según su modalidad. En caso de que el pin sea configurado como salida, al activarlo (HIGH) se establece un voltaje de 5V en el pin. En caso de ser configurado como entrada, estaremos activando (HIGH) o desactivando (LOW) las resistencias pull-up internas en el pin.

- **delay(valor)**: Corresponde a una función de **tiempo**, está programada en milisegundos, de esta manera 1000 es igual a 1 segundo. Se sugiere hacer un esquema donde cada función sea una cajita, la entrada sean los argumentos, y la salida el resultado de su lectura.

Ej: VALOR → **delay()** → Mantiene el estado actual del Arduino por VALOR en milisegundos

En la presentación sería bueno un esquema gráfico que sintetice esta información.

Cabe destacar que en Arduino(y en los lenguajes de programación en general) las funciones y nombres de ellas son **case sensitive**, es decir, se distingue el uso de mayúsculas y minúsculas. De esta manera es correcto usar la función *pinMode(pin,mode)*, no así *pinmode(pin,mode)*

Actividad: simulación de semáforo haciendo uso de código(15 minutos)

Por ahora se repetirá la actividad de la clase pasada pero agregando la dificultad de ahora programar en código, no bloques. Por ello se les solicita programar y montar el circuito de un semáforo basándose en el diagrama hecho en la sesión anterior.

El código debe ser escrito en texto, iniciando las salidas del Arduino en el `void setup()` y el funcionamiento del semáforo en el `void loop()`.

Al final de la actividad y si es que aún no lo descubren los estudiantes, mostrar que en Tinkercad es posible pasar el código de bloques a código de texto.

Sería interesante mostrar a través de un ejemplo las diferencias entre un código de bloques y un código de texto.

Descanso de 15 minutos

Funciones y tipos de variables de interés

Comunicación Serial (5 minutos)

Es de interés tener alguna herramienta que permita monitorear desde el computador la información que procesa Arduino en su interior. Para ello se puede establecer una *conversación* con Arduino a través del cable mientras trabaja. Sin embargo, para que una comunicación sea efectiva, es necesario establecer una *forma* en la que comuniquen nuestros dispositivos. Existen muchos tipos de comunicación, pero en este caso se utilizará la comunicación serial. Para utilizar el monitorio serial primero es necesario activarlo, para ello se utilizará la sentencia:

```
Serial.begin(9600) ;
```

¿Dónde debe ir dicho comando? ¿void setup()/ void loop()/ Afuera? ¿Por qué tiene/qué significa el número 9600?

La información que recibe Arduino desde el PC se puede observar en la parte inferior del código de Arduino en *Monitor Serie*: **¿Qué se observa en el monitor serie?**

Cada vez que se quiera enviar información al monitor serie se debe utilizar las funciones:

- `Serial.print(...);`
- `Serial.println(...);`

Ambas funciones enviarán variables o palabras desde Arduino hacia el monitor del computador: **¿Cuál es entonces la diferencia entre ambas?**

La principal diferencia es que `Serial.print()` imprime los valores de lectura en el Monitor Serial de forma continua hacia la derecha uno junto a otro. Mientras el `Serial.println()` agrega al final de la lectura un salto de línea, escribirá el siguiente valor de lectura en una nueva línea, por esto es el **ln**.

Un buen hábito es utilizar `Serial.print()` o `Serial.println()` para hacer un seguimiento a las variables dentro de un código y ver que todo está bien (o mal) o para tomar mediciones de distinta índole.

Mostrar ejemplos de cuando las cosas van bien o mal.

Puede utilizarse un potenciómetro, o una asignación errónea de pines de salida, de tal forma que no lea lo deseado en el pin deseado.

Variables (5 minutos)

Se pueden crear distintos tipos de variables, entre ellas están (Se debe explicar el concepto de variable, desde lo cotidiano a lo matemático e informático):

- **char**: se utiliza para almacenar un carácter (explicar qué es un carácter).
- **int**: será uno de los más usados, se utiliza para variables que tendrán números enteros (recordar cuáles son los números enteros).
- **float**: se utiliza para variables donde se trabajará con números decimales.

De forma breve, explicarle a la clase la razón de especificar el tipo de variable al código. Ya que esto está relacionado con la forma en que lo almacena Arduino y el espacio que determina para guardarlo.

En el caso de definir variables en la parte superior del código, es decir, antes de la función `void setup()`, se estará definiendo una variable global lo que quiere decir que será visible para todas las funciones, en cambio, al definir variables dentro de las funciones `void setup()` o `void loop()` se definirán como locales, es decir, será visible solo dentro de cada función.

Se debe mostrar un ejemplo de la situación anterior, incluyendo que el nombre de la variable se reserva para todo el código si es global, y si no, se puede volver a utilizar con otro valor.

Desafío: mecanismo de emergencia (30 minutos)

Se requiere como objetivo principal, diseñar un mecanismo de emergencia manual para tu semáforo de la primera parte.

Para ello se deberá crear un nuevo código que permita utilizar **señales externas** (Interruptores) para encender los LED's manualmente. Se dispone de las variables VERDE, AMARILLO y ROJO que deberán almacenar la información proveniente del *exterior* y asignarse como estado a cada LED.

Como objetivo adicional se propone implementar un sistema de seguimiento que permita observar el valor de las variables en el monitor serial.

Se deberá presentar el circuito llamado *Pre Desafío* a la clase a modo de ejemplo de lo que deberán implementar. En el código están presentes comentarios para facilitar la explicación.

Para el desarrollo de la actividad se les adjunta en la guía enlace al *Pre Desafío* y al *Circuito Desafío* sin código.

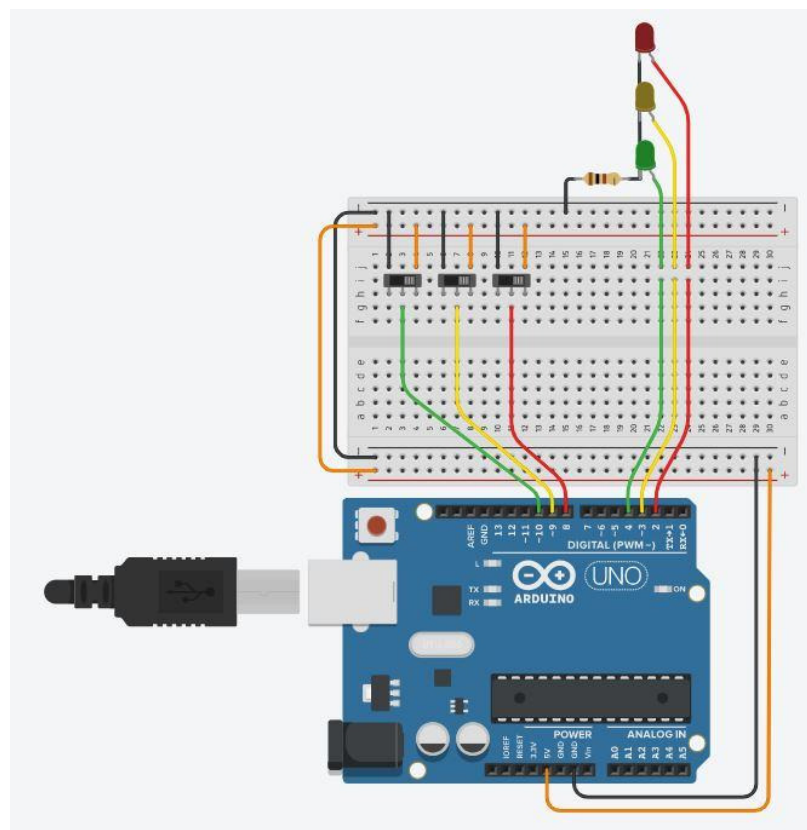


Figura B.3: Circuito Desafío Sesión N°4

- Pre Desafío: <https://www.tinkercad.com/things/7dRjntRvKtS>
- Desafío sin código: <https://www.tinkercad.com/things/0Ipjea3jh3T>
- Desafío con código: <https://www.tinkercad.com/things/1YTJmSzaNXz>

Al final de la clase presentar Desafío con código y explicar brevemente los resultados esperados.

Cierre (10 minutos)

- Recordar las actividades que ejecutaron. Solicitar a los estudiantes que señalen qué aprendieron, qué les gustó más y qué les resultó más complicado.
- El tutor, debe considerar estos comentarios para las próximas sesiones.
- Verificar asistencia nuevamente.
- Despedirse y recordar horario de próxima reunión.



Academias de Tecnología Online

Sesión N° 5

7 de julio de 2023

Sesión N°5: Estructuras condicionales en programación de Arduino.

Objetivo principal: Comprender la importancia y utilidad de las sentencias condicionales y bucles en la programación.

Objetivo secundario: Reconocer la aplicación de la programación en ejemplos de la vida cotidiana.

Muy importante:

Recordar a los y las estudiantes mantener el clima de respeto y las reglas de buena convivencia.

Iniciando (5 minutos)

Hacer un breve recordatorio de lo que se vio la sesión anterior.

Explicar los objetivos e importancia de la clase de hoy dejando claro que el enfoque de esta clase debe situarse en definir una forma de pensar las cosas más que en aprender sintaxis y códigos.

Los estudiantes deben ser capaces de asimilar situaciones de la vida cotidiana con las estructuras cíclicas y condicionales en programación.

El equipo debe entregar los circuitos a trabajar lo más completos posibles, de manera que el o la estudiante solo deba rellenar con la información particular y precisa los desafíos propuesto.

En base a lo anterior, es que un tutor o tutora auxiliar debe crear cuentas particulares (solo uso de esta sesión) que contengan las réplicas de los circuitos. Como alternativa, se puede proponer a la clase ingresar a un determinado proyecto de la cuenta ACATEC y copiar ellos mismos el modelo. Se sugiere escoger una sola opción para dirigir mejor el trabajo entre los diversos grupos y tutores.

El equipo debe apoyarse en material como diapositivas para dinamizar el traspaso de conocimientos a los alumnos. La estructura del material de apoyo recae en criterio personal de cada equipo de tutores asignado, pero debe existir.

- **Actividad N°1:** Trabajo individual con separación previa por Zoom en n grupos según n tutores o tutoras disponibles. Se deberá explicar la actividad, designar regiones de la pantalla a rayar y realizar un feedback a la clase.

- **Actividad N°2:** Trabajo individual con separación previa por Zoom en n grupos según n tutores o tutoras disponibles. Se deberá supervisar el trabajo de cada uno y resolver dudas.
- **Actividad N°3:** Trabajo colaborativo en sala principal de Zoom, llamar por nombre a los estudiantes presentes para que intenten responder a las preguntas y designen a un par para seguir respondiendo y así dinamizar la clase.
- **Desafío:** Trabajo en parejas separando previamente a cada par en una sala de Zoom.

Chequear asistencia de presentes.

Valores de verdad y comparadores (10 minutos)

Definiremos con valores de verdad a las posibles respuestas acerca de la existencia de un suceso, situación, algo observable de alguna forma. Un ejemplo sería: ¿Has desayunado? Si tengo conocimiento sobre esto, puedo responder si o no, no hay respuestas intermedias, o incluso si ésta fuera una afirmación, podría ser verdadera o falsa. Así cada suceso conocido por nosotros o que este dentro de nuestras posibilidades en el momento, podemos convertirlo en una afirmación o negación con un valor de verdad, o será **VERDADERO** o será **FALSO**.

Apoyándonos en lo anterior, podemos asegurar que los valores de verdad solo son dos, los cuales usualmente designamos como VERDADERO y FALSO, que a veces son representados por pares como (1,0), en donde 1 corresponde a verdadero y 0 a falso.

True(1)

False(0)

En programación estos valores de verdad se dan por lo general como resultado de una comparación de este tipo:

$x == y$ ¿es x igual a y?

$x \geq y$ ¿es x mayor o igual a y?

$x > y$ ¿es x mayor a y?

$x \leq y$ ¿es x menor o igual a y?

$x < y$ ¿es x menor a y?

$x != y$ ¿es x distinto de y?

Operadores Lógicos (10 minutos)

En relación con lo anterior, a veces es necesario que se cumpla mas de una condición al mismo tiempo para que ocurra un suceso, o bien, de que 2 condiciones, puedan llevar al mismo resultado, y pueden ocurrir de forma separada. A esto se le llama una operación lógica, a continuación se mostrará la forma en la que se representa en Arduino:

&& Conjunción lógica, representa el “y”.

|| Disyunción lógica, representa el “o”.

Dichas expresiones, significan, en el caso de la primera línea, que entregará un valor de *true* (verdad), si se cumplen todas las condiciones al mismo tiempo en la sentencia. Esto se puede leer como: Se cumple *condición 1 Y condición 2*.

Análogamente, en la segunda línea, que entregará un valor de *true*, si se cumple alguna de las condiciones en la sentencia o bien las dos. Esto se puede leer como: Se cumple *condición 1 O condición 2*.

Ya que no es un conocimiento que se suele materializar comúnmente, es probable que sea lento el proceso de comprensión de la lógica, por lo que se recomienda dar ejemplos de estas situaciones, mientras más claros y simples mejor, que se acerquen a sus vidas cotidianas.

Condicionales If y else (10 minutos)

Las estructuras condicionales nos entregan mayor versatilidad a la hora de programar, en este caso la condicional If nos permite definir acciones en caso que se cumpla la condición. Por ejemplo, en caso de que hoy haya sol, saldré a caminar:

```
If ( soleado ) {  
  saldre_a_caminar();  
}
```

Luego, podemos complementar la función utilizando la expresión **else** que nos permitirá realizar una acción en caso que la afirmación sea falsa (que no haya sol), es decir, si esta soleado saldré a caminar, sino me quedaré en casa:

```
If ( soleado ) {  
  saldre_a_caminar();  
}  
else {  
  quedarme_en_casa();  
}
```

Actividad N°1: ¡Alto ahí!(20 minutos)

Últimamente los robos a una tienda de salchichas se han visto aumentados y resulta imposible detectar a los salchi-ladrones debido a la falta de luz, es por ello que se te ha contratado para programar un sistema de Salchi-seguridad, el cual esta compuesto por un foco (LED) el cual se encenderá cada vez que alguien se acerque a la salchi-tienda, para ello un técnico experto en robo de salchichas a hecho el conexionado de un LED junto a un sensor de movimiento PIR (Sensor Infrarrojo Pasivo), solo falta tu parte del salchi-trabajo, programarlo ¡Tú puedes!

El siguiente enlace contiene el circuito a programar:

<https://www.tinkercad.com/things/5LtmkQiUnXS-sensor-pir-con-led>

El siguiente código sería una posible solución. Notar que el uso del PIR es muy similar al comportamiento de los botones.

```
//Declaramos pines
int sensorpir = 12;
int led = 13;

void setup()
{
    //Asignamos las entradas y salidas
    pinMode(sensorpir,INPUT);
    pinMode(led,OUTPUT);
    Serial.begin(9600);
    digitalWrite(led,LOW);
}

void loop()
{
    digitalWrite(led,LOW);
    Serial.println(sensorpir);//El valor del sensor
    if(digitalRead(sensorpir) == HIGH)
    {
        //Cuando el sensor detecte algún movimiento entrara al if
        //Dentro del if lo que pasará es que el sensor pir detectará movimiento y el led parpadeará
        //hasta que deje de detectar movimiento
        Serial.println("Detectado movimiento por el sensor pir");
        digitalWrite(led,HIGH);
        delay(1000);
        digitalWrite(led,LOW);
    }
}
```

Descanso de 15 minutos

Condicional Switch-case (15 minutos)

Otro tipo de condicional presente en la programación de Arduino es el **switch-case**, este, a igual que los condicionales **if/else**, permiten controlar la ejecución de una serie de tareas a ejecutar, dependientes de una o varias condiciones que se debe cumplir.

El **switch-case**, compara el valor de una variable, con una serie de valores específicos, los cuales son declarados en la sentencia **case**, cuando el valor de la variable coincide con el caso a analizar, se ejecutará la acción declarada para ese caso.

Una consideración que se debe tomar en cuenta en la programación de los **switch-case** es el uso de la función **break**, que permite detener la ejecución de un código y que en este caso se

utilizará para finalizar la ejecución de la tarea asociada a la sentencia **case**, y por otro lado, la sentencia **default**, que define una tarea a ejecutar, en caso de que la variable no cumpla ninguno de los casos planteados en las sentencias **case**.

Con estas consideraciones, se plantea el código general de programación de un **switch-case** como se observa a continuación, donde nuestra variable a analizar esta determinada por *var* y los casos posibles son *label1*, *label2* y **default**.

```
switch(var){
    case label1:
        //tarea2 a ejecutar
        break ;

    case label2:
        //tarea 2 a ejecutar
        break ;

    default:
        //tarea a ejecutar por defecto
        break ;
}
```

Se sugiere complementar la explicación con un ejemplo como el siguiente o más completo:

```
switch(luz_semaforo_auto){
    case VERDE:
        //Si la luz del semáforo de autos es verde se enciende la luz de peatones roja
        digitalWrite(luz_roja_peaton, HIGH) ;
    case ROJA:
        //Si la luz del semáforo de autos es roja se enciende la luz de peatones verde
        digitalWrite(luz_verde_peaton, HIGH) ;
    default:
        //tarea a ejecutar por defecto, salir del case
        break ;
}
```

Desafío N°1: el menú (25 minutos)

Como una forma de organizar los pedidos de comida que llegan a la cocina del restaurante ACATEC, un grupo de chefs del lugar propone diseñar un sistema de notificación de colores, según el platillo solicitado, para gestionar de mejor manera los platillos a preparar, es por esto que plantearon un circuito utilizando Arduino, en el cual se le asigna un color específico de un LED a cada plato de comida preparado en el restaurante.

Por problemas de gestión, el grupo a cargo de la implementación del sistema solo logro diseñar el circuito pero no lograron generar la programación necesaria para implementar el sistema y solicitan ayuda para finalizarlo. Es por esto que se deberá probar el sistema de alarmas.

Utilizando lo aprendido en clases sobre la estructura condicional switch-case y considerando el circuito descrito anteriormente, se debe programar las partes faltantes del código a manera de que se vayan cambiando las notificaciones de cada platillo cada 1 segundo.

A medida que va aumentando el tiempo, se debe encender la luz correspondiente a cada segundo y apagar las anteriores. Además, se deberá ver en el monitor serial que platillo esta siendo pedido. Para esto se contará con un código base de ejemplo.

Dar a entender por qué es práctico usar un for(un ciclo) en estos casos donde queremos repetir una tarea. Para ser mas prácticos, se utilizará la función for(i = 0; i <5; i++), la cual se verá en clases posteriores. Solo bastará con indicar qué es lo que hace toda esa función: que cada un segundo el valor de i va cambiando(va desde 0 hasta 4, y se reinicia cada vez después de llegar a 4), y bajo esta variable los platillos deben ir cambiando.

Se debe evitar profundizar en una explicación de la función for para no confundir a la clase. Pueden reducirlo a: **cada 1 segundo va cambiando el valor de i ascendentemente y luego se reinicia al llegar a 4 para empezar nuevamente con 0.**

Se sugiere cambiar los platillos A, B, C, D y E por platillos con nombres reales, como por ejemplo: arroz con hamburguesa, etc. Podría ayudar a la hora de la memorización.

```
int rojo=13; //platillo A
int blanco=10; //platillo B
int azul=7; //platillo C
int naranja=4; //platillo D
int verde=2; //platillo E

void setup()
{
  Serial.begin(9600);

  pinMode(rojo, OUTPUT);

  //.....//
  //complete código aquí//
  //.....//
}

void loop()
{
  int i;
  for(i = 0; i<5; i++){
    switch(i){
      case 0:
        digitalWrite(rojo, HIGH);
        digitalWrite(blanco, LOW);
        digitalWrite(azul, LOW);
        digitalWrite(naranja, LOW);
```

```

        digitalWrite(verde, LOW);
        break;
        //.....//
//complete código aquí//
//.....//
    }
delay(1000);
}
}

```

El siguiente enlace contiene el circuito con el que se trabajará y el código a completar:

<https://www.tinkercad.com/things/4pzSPsk21i0-funky-fulffy-kasi/>

Cierre (10 minutos)

- Recordar las actividades que se ejecutaron. Solicitar a los estudiantes que señalen qué aprendieron, qué les gustó más y qué les resultó más complicado.
- El tutoro tutora, debe considerar estos comentarios para las próximas sesiones.
- Verificar asistencia nuevamente.
- Despedirse y recordar horario de próxima reunión.



Academias de Tecnología Online

Sesión N° 6

7 de julio de 2023

Sesión N°6: Conceptos de electrónica básica y componentes de uso común.

Objetivo principal: Reconocer conceptos de electrónica básica, incluyendo señales análogas y digitales.

Objetivo secundario: Introducir componentes como medios de comunicación con el exterior.

Muy importante:

Recordar a los y las estudiantes mantener el clima de respeto y las reglas de buena convivencia.

Iniciando (5 minutos)

Hacer un breve recordatorio de lo que se vio la sesión anterior.

Se debe explicar que la intención de esta clase es conocer conceptualmente los distintos elementos y componentes con los cuales ya se ha trabajado.

El equipo debe apoyarse en material como diapositivas para dinamizar el traspaso de conocimientos a la clase. La estructura del material de apoyo recae en criterio personal de cada equipo de tutores asignado, pero debe existir e ir de acuerdo a los objetivos de la sesión.

Chequear asistencia de presentes.

Voltaje y corriente (10 minutos)

Introducción

Todo elemento electrónico requiere de energía (Se recomienda hacer una analogía simple de qué es la energía) para funcionar, específicamente necesita de la energía eléctrica (dar ejemplos de qué utiliza energía eléctrica u otros tipos de energía) por lo tanto es de nuestro interés analizar y comprender el comportamiento de la electricidad con el fin de manipularla de forma controlada.

¿Qué es la electricidad?

Una forma tal vez más simple para entender la electricidad es verla como el agua que corre por una manguera o cañería de una casa. Se identifican 3 partes: la fuerza que empuja el agua (Voltaje), las imperfecciones o lugares más angostos en la manguera que impiden el paso del agua (Resistencia) y finalmente se tiene el caudal (éste es definido como cantidad de fluido que pasa por la cañería en cierta cantidad de tiempo, podría medirse como $[m^3/s]$) de agua que logra salir de la manguera (Corriente). Estas 3 variables son **dependientes** (Voltaje – Resistencia - Corriente) en sí y son las bases de la electrónica.

Nota: Se recuerda que las variables son elementos, datos que normalmente son expresados a través de símbolos (matemática) que son libres de tomar cualquier valor dentro del conjunto de números donde están definidos. Un ejemplo podría ser: la cantidad de veces que estornudamos en un día o la cantidad de veces que ladra un perro en un día, etc.

Si este último fuera el caso la variable sería **Ladridos de Lomito en un día** y ésta todos los días cambiaría su valor, o sea variaría (variable). Ahora, ¿y si Lomito solo ladra cuando ve pasar el camión de Lipigas por afuera de la casa? Podríamos decir que: el número de veces que Lomito ladre estará relacionado con la cantidad de **veces que pase el camión del gas afuera de la casa en un día** (nueva variable), por lo que podríamos concluir que SON VARIABLES DEPENDIENTES, ya que dependerá de la cantidad de veces que pase el camión de Lipigas la cantidad de veces que ladre Lomito en un día.

En contraste, si aparece la variable **veces que pasa el camión del gas por una calle lejana a la casa de Lomito**, Lomito no sabrá de este suceso por lo que no ladrará y la variable **Ladridos de Lomito en un día** NO estará relacionada a dicha variable, por lo que se concluye que ambas son VARIABLES INDEPENDIENTES una de otra.

Relaciones entre las 3 variables **Voltaje - Resistencia - Corriente**:

- Mientras mayor sea la variable voltaje, mayor es la (variable) corriente, la cual fluirá a través del cable o alambre.
- Mientras mayor sea la resistencia (variable), menor es la corriente que fluirá por el cable o alambre.
- Si queremos que fluya o circule menos corriente, podemos: aumentar la resistencia, reducir el voltaje o ambos juntos.



Figura B.4: Analogía relación voltaje- corriente- resistencia.

Ya en la práctica, para que exista una corriente, debe existir una fuente de voltaje (pilas, red domiciliaria, batería, etc.) y un camino que guíe a la corriente (cables o alambres); dicho camino debe siempre ser cerrado para que pueda producirse dicho flujo.

Observación: El alambre es un único conductor eléctrico, mientras un cable se compone de muchos conductores (filamentos) recubiertos con aislantes.

Tierra → Voltaje positivo → componente → Tierra

¡OJO! si no hay un componente intermedio se produce un cortocircuito, una descarga eléctrica peligrosa.

¿Qué es un cortocircuito?

Imagina una pila como si fuera un sartén en el fuego, el mango de este la componente o resistencia y finalmente nuestra mano tocando el mango de ésta como Tierra. Si comenzamos a acortar cada vez más el mango del sartén, llegará un momento en el que nuestra mano quedará tan cercana al sartén que comenzaremos a quemarnos y finalmente cuando el mango desaparece tendremos que tomar el sartén directamente, sintiendo la totalidad del calor en nuestras manos.

Lo mismo sucede con la corriente, si no existe componente intermedia que ocasione resistencia a ésta, la energía se transmitirá su totalidad a tierra, ocasionando lo que es llamado un **corto circuito**.

Un corto circuito en una red domiciliaria podría fácilmente ocasionar una explosión y luego un incendio. Si bien los aparatos electrónicos en su interior no suelen tener altas corrientes, como para dañar a una persona, si se produce un corto circuito podría fácilmente quemarse nuestro microcontrolador, como Arduino, u otras componentes de elevado valor.

Observación: Para graficar de mejor forma el cortocircuito, pueden apoyarse de algún video donde hagan cortocircuito controlado con fines educativos. Además, se podría armar una cápsula breve mostrando un corto circuito. Es importante que destaquen lo peligros de un cortocircuito, por seguridad de los y las estudiantes.

Es muy importante recalcar el fin de **cerrar el circuito** o como se suele decir también, cerrar el camino de la corriente. Se sabe que la corriente podemos asociarla a agua fluyendo por una manguera (cable) esta fluye desde el polo positivo al negativo (o tierra) de la batería u otro objeto que cumpla el fin de energizar, si esta manguera en algún punto es cortada, el flujo de agua ya no llegará a su destino y es lo que llamaremos **circuito abierto**.

Nota: Se deben mostrar al menos dos ejemplos de de corto circuitos, estos deben incluir solo componentes conocidos por los y las estudiantes.

¿Qué es un circuito abierto?

En cada circuito que se arme, nuestro objetivo es que la corriente siga el camino previsto, fluyendo desde el polo positivo al negativo sin ser interrumpida en ningún punto intermedio.

En una carrera de relevos se busca algo similar al hacer correr entre los participantes el testigo (barra) completando un camino llegando a la meta propuesta.

Imaginar que el testigo es la corriente, y el circuito que recorrerá solo se podrá energizar si se termina esta carrera de relevos, si en cualquier lugar de la carrera el testigo no se transfiere, entonces el circuito no tendrá energía y no funcionará, por lo tanto, se debe ser capaz de seguir con la vista este camino hasta llegar a tierra (o polo negativo), solo de esta forma se habrá energizado el circuito deseado.

En el caso de que no se pueda terminar el camino visualmente, podremos hablar de un circuito abierto, por lo que no estará energizado.

Nota: Al igual que en la sección de corto circuito, en esta se deben mostrar al menos dos ejemplos de circuitos abiertos, estos deben incluir solo componentes conocidos por los y las estudiantes.



Figura B.5: Analogía circuito abierto.

Actividad N°1: *¡el camino sí importa!* (10 minutos)

El tutor o tutora planteará como mínimo 4 configuraciones de circuitos en donde se deberá poder reconocer si el circuito presenta en algún lugar un **corto circuito** o **circuito abierto**, se debe marcar el lugar puntual dentro del circuito en donde ocurre, además de indicar qué camino toma la corriente antes de llegar a este punto.

Nota: Se deja a libertad si se hacen más de 3 circuitos, lo ideal es que se entienda qué se observa desde fuera cuando una de estas dos situaciones sucede y que sepan reconocerlas luego de un análisis. Se recomienda hacer material como diapositivas para que puedan apreciarse mejor cada ejemplo, y utilizar componentes que se hayan utilizado para no confundirlos ya que al finalizar esta sesión se verá el conexionado de componentes más utilizados.

Descanso de 15 minutos

Señales analógicas y digitales (15 minutos)

Una señal eléctrica corresponde a una señal que nace de la interacción entre voltaje, corriente y resistencia.

Existen dos tipos de señales eléctricas, las digitales y las analógicas o análogas:

- **Señal Digital:** Es una señal no continua (para utilizar el término continuo se recomienda hacer un pequeño repaso de qué se considera una señal continua) que suele tomar solo dos valores posibles.

Particularmente en Arduino una señal digital toma valores de 0 y 5[V], voltajes que se pueden interpretar como un 0 o un 1, como un falso o un verdadero, como un apagado o un encendido respectivamente. Es importante no confundir una señal binaria con un digital, una señal binaria ES digital, en cambio una señal digital no es necesariamente binaria, pero sí tendrá dos valores posibles.

- **Señal Analógica:** Es una señal continua que tiene la libertad de tomar cualquier valor dentro de algún rango específico. Particularmente en Arduino, si nos encontramos midiendo voltaje, este rango suele ser entre 0 y 5[V], por lo que la señal puede tomar valores intermedios (ejemplo: 3[V], 4.21[V], etc).

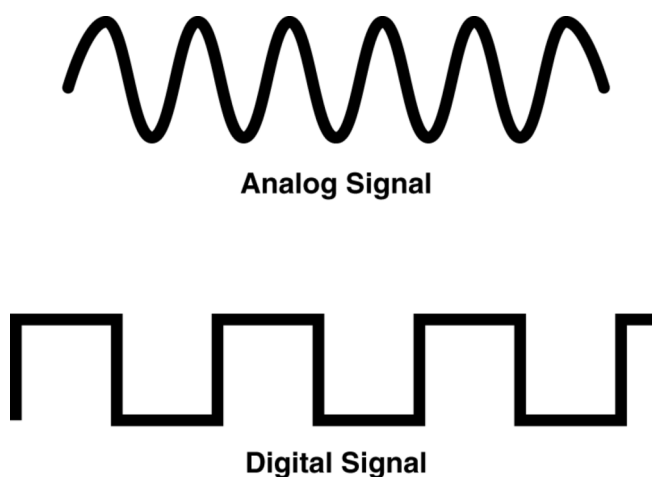


Figura B.6: Ejemplo señal análoga y digital.

Actividad N°2: *nuestro mundo digital y análogo.* (7 minutos)

Identifica si las siguientes señales o variables son digitales o análogas:

1. La temperatura.
2. La Hora en tu celular.
3. La Información almacenada en un Pendrive.
4. Tu altura/peso.

5. La edad de una persona.
6. La distancia entre nosotros.
7. Eventos como la lluvia, un temblor, etc.(suceden o no suceden)
8. El interruptor de luz.

Se sugiere poner la mayor cantidad de ejemplos posibles dentro del tiempo destinado a esta actividad y que estos sean claros y precisos. Además, para aumentar la participación, sería ideal plantear estas preguntas a nivel clase y hacer votación sobre quienes creen que la señal es de un tipo u otro.

Solicitar a la clase dar sus propios ejemplos de señales digitales o análogas, esto servirá para confirmar el aprendizaje.

Conexionado de componentes

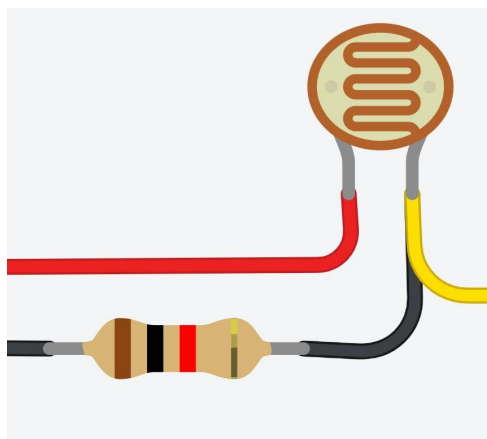
Dado que la base de la electrónica es guiar y controlar la cantidad de corriente que fluye por nuestros componentes, es de vital importancia saber cómo conectarlos correctamente.

Hasta ahora solo hemos seguido instrucciones o se han entregado estos ya conectados, por lo que hoy se formalizarán estos conocimientos.

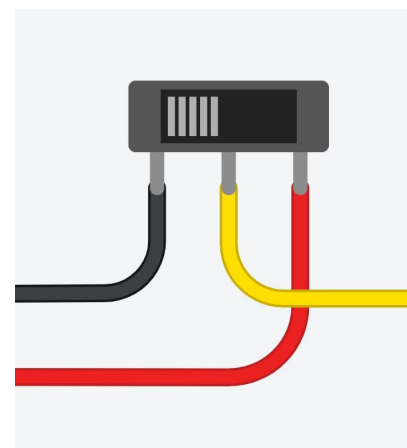
La mayoría de las conexiones (unión de dos elementos, en este caso: conductores) son de 2-3 terminales (lugares de los componentes destinados a la conexión), 2 esenciales para +5V y GND (recordar que GND es la abreviación ground, tierra en inglés) pertenecientes al camino principal y una tercera para una señal de información que será entregada (sensores) o recibida (leds, motores, etc) por el componente, algunas conexiones requieren de resistencias adicionales para evitar realizar cortocircuitos o regular corriente.

A continuación, algunas conexiones útiles, se utilizará por convención +5V, GND y Señal (recibida o enviada por el componente):

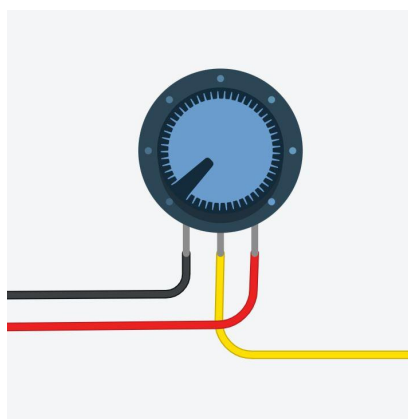
Nota: Se debe recalcar que no es necesario que recuerden de memoria las conexiones en las siguientes imágenes, ya que estas serán una guía futura si es que necesitan conectar las componentes que se muestran, por lo que no deben abrumarse si es que no lo comprenden al inicio.



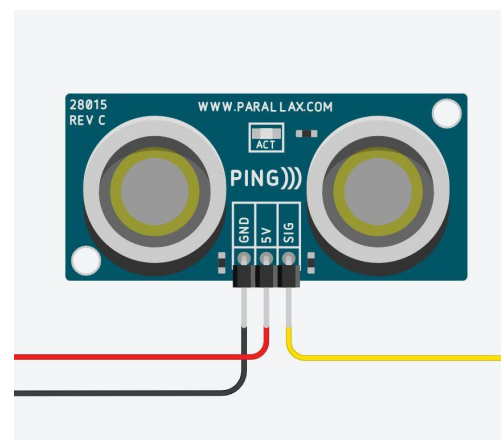
(a) LDR - La resistencia evita un cortocircuito y permite que en la señal se lea un valor de voltaje entre +5V y GND según la cantidad de luz.



(b) Interruptor - Da el paso de corriente directa en un sentido u otro (no necesita resistencia) En la señal se lee +5V o GND según su posición (es similar al interruptor de una lámpara).

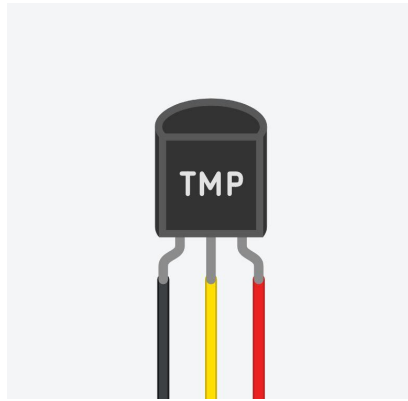


(c) Potenciómetro- Al ser ya una resistencia en sí, no se necesita otra adicional. En la señal se lee un valor entre +5V y GND según su posición.

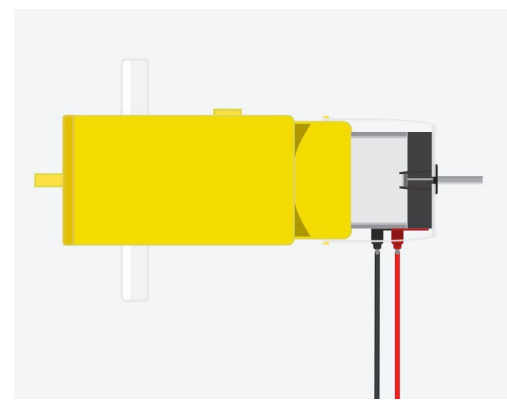


(d) Ultrasonido - Se usa para estimar la distancia a partir del valor de señal, se energiza con +5V y GND.

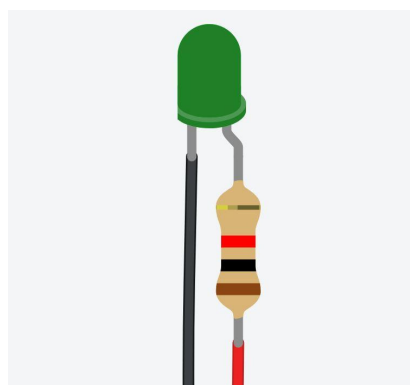
Se sugiere hacer una actividad en la cual se puedan hacer fichas de cada componente, detallando su funcionamiento con las palabras de los y las estudiantes y que agreguen uno que consideren interesante, que se encuentre dentro de Tinkercad y que investiguen su funcionamiento para la siguiente sesión.



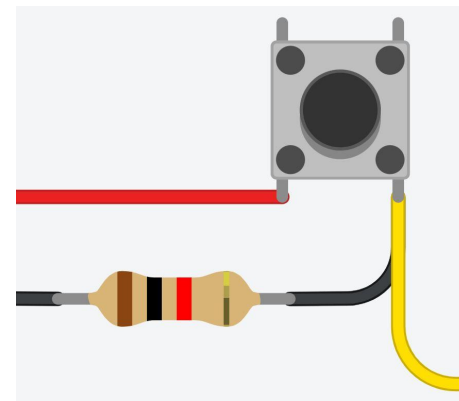
(e) Sensor temperatura - Se utiliza para estimar la temperatura a partir del valor de señal leída en este, se energiza con +5V y GND.



(f) Motor de corriente continua. Se conecta a +5V y GND para que gire en un sentido. Si se conecta al revés, el motor gira en sentido contrario.



(g) LED - Es necesario regular (disminuir) corriente con una resistencia.



(h) Botón - La resistencia permite que en la señal se lea +5V al presionarse y GND al soltarse (pull up o pull down).

Figura B.10: Conexión de componentes básicas para el curso.

Circuitos funcionales en Tinkercad

Se muestran a continuación algunos circuitos funcionales que ayudarán a entender a la clase cómo pueden condensar sus conocimientos básicos en circuitos más complejos.

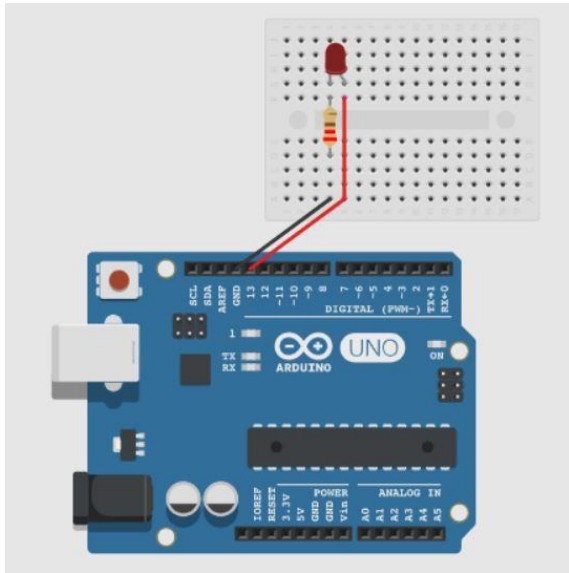


Figura B.11: Circuito básico led.

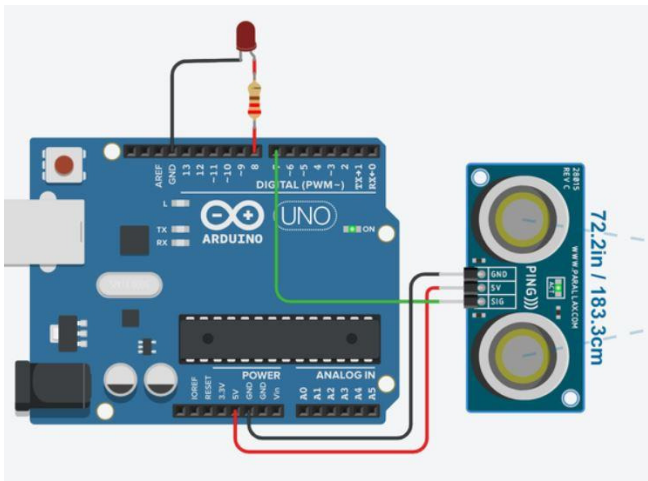


Figura B.12: Alarma de proximidad con led.

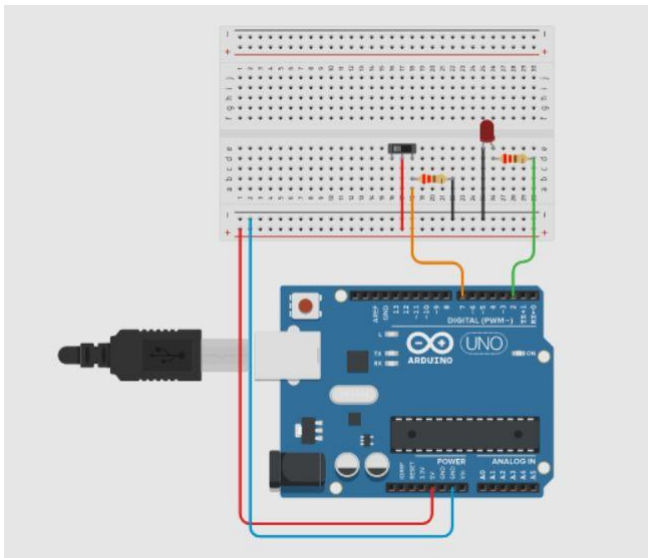


Figura B.13: Led con interruptor de encendido.

Cierre (10 minutos)

- Recordar las actividades que se ejecutaron. Solicitar a los estudiantes que señalen qué aprendieron, qué les gustó más y qué les resultó más complicado.
- El tutoro tutora, debe considerar estos comentarios para las próximas sesiones.
- Verificar asistencia nuevamente.
- Despedirse y recordar horario de próxima reunión.



Academias de Tecnología Online

Sesión N° 7

7 de julio de 2023

Sesión N°7: Funciones de Arduino: señales, sensores y actuadores.

Objetivo principal: Conocer y definir funciones básicas de Arduino que permitan trabajar con señales.

Objetivo secundario: Introducir componentes sensores y actuadores como medios de comunicación con el exterior.

Muy importante:

Recordar a los y las estudiantes mantener el clima de respeto y las reglas de buena convivencia.

Iniciando (10 minutos)

Hacer un breve recordatorio de lo que se vio la sesión anterior. Además, se recomienda preguntar a los asistentes a la clase si recuerdan los contenidos abordados la sesión anterior.

Se debe explicar que la intención de esta clase es conocer en detalle algunas de las funciones básicas para programar Arduino y hacer uso de estas de forma correcta.

El equipo debe apoyarse en material como diapositivas para dinamizar el traspaso de conocimientos a la clase. Tanto esta guía como el material de apoyo recae en el equipo desarrollador de estas.

Chequear asistencia de presentes.

Funciones para señales leídas por Arduino (10 minutos)

Si bien las siguientes funciones de las cuales se hablará esta sesión ya se han usado anteriormente, ahora se verán en profundidad, logrando aprender un poco más sobre ellas:

- **digitalRead(pin):** Permite leer el valor digital del pin especificado entre paréntesis, la salida de esta función puede tomar el valor 0 o 1, por lo que se considera una señal digital (solo puede tomar dos valores). Esta lectura se perderá si no es guardada en alguna variable, por lo que suele almacenarse de la siguiente forma:

```
int variable_digital_guardada=digitalRead(8);
```

La anterior línea almacena en la variable de tipo entero **variable_digital_guardada** el valor digital leído en el pin 8 de Arduino. Así durante el desarrollo del código se podrá recurrir al este valor solo escribiendo el nombre de la variable establecido.

Si le pidiéramos a un amigo que lea el número de teléfono de nosotros desde un papel, lo hará sin problemas, pero si 10 minutos después le pedimos repita este número de memoria no podrá, ya que solo fue lectura, no fue almacenado en su memoria, lo mismo sucede con esta función, es solo lectura en el instante en el que se ejecuta la línea de código. Por lo anterior es que se debe almacenar en una variable esta lectura, para poder conservarla, ejecutando el símil de la línea de código de ejemplo anterior.

- **digitalWrite(pin,Estado):** Permite escribir el estado especificado: HIGH (1) o LOW (0), como un valor digital (5[v] o 0[V] respectivamente), en el pin indicado.
- **analogRead(pin):** Esta función permite leer el valor de un pin analógico especificado en *pin*, para esto reconoce el rango de voltajes de 0 y 5 [V] y los transforma en una variable que toma valores entre 0 y 1023 (1024 niveles en que se puede encontrarla variable) que será la que arroje esta función al ejecutarse.
- **analogWrite(pin,Valor):** Esta función permite escribir en un pin análogo valores intermedios entre 0 y 5[V]. Los pines para este uso son restringidos en Arduino y sus números corresponden a 3, 5, 6, 9, 10, y 11 (pines acompañados por un ~). A la variable *Valor* se le pueden asignar valores entre 0 y 255, lo que permite generar 256 voltajes intermedios entre 0 y 5[V]. Véase las Figuras 1 y 2 para una mejor comprensión:

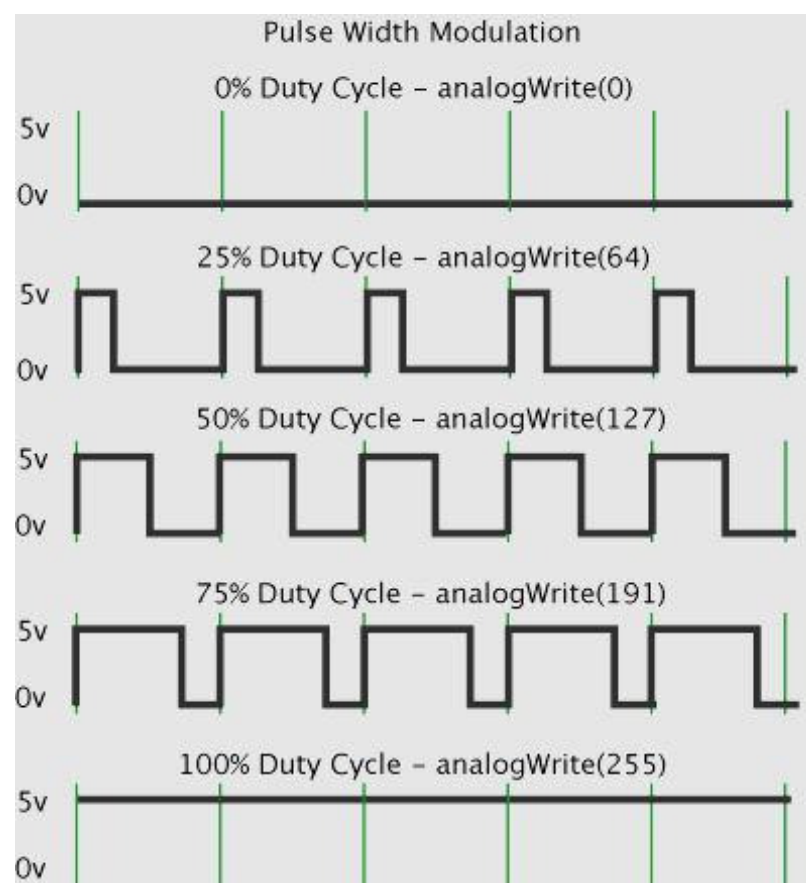


Figura B.14: Ancho de pulsos de la función `analogWrite(Valor)`, según variación de la variable `Valor`.

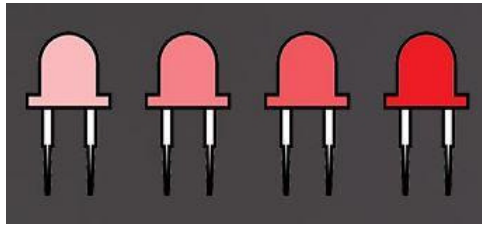


Figura B.15: Efecto del cambio en anchos de pulso en un led alimentado por estos.

- `pinMode(pin, mode)`: Corresponde a la función que define si los pines se comportarán como entradas o salidas. En la casilla *pin* ingresamos el número del pin digital o análogo y en *mode* ingresamos INPUT (entrada) y OUTPUT (salida) según sea el caso.

Actividad N°1: *y así se hizo la luz.* (20 minutos)

Uno de los aldeanos de la ciudad necesita implementar un sistema de iluminación para una nueva habitación de su casa que utilizará como estudio.

El estudio necesita dos tipos de iluminación diferentes:

- Una lámpara capaz de encenderse al tacto, es decir, la luz se mantiene encendida solo cuando el botón se mantiene presionado.
- Una lámpara que sea capaz de regular su intensidad según lo que necesite el usuario por medio de una perilla (potenciómetro).

Se deberá realizar el correcto conexionado (conexión de alambres) y programación del microcontrolador para el manejo de los componentes solicitados de acuerdo a los requisitos planteados

Ayuda: Para convertir valores entre 0-1023 a valores entre 0-255, una buena aproximación se logra con multiplicar el valor entre 0-1023 por 0.25, obteniendo el valor entre 0-255 equivalente. O se puede hacer uso de la función `map(valor)`, esta devuelve como resultado el mapeo del número deseado en un rango distinto al actual:

```
int valor=map(a,b,c,d,e);
```

En donde,

- a: Valor de entrada.
- b: Valor inicial del rango de entrada.
- c: Valor final del rango de entrada.
- d: Valor inicial del rango de salida.
- e: Valor final del rango de salida.

En este caso se utilizaría de la siguiente forma:

```
valor_convertido=map(valor_a_convertir,0,1023,0,255);
```

La anterior línea guardará el valor que necesitamos en `valor_convertido` para su utilización.

Link circuito base para la actividad:

<https://www.tinkercad.com/things/hcBFnztmUaD>

Descanso de 15 minutos

Sensores y Actuadores

¿Cómo conocer mis componentes? (7 minutos)

Si deseas saber las especificaciones de un modelo determinado de componente, se debe revisar su hoja de datos (se denomina en inglés: datasheet).

Llamaremos actuador a cualquier componente que recibe una señal eléctrica del sistema en este caso del microcontrolador (Arduino) y transforma está en energía de uso común, puede ser mecánica o energía eléctrica para alimentación.

Además, llamaremos sensor a cualquier componente que reciba una señal eléctrica de alimentación (sin información) y entregue una señal con información, datos, etc, hacia el sistema (Arduino).

Nota: Se recomienda mostrarles un Datasheet de algún componente que sea conocido por los y las estudiantes, recalcar la importancia de este en la aplicación y sus datos más importantes.

Actividad N°2: *reconocer actuadores y sensores* (10 minutos)

La siguiente actividad es de importancia que sea guiada por los tutores y tutoras, deben avanzar de forma conjunta y obtener feedback a medida que se van respondiendo las preguntas.

Tomando en cuenta lo aprendido, ahora deben reconocer si cada componente electrónica que se muestre en la siguiente lista es un actuador o sensor, y explicación de por qué considera lo es:

- Pantalla (display).
- LM35
- Botón.
- Bocina/parlante.
- Potenciometro.
- Led
- Fotorresistencia.
- FC-28.
- HC-SR04.
- PIR HC-SR501.

Ejemplos posibles:

Actuadores:

- Led (semáforo).
- Motor DC.
- Piezo Buzzer.
- Servomotores.

Sensores:

- Botones e interruptores.
- Sensor Temperatura.
- Sensor de Luz (LDR).
- Potenciómetro.

Componentes bidireccionales:

- Sensor ultrasonido.
- Sensor infrarrojo.
- Comunicaciones: UART, I2C, Bluetooth, Wifi, telefonía satelital, etc. . .

Nota: Si alguno de los anteriores componentes no es del completo conocimiento del tutor o tutora, son libres de reemplazarlo por uno que si manejen o simplemente encuentren más acorde a la lista. Además, incentivar a la clase a buscar las hojas de datos de los componentes que aparecen por modelo.

Desafío: *¡Con mi queso no!* (25 minutos)

El ratón Jean Paul, amigo francés de Condensín vino de intercambio a estudiar a las Academias de Tecnología y luego de asistir a todas las sesiones decidió hacer un súper proyecto: un sistema seguridad para su queso máspreciado, un queso francés con 10 años de maduración y hecho con leche de vacas rosadas y que pastan en los jardines de la Torre Eiffel.

El sistema de seguridad que él necesita debe cumplir las siguientes funciones:

- Debe encender un led cuando haya movimiento al rededor de su lujoso queso.
- Debe enviar un mensaje a través el monitor serial que diga: **¡Policía, se llevan mi queso!**, cada vez que Jean Paul apriete un pulsador.

Para estos requisitos el ratón Jean sabe más o menos qué necesita, así que en su kit para el proyecto tiene:

- Un sensor PIR

- Un led rojo
- Un pulsador
- Dos resistencias
- Una protoboard
- Un Arduino
- Y por su puesto un trozo de queso de colación, porque de seguro nos dará hambre trabajando.

Con toda la información anterior ayuda al ratón Jean Paul a hacer una alarma para su queso súper lujoso.

Es clave para este desafío la correcta conexión del pulsador y el sensor PIR, si estos no se conectan correctamente será muy difícil avanzar.

Una de las soluciones más simples es hacer dos if al mismo nivel, uno para el sensor PIR y el led y otro aparte para el pulsador y el monitor serial. Además, se debe tomar en cuenta el funcionamiento negado del pulsador para la comparación del if respectivo.

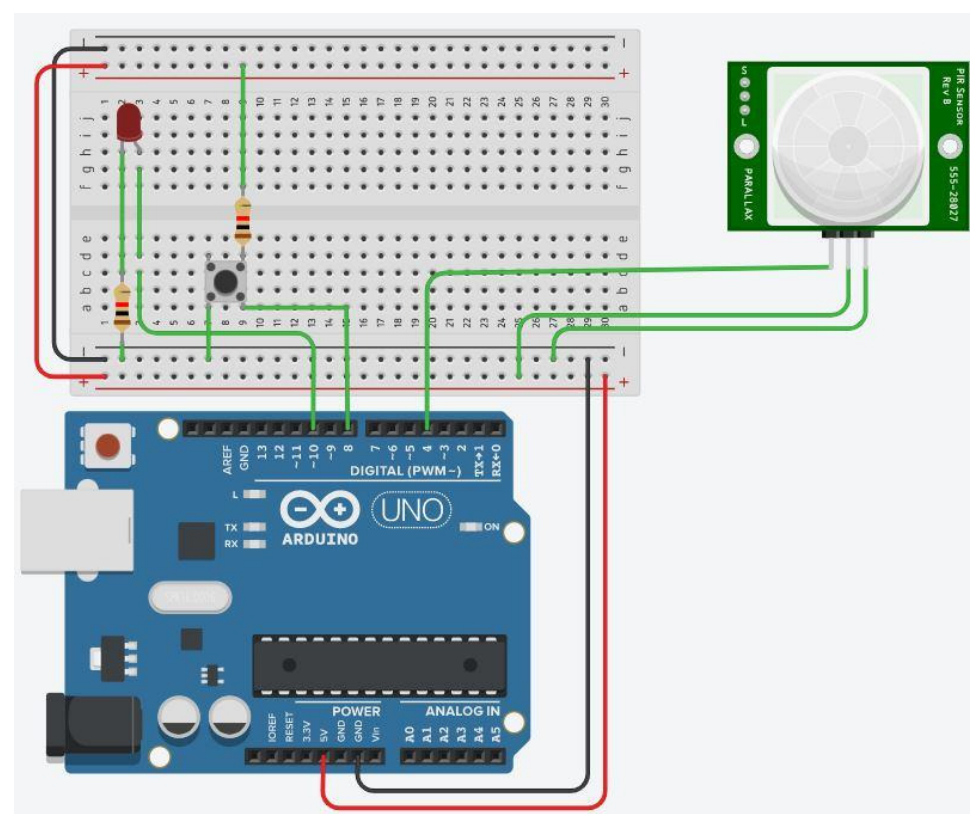


Figura B.16: Opción de conexión circuito.

Cierre (7 minutos)

- Recordar las actividades que se ejecutaron. Solicitar a los estudiantes que señalen qué aprendieron, qué les gustó más y qué les resultó más complicado.
- El tutor o tutora, debe considerar estos comentarios para las próximas sesiones.

- Verificar asistencia nuevamente.
- Despedirse y recordar horario de próxima reunión.

C | Anexo

Este anexo contiene los instrumentos de evaluación diseñado para los desafíos presentes en las sesiones.

Desafío Sesión N°1

Proyecto casa.

Dimensión	Indicadores
I Aplicación de pensamiento lógico computacional (software)	1.1 Se presenta el plano de trabajo, con sus ejes coordenados.
	1.2 Indica coordenadas para trasladar figuras en distintos lugares del plano.
	1.3 Determina dimensiones de la figura.
II Escala y desplazamiento:	2.1 Realiza una composición con diversas figuras geométricas.
	2.2 Mueve la figura a través de coordenadas.
III Uso de Colores	3.1 Escoge entre una gama de colores para cada objeto seleccionándolo y pinchando en “sólido”.
IV Composición de figura	4.1 Fusionar dos figuras distintas creando composición.
	4.2 Utiliza opción “hueco” para crear espacio vacío dentro de la figura sólida.
V Presentación oral	5.1 Tono voz audible por todos los presentes.
	5.2 Modulación permite un adecuado entendimiento de lo presentado.
	5.3 Utiliza lenguaje formal culto.
	5.4 Evita el uso de muletillas.
	5.5 Hay un orden lógico en las ideas presentadas.

Figura C.1: Instrumento Desafío N°1.

- 0: No logrado
- 1: Medianamente logrado
- 2: Logrado
- 3: Destacado

Desafío Sesión N°2

La Aldea Konoha/Kage Bunshin no Jutsu.

Dimensión	Indicadores
I Dominio de espacio tridimensional	1.1 Se presenta el plano de trabajo, con sus ejes coordenados.
	1.2 Indica coordenadas para trasladar figuras en distintos lugares del plano.
	1.3 Determina dimensiones de la figura.
II Escala y desplazamiento	2.1 Realiza una composición con diversas figuras geométricas.
	2.2 Mueve la figura a través de coordenadas.
III Composición de figura	3.1 Fusionar dos figuras distintas creando composición.
	3.2 Utiliza opción “hueco” para crear espacio vacío dentro de la figura sólida.
IV Desplazamiento por el plano	4.1 Logra dimensionar el espacio de la casa de modo que sus réplicas se distribuyen en el plano sin traslaparse.
	4.2 Logra dimensionar el espacio de la aldea considerando lugares de tránsito para el avatar.
	4.3 Logra que avatar se desplace por los lugares de tránsito previamente establecidos.
	4.4 Logra movimiento en más de un eje.
V Repetición de Instrucciones	5.1 Comprende el orden de ejecución de las instrucciones en bloque estableciendo un algoritmo de repetición.
	5.2 Usa el bloque de repetición de la forma indicada en la actividad.
	5.3 El avatar realiza un recorrido en forma de circuito repetitivo.
VI Presentación oral	6.1 Tono voz audible por todos los presentes.
	6.2 Modulación permite un adecuado entendimiento de lo presentado.
	6.3 Utiliza lenguaje formal culto.
	6.4 Evita el uso de muletillas.
	6.5 Hay un orden lógico en las ideas presentadas.

Figura C.2: Instrumento Desafío N°2.

- 0: No logrado
- 1: Medianamente logrado
- 2: Logrado
- 3: Destacado

Desafío Sesión N°3

La Aldea Konoha/Kage Bunshin no Jutsu.

Dimensión	Indicadores
I Aplicación de pensamiento lógico computacional (software)	1.1 Comprende el orden de ejecución de las instrucciones en bloque estableciendo un algoritmo de repetición.
	1.2 Usa el bloque de repetición de la forma indicada en la actividad.
	1.3 Logra simultaneidad de los estados de los 4 semáforos.
	1.4 Programa los pines de arduino en concordancia al circuito armado.
	1.5 Establece un orden lógico en el encendido de las luces del semáforo.
II Eso de componentes Tinkercad (Hardware)	2.1 Identificación de componentes relevantes: Leds, Resistencias, Protoboard, Cables, Arduino y Batería.
	2.2 Se observa orden físico de los componentes, Arduino y protoboard, que permite entender a simple vista las conexiones existentes.
	2.3 Uso de los diversos componentes permitiendo al Hardware ejecutar la funcionalidad especificada en la actividad.
III Trabajo de pares	3.1 Se desarrolló una comunicación efectiva entre los integrantes del grupo, permitiendo la comprensión mutua de sus respectivos trabajos.
	3.2 Logran establecer semejanzas y diferencias entre sus respectivos trabajos.
	3.3 En la exposición relata dinámicas colaborativas, existiendo coherencia en las presentaciones.
IV Presentación oral	4.1 Tono voz audible por todos los presentes.
	4.2 Modulación permite un adecuado entendimiento de lo presentado.
	4.3 Utiliza lenguaje formal culto.
	4.4 Evita el uso de muletillas.
	4.5 Hay un orden lógico en las ideas presentadas.

Figura C.3: Instrumento Desafío N°3.

- 0: No logrado
- 1: Medianamente logrado
- 2: Logrado
- 3: Destacado

Desafío Sesión N°4

Mecanismo de emergencia.

Dimensión	Indicadores
I Definición de variables globales	1.1 Se definen las variables VERDE, AMARILLO y ROJO fuera de las funciones void setup() y void loop()
	1.2 Define el tipo de variable correctamente: int, char, float, boolean, etc
II Inicialización de variables	2.1 Se inicializan el total de las variables: 6
	2.2 Todas se inicializan en la función void setup()
	2.3 Los modos de los pines corresponden a INPUT los conectados a los switch y a OUTPUT los conectados a los leds.
	2.4 Los números de los pines inicializados corresponden a los conectados en el circuito.
III Instrucciones repetitivas en el código	3.1 Usa la función digitalWrite() para leer los pines conectados a los switch.
	3.2 Almacena en las variables VERDE, ROJO y AMARILLO los valores leídos (asignación)
	3.3 Usa la función digitalWrite() para asignarle un valor a los pines conectados a los leds.
	3.4 En la función digitalWrite() el valor a escribir en los pines es el valor almacenado en las variables VERDE, ROJO y AMARILLO.
IV Comunicación Serial	4.1 Inicia la comunicación serial con la línea Serial.begin(9600); dentro de void setup().
	4.2 Conoce cómo utilizar la función Serial.print() para escribir texto y escribir variables (incluyendo sintaxis de "").
	4.3 Ordena coherentemente las instrucciones de qué es lo que quiere ver en el monitor serial, y las escribe en la función void loop().
V Presentación oral	5.1 Tono voz audible por todos los presentes.
	5.2 Modulación permite un adecuado entendimiento de lo presentado.
	5.3 Utiliza lenguaje formal culto.
	5.4 Evita el uso de muletillas.
	5.5 Hay un orden lógico en las ideas presentadas.

Figura C.4: Instrumento Desafío N°4.

- 0: No logrado
- 1: Medianamente logrado
- 2: Logrado
- 3: Destacado

Desafío Sesión N°5

El Menú.

Dimensión	Indicadores
I Aplicación de pensamiento lógico computacional (software)	1.1 Comprende el orden de ejecución de las instrucciones en código propuesto permitiendole desarrollar el problema.
	1.2 Reconoce las variables de salida del código y las declara en la ubicación correcta .
	1.3 Comprende a nivel de su función en el código el bloque for y lo utiliza correctamente.
	1.4 Identifica y logra plantear claramente los case existentes del problema.
	1.5 Entiende claramente el uso del break dentro de cada case.
	1.6 Utiliza e imprime en el monitor serial el platillo que se pide en tiempo real.
	1.7 Respeta la sintaxis del language Arduino.
	1.8 Existe orden en el código y se respetan las tabulaciones.
II Eso de componentes Tinkercad (Hardware)	2.1 Identificación de componentes relevantes: Leds, Resistencias, Protoboard, Cables, Arduino y Batería.
	2.2 Identifica en el circuito la correcta conexión de un LED. Conexiones a tierra, además de resistencia limitante de corriente y pin de salida.
III Trabajo de pares	3.1 Se desarrolló una comunicación efectiva entre los integrantes del grupo, permitiendo la comprensión mutua de sus respectivos trabajos.
	3.2 Logran establecer semejanzas y diferencias entre sus respectivos trabajos.
	3.3 En la exposición relata dinámicas colaborativas, existiendo coherencia en las presentaciones.
IV Presentación oral	4.1 Tono voz audible por todos los presentes.
	4.2 Modulación permite un adecuado entendimiento de lo presentado.
	4.3 Utiliza lenguaje formal culto.
	4.4 Evita el uso de muletillas.
	4.5 Hay un orden lógico en las ideas presentadas.

Figura C.5: Instrumento Desafío N°5.

- 0: No logrado
- 1: Medianamente logrado
- 2: Logrado
- 3: Destacado

Desafío Sesión N°7

¡Con mi queso no!

Dimensión	Indicadores
I Conexiones electrónicas	1.1 Conecta correctamente Arduino con pulsador.
	1.2 Conecta el led correctamente.
	1.3 Conecta correctamente el sensor PIR
	1.4 Hace uso correcto de la protoard.
II Programación	2.1 Sintaxis correcta y orden al programar.
	2.2 Define pines utilizados de forma correcta.
	2.3 Hace uso de condicionales correctamente.
	2.4 Hace escritura y lectura correcta de pines digitales.
	2.5 Hace uso de el monitor serial de forma correcta.
	2.6 Selecciona información relevante desde los datasheets para la programación de los sensores involucrados.
	2.7 Comprende la diferencia entre pines de entrada y salida.
III Pensamiento lógico del problema (Indicadores esta dimensión serán dicotómicos, o sea solo habrán dos opciones, o se logró o no se logró)	3.1 El led rojo solo debe encenderse cuando haya un intruso en movimiento.
	3.2 El monitor serial debe enviar el mensaje a la policia solo cuando se presione el pulsador.
	3.3 Comprende la independencia de ambos condicionales.
IV Presentación oral	4.1 Tono voz audible por todos los presentes.
	4.2 Modulación permite un adecuado entendimiento de lo presentado.
	4.3 Utiliza lenguaje formal culto.
	4.4 Evita el uso de muletillas.
	4.5 Hay un orden lógico en las ideas presentadas.

Figura C.6: Instrumento Desafío N°7.

- 0: No logrado
- 1: Medianamente logrado
- 2: Logrado
- 3: Destacado