

Micro:gim

Deporte, salud y placas micro:bit


Micro:gim propone utilizar placas micro:bit para asistir la práctica de la gimnasia y el deporte. Diferentes sensores incorporados en las placas se programan para construir instrumentos de medición que forman parte de dispositivos físicos contruidos con material reciclable.

En función de un circuito de prácticas corporales, acordado entre estudiantes y docentes, se diseñarán dispositivos que usen las placas disponibles para asistir la práctica automática. Por ejemplo, avisar cuando se complete una cantidad de saltos prevista o contar la cantidad de abdominales o flexiones realizadas en un determinado período.

El resultado de esta propuesta puede abordarse en el marco de un proyecto escolar de Salud y vincularse con el Área del Conocimiento Corporal.

Duración: 9 a 12 semanas.

Materiales: Microbit.

Ficha Curricular ↓	2
Síntesis del recorrido	4
Acuerdos iniciales de coordinación	5
Placas micro:bit	6
ETAPA 1 ↓	7
ETAPA 2 ↓	12
ETAPA 3 ↓	18
ETAPA 4 ↓	25
ETAPA 5 ↓	35
ETAPA 6 ↓	39
 ANEXO 1	43

Objetivos de aprendizaje de 2º año de Pensamiento Computacional

Comunicación y Colaboración

- Participar de forma proactiva en un proyecto grupal.
- Transmitir y escuchar ideas dentro del grupo de trabajo.
- Integrar el uso de herramientas de documentación de la información y el proceso de desarrollo del proyecto.

Computación, Sociedad y Equidad

- Entender que un dispositivo automatizado puede interactuar con el entorno.

Resolución de problemas computacionales

- Utilizar la estrategia de división de un problema en subproblemas.
- Resolver problemas computacionales utilizando algunas herramientas básicas de programación (como la alternativa condicional, las repeticiones, las variables, etc).
- Recuperar soluciones construidas en experiencias anteriores para adaptarlas a nuevos problemas.

Análisis de datos, desarrollo y uso de Abstracciones

- Identificar los aspectos importantes y seleccionar la información relevante de los datos de un problema.
- Reconocer las generalidades de los comportamientos de los programas, en términos abstractos, para que puedan ser adaptados, combinados y/o reutilizados.

Creación de algoritmos, programas y dispositivos

- Controlar aspectos gráficos de sus programas.
- Utilizar en sus programas de forma independiente o combinada bloques de control, variables, sensores, eventos y operadores.

- Reconocer el uso del método de desarrollo incremental.

- Planificar, crear y modificar un programa y/o solución tecnológica particular que interactúe con el entorno.

Evaluación

- Probar, ajustar y corregir mediante la iteración durante el desarrollo de las creaciones.

Contenidos PC

- Modelo abstracto de computadora • Alternativa condicional • Variables • Sensores • Dispositivos de entrada
-

Perspectiva de género

Propiciar una experiencia educativa inclusiva y promotora de equidad de género que desnaturalice en forma constante el sesgo de la computación como tarea exclusiva de varones. Buscamos incentivar el trabajo de las niñas y brindarles las herramientas necesarias (atención, apoyo, retroalimentación positiva, entre otras).

Referencias al Marco Curricular Nacional

Espacio Técnico - Tecnológico. Unidad curricular Ciencias de la Computación y Tecnología Educativa. Tramo 4

Competencias generales

Comunicación, Pensamiento creativo, Pensamiento computacional, Metacognitiva, Relación con los otros, Iniciativa y Orientación a la Acción.

Competencias específicas

CE4 Indaga e identifica técnicas, herramientas y aplicaciones que la tecnología proporciona para la resolución de problemas así como situaciones que pueden abordarse como problemas computacionales.

CE5. Recupera soluciones propias o ajenas y construye modelos, para resolver problemas simples, en grupo y de forma mediada, enriqueciendo sus construcciones y las de otros.

CE6. Utiliza la programación y dispositivos tecnológicos en la implementación colectiva de soluciones para la resolución de problemas.

Contenidos específicos

Pensamiento Computacional. Estrategias para la resolución de problemas o creación de juegos y otros recursos: patrones, reutilización, descomposición, iteración, ensayo y error, método incremental, entre otros.

Programación en lenguajes de bloques: aspectos gráficos, bloques de control, variables, sensores, eventos y operadores.

Criterios de logro

Reconoce patrones o características comunes entre elementos y situaciones en la resolución de problemas.

Divide un problema en subproblemas, analizando datos importantes e información relevante.

Aplica soluciones conocidas en nuevos contextos para elaborar procedimientos más complejos.

Resuelve problemas computacionales utilizando algunas herramientas básicas de programación (condicionales, iteraciones, variables, etc.).

Planifica, crea y modifica, con ayuda del docente, un programa o solución tecnológica.

Posibles vinculaciones a otros espacios y unidades curriculares.
A definir por maestro/a de aula

La temática de Deporte y salud, que desde las VC se abordará a través de la creación de instrumentos de medición de prácticas corporales con las placas micro:bit, es una oportunidad para abordar desde el aula contenidos de salud en general: salud cardiovascular, respiración, el sistema de regulación y coordinación: el sistema nervioso; la promoción de la actividad física y la alimentación sana y equilibrada para cuidar la salud.

Educación Física:

Conciencia corporal. Las nociones topológicas y de temporalidad en prácticas motoras combinadas con ritmos propios y secuencias rítmicas, individual y colectivamente

Deporte. Diferentes deportes, sus reglas y su práctica deportiva de forma autónoma (terrestre, acuáticos, aéreos).

Gimnasia. Secuencias gimnásticas y acrobacias con las habilidades aprendidas mediadas con docentes incluyendo elementos o música.

Ciencias del ambiente (Biología):

Nutrición. Dieta y Salud. Respiración. Aparato circulatorio. Sistema nervioso.

Matemática:

Sistema métrico decimal. Múltiplos y submúltiplos (metro cuadrado, centímetro cuadrado, kilómetro cuadrado, hectárea).

Introducción a la estadística y a la probabilidad. Gráficos: diagrama de barras, histogramas Espacio muestral

Materiales complementarios sugeridos



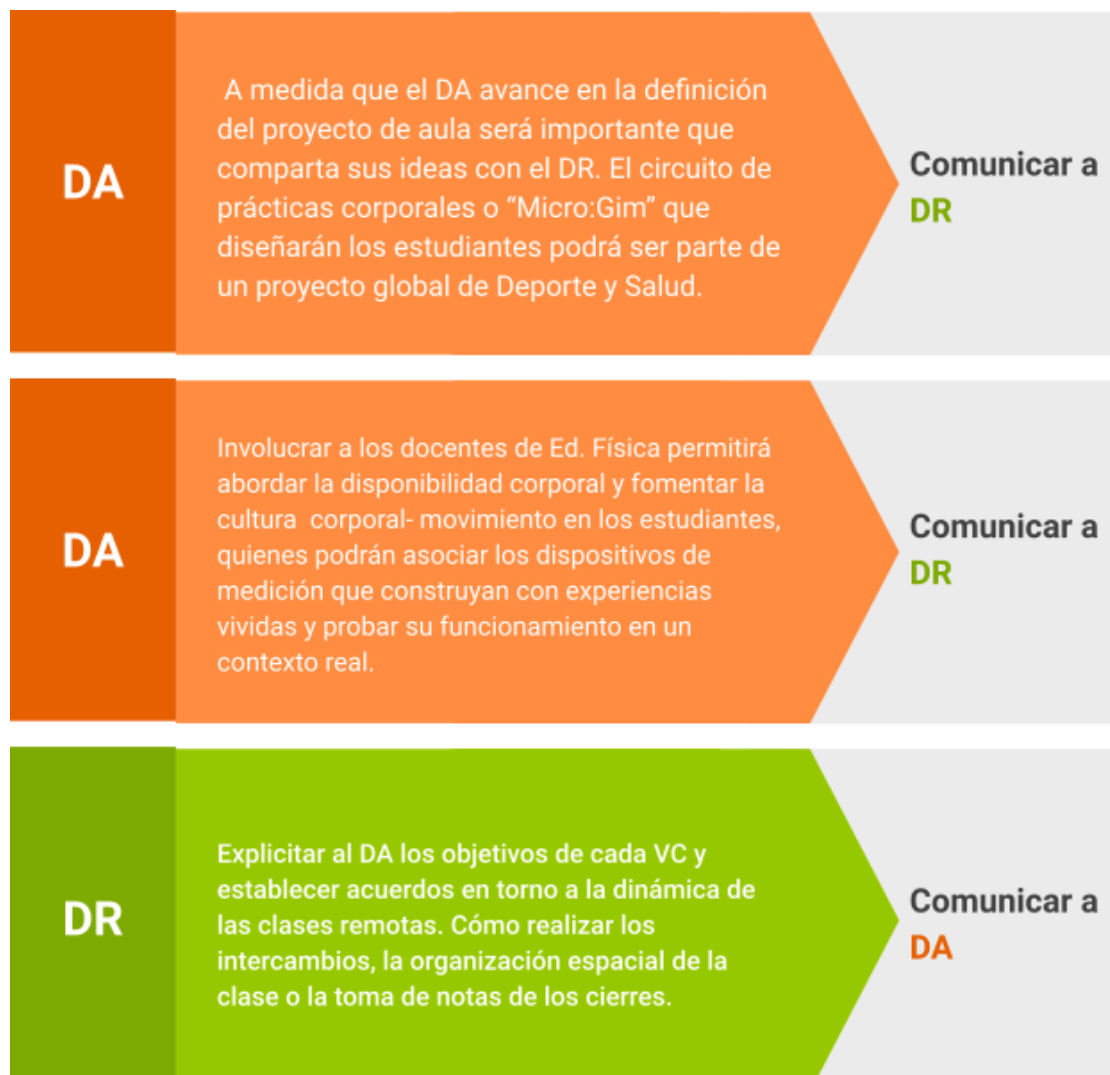
Síntesis del recorrido



⌚ En esta etapa, las actividades de la VC están previstas para más de un módulo de 45 minutos.

Acuerdos iniciales de coordinación

El diálogo permanente entre **docente remoto (DR)** y **docente de aula (DA)** es fundamental para llevar adelante esta propuesta.



DA	DR
Rol del DA durante las VC <ul style="list-style-type: none">● Durante las actividades de inicio organiza el intercambio para que los estudiantes relaten al DR lo realizado en el aula.● Durante las actividades de desarrollo, será importante intervenir para vincular el trabajo a lo realizado en el aula y al proyecto global en el que se inscribe esta propuesta.● Durante las actividades de cierre y reflexión, su participación es fundamental para recuperar momentos que haya observado durante el desarrollo de las actividades y apelar a experiencias previas de los estudiantes que aporten a las reflexiones propuestas por el DR.	
Rol del DR <ul style="list-style-type: none">● Consultar al DA todo lo necesario para estar al tanto del modo en que se abordará el contenido programático que se pondrá en juego en la programación.● Anticipar al DA el modo y el contenido planificado para cada VC.● Llevar adelante las clases por VC en conjunto con el DA.● Gestionar el curso en Crea de la propuesta, realizar los ajustes necesarios y las devoluciones a los estudiantes de sus programas o participaciones en foros.	

Placas micro:bit

El DR necesita conocer las experiencias previas por parte de los estudiantes y el DA en el trabajo con placas micro:bit.

En caso de que sea la primera experiencia del DA, el DR realiza una introducción sobre el funcionamiento de la placa, especialmente destinada a anticipar la dinámica de trabajo y la operatoria de las placas que sucederá en todas las VC de la propuesta:

- El uso del entorno Makecode <makecode.microbit.org>.
- El guardado del programa en un archivo .hex en la computadora.
- La conexión de la placa a la computadora mediante el cable USB.
- El copiado del archivo .hex a la placa a través del administrador de archivos.
- El uso del portador de pilas para que la placa funcione sin cable.

Seguramente se irá afianzando este procedimiento en forma paulatina a partir de la colaboración entre DA y DR.

Disponibilidad de placas micro:bit entre los estudiantes

Como mínimo se sugiere tener 1 placa cada 3 o 4 estudiantes e idealmente que la mayoría disponga de su placa. En el sitio <https://microbit.ceibal.edu.uy/> sección Recursos y allí Documentos encontrarán respuestas a las preguntas para [solicitar una micro:bit](#).

Tutoriales Micro:bit

En el canal de youtube micro:bit Plan Ceibal, el **video ¿Cómo programar mi micro:bit?** <https://youtu.be/pKt5k1wXSg> explica el proceso completo del armado, la programación y la instalación de un programa en la placa *micro:bit*.

En el **sitio micro:bit del Plan Ceibal** <https://microbit.ceibal.edu.uy/>. En la sección "Recursos" está disponible el video [Mis primeros pasos](#) o en pdf la [Guía básica](#).

Curso en plataforma Crea ↓



Se destinará una carpeta en Crea para este proyecto dentro del Curso de PC, organizando cada carpeta internamente en las etapas previstas. En este espacio virtual de encuentro, el DR compartirá consignas, foros de intercambio y actividades interactivas. Los estudiantes publicarán allí tanto los avances en sus bitácoras como en los archivos .hex

Registro de actividades de reflexión

A lo largo de toda esta propuesta se propone plasmar los intercambios producto de las actividades de cierre en un mapa conceptual común para toda la clase que se va enriqueciendo en cada etapa.

Cada pareja de docentes considerará la herramienta más adecuada (Mindomo, Mindmeister, Bubble.us, u otras) que permita compartir un enlace con los estudiantes.

Las dinámicas para la escritura en este archivo podrán ir variando entre una etapa y otra. Algunas veces se podrán mostrar recuadros para que un estudiante designado los rellene, otras veces podrán ser conceptos para reordenar o clasificar o preguntas para responder. Otras actividades de cierre se podrán plasmar directamente en el foro o en otro tipo de documentos.

ETAPA 1 ↓

Introducción a la propuesta

En el aula, se da comienzo a la propuesta de trabajo. Los estudiantes recuperan sus experiencias y conocimientos sobre la aplicación de las computadoras al servicio del deporte, a la vez que dan inicio, junto a su DA, del proyecto áulico de Deporte y Salud. Se plantea el desafío de diseñar un circuito de prácticas motrices basado en estaciones.

En la VC, desarrollan un juego con temática deportiva como introducción o repaso de la programación de las placas y el uso de sus botones. Para terminar, se reflexiona sobre las características generales de una computadora para concluir que la micro:bit es una de ellas.

Objetivos

Se espera que los estudiantes sean capaces de:

- Utilizar el entorno Makecode y programar los botones de las placas.
- Reconocer que la placa micro:bit es una computadora, que recibe datos de entrada, ejecuta un programa y produce datos de salida.
- Establecer relaciones entre la programación de micro:bit y de Scratch para observar generalidades de los entornos y lenguajes de programación.

Coordinación dupla pedagógica**Información que necesita tener el DR:**

- El DA comparte el diseño del proyecto de aula que se articulará con la propuesta del Micro:Gim.

Decisiones del DA:

- Conformar grupos de 2 a 4 integrantes para trabajar a lo largo de toda la propuesta.
- Al finalizar la VC, los estudiantes dispondrán de sus micro:bit con un programa para hacer adivinanzas que se sugiere utilizar en otras clases o instancias.

Decisiones conjuntas entre DA y DR:

- La participación del DA para gestionar la primera experiencia de los estudiantes con las placas micro:bit será muy valiosa (la descarga del archivo .hex en sus computadoras, la conexión de la placa con el cable usb y el copiado del archivo en la placa). En comunicación y con la guía del DR, el DA puede acompañar a los estudiantes que necesitan más ayuda.

AULA ↓ Presentación de la propuesta

Propósitos mínimos

- Alentar a los estudiantes para que establezcan relaciones entre las computadoras, el deporte y la salud.

Propósitos óptimos

- Presentar el proyecto de aula que se articulará con la propuesta de Pensamiento computacional.

Notas para el DA ↓



Considerar el planteo del desafío global a abordar en las clases de PC para presentarlo en el marco del Proyecto de Aula:

Idear y diseñar un circuito de prácticas motrices compuesto por varias estaciones. Cada grupo de trabajo será responsable de una estación y tendrán que resolver cómo programar una placa micro:bit que permita medir la actividad realizada.

Por ejemplo, si consideran una estación de abdominales, deberán crear un programa para la micro:bit que para contar la cantidad de abdominales que se realiza en un determinado tiempo.

Se sugiere trabajar en la identificación de aplicaciones de computadoras al servicio del deporte y la salud y registrar, como mínimo, el nombre del dispositivo y su función en el deporte o la salud. Ej. Medidor de glucosa / Para controlar el nivel de glucosa en sangre. Tensiómetro / Para medir la presión arterial.

Algunos dispositivos que se asemejan a lo que realizaremos con la placa micro:bit son las las "pulseras inteligentes" para llevar control de la actividad física: contar pasos diarios, distancias recorridas, tiempo en movimiento. Con esta información puede indicarnos el avance de un objetivo diario de actividad.



Videos ilustrativos:

- "Pulseras de actividad ¿qué son?" (<https://youtu.be/Pcv4l4lfxPw>) (primer minuto y medio) nos puede ayudar a comprender de qué se tratan.
- "¿Funcionan realmente bien los pulsómetros y pulseras de fitness? Los ponemos a prueba" (https://youtu.be/Fwua_x4LIaW) (primeros tres minutos) muestra las pulseras en uso.

VC 1↓

Inicio con micro:bit

🕒 Dedicar a la actividad el tiempo necesario para que los estudiantes se familiaricen con el entorno Makecode, el simulador y el proceso de descarga del programa y su instalación en la placa.

1. Inicio (5 min)

Se recuperan las aplicaciones al servicio del deporte y la salud que relevaron en el aula y lo que se imaginan realizaremos en PC a partir de la presentación de la propuesta. Se sugiere repasar el desafío global del proyecto para asegurarse que los estudiantes conozcan la propuesta.

2. Desarrollo

Primer programa en micro:bit. Juego grupal (30 min)

El objetivo de esta actividad es trabajar sobre una consigna sencilla que se resuelva de forma rápida e involucre la programación de los botones, el guardado del archivo .hex y la instalación del programa en la placa. Para ello se les presenta el desafío de programar una placa para “esconder un deporte”. Es decir, realizar una adivinanza relacionada al deporte utilizando la micro:bit. La placa deberá mostrar pistas cuando se presionen los distintos botones, para que el resto de la clase adivine. Por ejemplo, para esconder “natación”:

- al presionar el botón A, se muestra la inicial del deporte (“N”)
- al presionar el botón B, se muestra una pista de una palabra (“agua”)
- al presionar el botón A+B, se muestra otra pista de una palabra (“pileta”)

Los estudiantes inician el desafío y se habilita la exploración en el entorno de programación. De ser necesario, el DR muestra cómo ingresar al sitio de MakeCode para micro:bit (<https://makecode.microbit.org/>) y a abrir un nuevo programa.

- ¿Qué instrucciones queremos darle a la placa? ¿Qué componentes están involucrados?



Desafío

Programar una placa para “esconder” un deporte.

La placa deberá mostrar pistas cuando se presionen los distintos botones, para que el resto de la clase adivine.

Los grupos deben reconocer que necesitan los bloques *al presionar el botón* y *mostrar cadena* o *mostrar leds*.

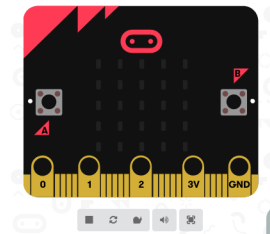
Cuando puedan ver en el simulador que algunos de los botones se comportan como es deseado, se hace una puesta en común de los bloques necesarios. Asimismo, durante el desarrollo, también se motiva la búsqueda de similitudes entre los eventos de micro:bit y los eventos en Scratch: si bien Scratch y micro:bit son muy distintas en apariencia, ambas son herramientas para programar computadoras, y, por lo tanto, comparten muchas similitudes que les permitirán a los estudiantes aprovechar sus habilidades de Scratch en este nuevo entorno.

- ¿Qué elementos usaron para construir su programa? ¿Cómo dábamos instrucciones en Scratch? ¿Qué parecidos y diferencias encuentran entre Scratch y MakeCode?



Atención: metodología de trabajo durante todo el proyecto

Es importante señalar la **función del simulador de la placa**. Mientras se esté ajustando el programa, se utilizará esta herramienta como **una manera rápida y controlada de realizar pruebas**. Recién con una versión confiable del programa se procederá a su descarga e instalación en las placas para una prueba más exhaustiva y su utilización en el mundo físico. **Es importante que en todas las VC exista una instancia de ejecución en las placas** para materializar el trabajo realizado y tener presente la forma del producto final.





Variantes de complejidad

Aquellos grupos que estén familiarizados con la dinámica de programación, prueba e instalación del programa pueden complejizar el contenido de las adivinanzas. Por ejemplo, mostrar en la pantalla el texto completo de la adivinanza o brindar las pistas en forma de dibujo o animación utilizando los bloques de LEDs.

3. Cierre (10 minutos)

Hacia una conceptualización más general de las computadoras para identificar a la micro:bit como una de ellas.

¿Cuántas computadoras hay en el salón de clases en este momento? ¿Qué computadoras conocen?

Es esperable que los estudiantes contabilicen las ceibalitas u otros dispositivos de escritorio o portátiles disponibles. Sin embargo, esta representación deja afuera muchísimos dispositivos computacionales con los que interactuamos diariamente.

Se plantea a los estudiantes que lo que distingue a una computadora son 3 características básicas y que todas deben estar presentes:

- Toma datos, acepta entradas, recibe estímulos
- A partir de los datos de entrada, toma decisiones, calcula y/o memoriza algo
- Genera una salida o mueve alguna cosa.

Se propone realizar el *Test ¿Es una computadora?* (Ver [Anexo 1](#)). Los estudiantes proponen artefactos que les parece que son computadoras. Si no surgen ideas señalar algunos para empezar (consola de juegos, un celular, otros artefactos curiosos como el inodoro inteligente). Brindar ejemplos

donde no intervienen dispositivos digitales también sirve para arrojar luz sobre la frontera entre aquello a lo que se llama computadora y aquello que no lo es (una tabla de picar o almohada “inteligente”). El DR va completando la tabla con cada artefacto que mencionan los estudiantes y atraviesan juntos el análisis de las 3 preguntas. Finalmente se hace pasar por este test a la placa micro:bit.

¿Qué es lo que determina qué sucede cuando apretamos los botones de la micro:bit? ¿Qué será lo que determina qué sucede cuando apretamos los botones de la Nintendo o el celular?

A partir de reconocer a la placa micro:bit como una computadora y de la experiencia de programación, los estudiantes deben **identificar el rol del programa** en el comportamiento de la micro:bit y generalizarlo a los otros ejemplos de computadoras que propusieron. También se señala que los programas se construyen usando computadoras en un entorno como Scratch o MakeCode.

¿Fue muy distinto programar en MakeCode que en Scratch? ¿Por qué será? ¿Se animarían a utilizar un nuevo entorno de programación con bloques?

El DR agrega a las reflexiones de los estudiantes que las formas en las que se programan las computadoras suelen ser muy similares unas a otras, por lo tanto, es importante **no concentrarse en la herramienta** utilizada para programar **sino en lo que podemos expresar** con los bloques disponibles, qué tipos de instrucciones o comportamientos podemos indicar, como los eventos. Estos suelen ser compartidos por muchas herramientas de programación, por lo cual conocerlos permite utilizar cualquiera de ellas más fácilmente.

Foro y bitácora

El DR deja en Crea el enlace al archivo con la tabla creada para completar el Test.
Se invita a seguir analizando artefactos con la Actividad interactiva.

El intercambio de la comparación de entornos puede dar inicio a la Bitácora y sintetizarse en un esquema.
Además, en el foro los estudiantes pueden adjuntar sus programas y una breve descripción de hasta donde llegaron con su programación.



La Yapa: Propuestas para seguir en casa

Al abrir un programa nuevo aparecen dos bloques: al iniciar y para siempre. ¿Qué hará cada uno? ¿Cómo los pueden usar para hacer más vistosa la adivinanza? Por ejemplo, que al encenderse, la placa muestre un dibujo o diga las instrucciones del juego, o que cuando estamos esperando a que se aprieten los botones se muestre un dibujo o una animación.



ETAPA 2 ↓ Contador

En esta etapa se avanza sobre la temática de Deporte y salud.

En el aula, se trabaja específicamente sobre el conteo o las mediciones en el contexto de las prácticas motrices o entrenamiento deportivo.

En la VC, se construye un contador utilizando los botones de la micro:bit para contar cuántas veces saltan los estudiantes. Para esto, se presentan las variables cuya importancia se retoma en la reflexión final. Este primer programa será insumo para las próximas etapas.

Objetivos

Se espera que los estudiantes sean capaces de:

- Reconocer el hecho de que los programas necesitan almacenar datos para funcionar y que las variables son las herramientas de programación que se utilizan para esto.
- Identificar situaciones en las que un programa o computadora recuerda un valor a lo largo de su ejecución.

Coordinación dupla pedagógica

Decisiones del DA:

- Esta etapa puede ser una oportunidad para abordar en el aula contenidos del área de matemáticas vinculados a diferentes unidades de medida.

Información que necesita tener el DR: Si el docente de Ed. Física se suma a la propuesta, en esta etapa serán bienvenidas experiencias de mediciones y conteos durante sus prácticas corporales.

- Para introducir la noción de variable se requerirán de materiales concretos. Acordar la colaboración del DA con los elementos disponibles para la dramatización del guardado de información.
- La propuesta para la VC de esta etapa es contar saltos realizados, pero puede acordarse otro ejercicio repetitivo que pueda realizarse en el aula de VC.

AULA ↓ Mediciones y conteos

Notas para el DA ↓



Propósitos mínimos

- Impulsar en los estudiantes la identificación de mediciones y conteos durante la actividad física.

Propósitos óptimos

- Presentar el proyecto de aula desde el punto de vista de los contenidos curriculares seleccionados.

Para introducir a los estudiantes en la relación entre el deporte y las mediciones, se sugiere recoger sus experiencias y asociaciones. *¿Qué se cuenta cuando hacemos ejercicios motores? ¿Qué se mide en una carrera? ¿Y en un salto en largo? ¿En un lanzamiento de jabalina?* Seguramente el DA encontrará ejemplos cercanos a los estudiantes para guiarlos. También puede ser de gran valor sumar al docente de educación física en esta tarea.

El video “Test de capacidades: Fuerza, Resistencia, Flexibilidad y velocidad” puede contribuir a ver las mediciones en la práctica. <https://youtu.be/duAgaeqzpAY>.

Se sugiere registrar las ideas que surgen en el que se distinga el ejercicio físico y una pregunta que remita a la forma de medirlo o contarlo. Será interesante compartirlo luego con el DR. Por ejemplo:

- Velocidad: ¿Cuánto demoras en recorrer 10 metros?
- Fuerza abdominal: ¿Cuántos abdominales puedes realizar en 60 segundos?
- Resistencia: ¿Cuántas veces puedes recorrer un circuito en 2 minutos?

Es una oportunidad para vincular la tarea con diferentes unidades de medida.

VC 2↓ Contar

Desafío

Programar la placa para contar saltos utilizando los botones.

Cada vez que se presiona un botón, la cuenta aumenta en uno y se ve el número en la pantalla. Al presionar el botón A+B, se muestra la cantidad registrada.

1. Inicio

Recuperación y presentación del desafío (10 min)

¿Qué ejercicios físicos que identificaron con el DA, se pueden medir? ¿Cómo se imaginan la micro:bit que puede ayudar a medir esos ejercicios? ¿Algunos de ellos requieren contar?

Para **motivar el desafío**, el DR propone contar en voz alta entre todos cuántos saltos en el lugar puede hacer un estudiante antes de que el DR active una alarma o diga ¡alto! En una segunda oportunidad mientras otro estudiante salta, el DR prolonga más el tiempo para **hacer evidente que la tarea de contar puede ser tediosa o difícil de sostener**.

¿Es difícil contar? ¿Es fácil contar sin equivocarse?

El DR propone contar ayudándose con la micro:bit. De esta manera, no hace falta decir los números en voz alta, ni recordar el conteo, simplemente presionar un botón de la microbit por cada salto y mostrar su pantalla al resto. Se presenta, entonces, el desafío para esta VC: programar la placa para contar saltos utilizando los botones. Cada vez que se presiona un botón, la cuenta aumenta en uno y se ve el número en la pantalla. Al presionar el botón A+B, se muestra la cantidad registrada.

2. Desarrollo

Introducción de la noción de variable (30 min)

Se habilita una breve exploración en Makecode, para que los grupos identifiquen que necesitarán un bloque *al presionar el botón A* (al igual que en

la etapa anterior) y que todavía no conocen una manera de incorporar números ni de operar con ellos en un programa.

- *¿Qué partes del programa ya podrían resolver con lo que hicimos en la VC anterior? ¿Qué les falta? ¿Qué aparece en este programa que no aparecía en el de la adivinanza? Se busca que aparezca la necesidad de trabajar con números. ¿Qué pasaba en el programa de adivinanza cada vez que presionamos el botón A? ¿Y en este que debe pasar?*

Se buscan diferencias entre el programa contador y el de la adivinanza para distinguir que el contador, cada vez que se presiona el botón, debe mostrar un número diferente (mientras que en la adivinanza, siempre muestra lo mismo).

- *¿El contador debe mostrar cualquier número? ¿Cómo sabe cuál es?*

Los estudiantes deberán identificar que para que el programa cuente tiene que tener un valor almacenado que se modifica durante la ejecución: cada vez que presionamos el botón, el número almacenado tiene que crecer en 1.

Una vez que está clara la necesidad de guardar información para resolver el desafío se introduce la **noción de variable**, como herramienta de programación que se puede pensar como una cajita con un nombre donde se guarda un valor para no “olvidarlo” o perderlo.

Atención: idea para dramatizar una variable

Se sugiere dramatizar esta actividad para facilitar la visualización de la variable como un espacio de almacenamiento. Utilizar una cajita etiquetada y un papel dentro, o un sector del pizarrón con un título (o de la pantalla del DR) para anotar un número y recuperarlo o modificarlo cuando sea necesario.

Se repite el ejercicio de contar saltos del principio y además se dramatiza el funcionamiento en cámara lenta del contador: cada vez que salta el estudiante, se abre la cajita y toma el número almacenado (si es en el pizarrón se lee en voz alta). Para reemplazarlo por el siguiente se tacha el número anterior o se escribe en un nuevo papel y se vuelve a guardar (o se borra en el pizarrón y escribe el nuevo número en su lugar).



¿En qué se diferencia esta manera de contar con cómo nosotros contamos 10 saltos? ¿Por qué hace falta hacerlo de esta manera en la computadora?

El objetivo es que los estudiantes reconozcan que las personas siempre tienen memoria y recuerdan cosas simples y a corto plazo "sin darse cuenta" (en este caso, el último número que contaron), mientras que la computadora no. Al contrario, es responsabilidad de quien programa decir que almacene los datos que vayan a ser necesarios en el futuro, para no perderlos.

Planificación conjunta de una estrategia de programación

El DR interpela a los estudiantes para elaborar una estrategia de solución en diferentes pasos:

- *Mirando el desafío, ¿qué información debemos almacenar? ¿Nos servirá el mismo programa para contar saltos y para contar flexiones?*

Primer paso: identificar qué información hace falta guardar.

Los estudiantes reconocen la necesidad de almacenar la cantidad de saltos. Pero en particular, en el funcionamiento del programa, se debe

guardar la cantidad de veces que se presione cada botón (por eso, podrían usar la micro:bit para contar con los botones cualquier otro fenómeno).

- *¿Cómo hacemos para guardar información durante la ejecución de un programa?*
Segundo paso: definir la caja o variable para guardar esa información. Resaltar la importancia de asignarle un nombre representativo: siguiendo el ejemplo, sería una caja llamada "saltos".
- *¿En qué momento del programa se utiliza la información almacenada? ¿Para qué? ¿Se consulta? ¿Se modifica? ¿Ambas?*
Tercer paso: identificar cuándo y cómo se utiliza la información guardada en la variable. En este caso, se le suma uno (se modifica) cuando se presiona el botón correspondiente, y se lee (se consulta) el valor guardado cuando se informa el resultado al final. También se lee el valor almacenado para informarlo al presionar A+B.

Se va tomando nota de los aportes de los estudiantes para que la planificación quede disponible para todos.

Por ejemplo:

<p>Variables:</p> <ul style="list-style-type: none"> • saltos 	<p>Quando clic en A:</p> <ul style="list-style-type: none"> - incrementar "saltos" 	<p>Quando clic en A+B:</p> <ul style="list-style-type: none"> - mostrar valor de "saltos"
--	---	--

En esta situación no es importante la inicialización de las variables pues el entorno las inicializa automáticamente en cero.

📢📢📢 Variante de complejidad

Dos contadores en uno Dado que la placa cuenta con dos botones, quienes estén familiarizados con el entorno y la dinámica de programación, pueden pensar una estrategia y hacer un programa que lleve dos cuentas separadas, cada una asociada a un botón diferente. De esta manera, la placa se puede usar para contar los saltos de dos estudiantes al mismo tiempo. En este caso, al presionar A+B debe diferenciarse a qué botón corresponde cada cantidad; pueden dibujarse flechas en la pantalla para esto.

Trabajo en Makecode

Para acompañar la programación, se deja visible la estrategia general registrada.

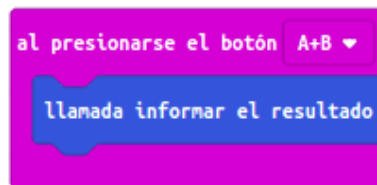
El DR muestra el procedimiento para crear una variable en el entorno MakeCode. Se recuerda la importancia de consignar un nombre representativo de los datos que se guardarán y señala los bloques *establecer _variable_ para 0* y *cambiar _variable_ por* . Dado que se trata de cuestiones puramente operativas de la herramienta a trabajar, no se dedica especial esfuerzo en la exploración y experimentación, pero sí a la asociación de estos bloques con las discusiones previas y los distintos momentos de la dramatización.

Cuando los grupos necesiten **usar el valor de la variable** para mostrarlo en la pantalla, es probable que necesiten ayuda. El DR facilita la asociación, señalando que es un bloque que se usa igual que un número, y, por lo tanto, se puede encastrar en los mismos lugares en los que hasta ahora los estudiantes habían colocado bloques de números. (Una diferencia fundamental que puede comentarse es que no es posible saber, en el momento de la programación, cuánto vale ese número)



Variantes de complejidad

Funciones Si los estudiantes ya conocieran la noción de procedimiento (o función, subtarea, etc.), pueden definir una función (así se llaman en este entorno) que reúna el comportamiento de mostrar el resultado y asociarlo al botón A+B. De esta manera, el programa resulta mucho más legible. Se pueden modificar la manera en la que se informa el resultado (en lugar de una flecha que señale cada botón, puede agregarse texto, alguna animación, íconos, etc.). Al haber definido la función, será claro dónde deberán hacerse estas modificaciones y no interferirá con otras partes del programa.



Reiniciar Si los estudiantes ya trabajaron con variables, pueden hacer que al presionar A+B, además de mostrarse el resultado de los valores, se reinicie la

cuenta. También puede ser una oportunidad para definir una función que reinicie el valor de las variables.

3. Cierre

La importancia de las variables en la programación (10 minutos)

Subrayar la importancia de las variables en la programación a partir de señalar su existencia en programas que controlan dispositivos computacionales de uso frecuente.

¿El programa de adivinanza que hicieron almacenaba algún tipo de información de su ejecución? ¿Podíamos saber qué botones habían sido presionados antes? La adivinanza no almacenaba información durante su uso. El comportamiento dependía únicamente del último botón presionado y no de los anteriores: el mismo resultado se obtiene presionando una vez el botón B, o 10 veces el botón A y luego el botón B).

Para **generalizar** esta situación, dar lugar a que los estudiantes encuentren ejemplos de situaciones concretas en las que sea evidente que alguna computadora o programa está recordando un valor durante su ejecución.

Sólo si es necesario se pueden mencionar los siguiente ejemplos: el marcador de un partido recuerda los tantos acumulados de cada equipo, un medidor de glucosa guarda las últimas mediciones, un aire acondicionado recuerda la temperatura establecida por el usuario; un molinete electrónico recuerda cuántos ingresos se registraron hasta el momento; un mensajero instantáneo recuerda el mensaje escrito hasta el momento a medida que se completa con el teclado.

Historia para profundizar el concepto de variable

“Me gustaría contarles de qué manera organizo mis pertenencias. Como algunos de ustedes, en mi dormitorio tengo un armario con varios cajones. Los cajones son pequeños, por lo que en cada uno de ellos puedo guardar solo un objeto. En un cajón guardo solo una remera, en otro solo tengo un pantalón y así. Además, como mi memoria es muy mala, a cada cajón le pongo una etiqueta que describe lo que está guardado.

Al programar, disponemos de algo similar a un armario con cajones, y en cada uno de ellos podemos guardar diferentes datos o valores. Los cajones vendrían a ser nuestras variables. Imagínense, si necesito etiquetar mis cajones para saber qué guardo en cada uno, lo importante que puede ser para nosotros ponerle a cada variable un nombre que describa su propósito. En el caso del armario nos facilitará saber dónde guardamos la ropa; un buen nombre para nuestra variable nos facilitará leer nuestros programas y comprender qué estamos guardando o modificando” (Areces, et.al, 2018:194)¹

Bitácora y Foro

Se completa entre todos la bitácora del proyecto identificando el programa construido (contador de botones), el concepto computacional abordado (variables) y con ayuda del DA los conceptos curriculares o experiencias de Ed. Física vistos en la semana.



“¿En qué situaciones un programa está recordando información? ¿Cuál es esta información?”

Los estudiantes publican sus ejemplos en el foro.

Pueden adjuntar sus programas para que el DR pueda realizar un seguimiento.

Se invita a realizar la actividad interactiva para repasar la noción de variable.

La Yapa: Propuestas para seguir en casa

¿Y si nos equivocamos al contar? Piensen cómo se puede modificar el programa que construyeron para que al presionar un botón avance en uno la cuenta y al presionar el otro, retroceda.



¹ Areces, C., Benotti, L., Cortez, J. J., Fervari, R., García, E., Gómez, M., ... Wolovick, N. (2018). Ciencias de la Computación para el aula - Manual para docentes (segundo ciclo primaria). Fundación Sadosky, Program.ar, FAMA. Recuperado de <http://program.ar/manual-segundo-ciclo-primaria/>

ETAPA 3↓ Cuenta regresiva

En el aula, se recuperan experiencias en las que el deporte utiliza cuentas regresivas para establecer un límite de tiempo.

En la VC, se programa una cuenta regresiva para desactivar el contador de la Etapa 2 al llegar a cero y se motiva la alternativa condicional. Los estudiantes realizan una primera experiencia de análisis y adaptación de un programa existente para construir uno nuevo.

Objetivos

Se espera que los estudiantes sean capaces de:

- Reconocer la alternativa condicional como una herramienta que permite expresar que un programa se comporte de distintas maneras, según se encuentre en distintas situaciones.
- Reconocer aspectos que se pueden reutilizar para resolver un nuevo problema en un programa existente.
- Reinventar un programa, combinando herramientas y estrategias ya construidas para construir nuevos programas.
- Identificar las ventajas de dividir un problema en partes, a partir de su propia experiencia de programación.

Coordinación dupla pedagógica

Información que necesita tener el DR:

- El DR necesita conocer con qué familiaridad manejan los estudiantes los **operadores de comparación** ($>$, $<$, \leq , \neq , etc.) ya que serán necesarios durante la VC para escribir una condición y los **números negativos** (aparecerán en la pantalla de la placa). Si se considera necesario, se sugiere realizar una breve actividad de aula para realizar este relevamiento y repasar.

Decisiones del DA:

- Organización de la etapa: dado que abordar las actividades de las VC llevará más de una semana, las actividades sugeridas para el aula se pueden desarrollar entre las VC.
- Para ofrecer vivencias cercanas a los estudiantes en el uso de la cuenta regresiva, será valioso incorporarla en actividades motoras con el docente de Ed. Física y evaluar la posibilidad de que el DA, cuando establece tiempos para realizar tareas en clase, use en algún momento un reloj virtual de cuenta regresiva (<https://reloj-alarma.es/temporizador>). También puede considerarse el uso de la placa con la cuenta regresiva (incluso su primera versión) en el aula o en las clases de Ed. Física mientras se completan estas VC.

AULA ↓**Experiencias con cuentas regresivas****Propósitos mínimos**

-
- Alentar la identificación de situaciones donde se utilice la cuenta regresiva y experimentarlas.

Propósitos óptimos

-
- Propiciar una aproximación a los números negativos.

Notas para el DA ↓

Se recomienda hacer alguna actividad utilizando un temporizador o un juego que incluya el conteo para atrás de modo de vivenciar su función.

Es una oportunidad para involucrar al docente de Ed. Física, para identificar situaciones en la práctica deportiva donde se utilice la cuenta regresiva (por ejemplo, en el basquetball, tiempo de posesión de la pelota; en gimnasia, el inicio de las carreras; en los juegos por equipos, el tiempo para cambiar de rol).

Números negativos

En la VC de esta etapa se trabajará con números negativos. Quedará a consideración del DA utilizar esta necesidad como excusa para trabajar el contenido. Por ejemplo, a partir de temperaturas bajo cero, gráficas utilizando coordenadas, juegos de cartas, juegos de computadora, la posición de objetos en Scratch, situaciones de compra-venta, etc.

VC 3↓ Cuenta regresiva

🕒 Actividades previstas para requerir más de una VC de 45 minutos

💡 Desafío

Programar la micro:bit para que, además de contar la cantidad de saltos, muestre una cuenta regresiva en la pantalla y, al finalizar, informe la cantidad de saltos registrada.

1. Inicio

Recuperación y presentación del desafío (10 min)

Comienza la VC con un refinamiento de la actividad propuesta en el Inicio de la VC anterior, en este caso, la consigna será “cuántas veces saltas en 10 segundos”: un estudiante salta y otro cuenta los saltos mientras el DR muestra una cuenta regresiva en su pantalla y dice ¡Tiempo!

¿Qué diferencia hay entre esta consigna y la de contar saltos? ¿Podemos usar el programa de contar saltos para esta? ¿Es suficiente? ¿Se imaginan qué se le podría agregar? Los estudiantes deben identificar que esta nueva actividad es como la anterior, pero con un límite de tiempo. Pueden usar la micro:bit para contar, pero se requiere que el DR mire el reloj y diga basta. Los estudiantes deducen que la placa, automáticamente, debería indicar el límite de tiempo, para facilitar la medición y agregarle precisión.

A partir de allí se presenta el desafío de la etapa: programar la micro:bit para que, además de contar la cantidad de saltos, muestre una cuenta regresiva en la pantalla y, al finalizar, informe la cantidad de saltos registrada. Para alcanzar el desafío de la etapa, se plantean cuatro desafíos intermedios.

2. Desarrollo

Desafío 1. Programar una cuenta regresiva en la pantalla de las micro:bit (30 min)

El DR motiva a los estudiantes a establecer relaciones con el contador construido anteriormente, para construir entre todos la estrategia de programación.

¿Qué semejanzas hay entre el contador que construimos en la Etapa 2 y una cuenta regresiva? ¿Qué aparece en la pantalla de la placa en cada uno?

Los estudiantes identifican que una cuenta regresiva, es una forma particular de contar (“hacia atrás”). Por lo tanto, comparte con el programa de la etapa anterior la necesidad de almacenar información (cuántos conté hasta el momento, cuántos segundos quedan).

¿Cómo hacíamos para almacenar o recordar información?

Entonces, ambos programas necesitan una variable para “llevar la cuenta”, independientemente de si avanzan hacia adelante o hacia atrás.

¿Qué diferencias hay entre el contador y una cuenta regresiva? ¿Cómo cambian los números? ¿Cuál es el primer número que se muestra? ¿Qué es lo que hace que cambien los números?

Una cuenta regresiva difiere de un contador en:

- el valor inicial y el avance de la cantidad: la cuenta regresiva comienza en un valor máximo (10, por ejemplo) y se decrementa en 1;
- el contador cambia cuando el usuario presiona un botón, mientras que la cuenta regresiva se modifica cuando se cumple un intervalo regular de tiempo.

Trabajo en Makecode

¿Cómo saber cuántos segundos nos quedan? ¿Dónde guardamos la información de en qué número estamos? ¿Cómo hacemos para disminuir el valor de una variable? ¿Cuándo debemos modificar la variable?

Los estudiantes inician la exploración para resolver el Desafío 1. Mientras trabajan en sus computadoras, el DR realiza preguntas de chequeo para monitorear los avances y asistir en caso de necesidad:

¿Cómo sabe el usuario cuántos segundos le quedan? ¿Qué bloques podemos usar para eso?

Debe mostrarse en pantalla los segundos restantes, por ejemplo, mediante un bloque *mostrar número*.

¿Cómo sabe el programa cuántos segundos quedan o hasta qué número llegó?

Debe definirse una variable para llevar registro de “cuántos segundos quedan” o “en qué número estamos”. Su nombre debe reflejar esta función, por ejemplo: *segundos restantes*.

¿Cuál es el primer número que debe mostrar la cuenta regresiva? ¿En qué valor comienza la cuenta?

A diferencia del contador, en el que servía la inicialización automática de las variables en 0, en este caso, debemos **inicializarla explícitamente** en 10 (en el número en el que comienza la cuenta regresiva). Puede ser necesario para esto señalar la existencia del bloque *al iniciar* y el bloque *establecer _variable_ para*.

¿Podemos aprovechar el bloque *cambiar para restarle uno a la variable*?

Dependiendo del **conocimiento de los números negativos**, tal vez sea necesario explicitar que una forma de disminuir en uno una variable es utilizar el bloque *cambiar por -1*.

La cuenta regresiva involucra tiempo. ¿Qué herramientas de programación nos ofrece el entorno para trabajar con tiempo?

Es necesario **decrementar la cuenta cada un segundo**². En este caso, se señala

² El programa propuesto, además de detenerse por el bloque *pausa*, se detiene una breve cantidad de tiempo después de mostrar cada dígito (pues el bloque *mostrar número* así lo prevé). Esto hace que la cuenta regresiva demore más de diez segundos en llegar a 0. Esta imprecisión no representa un problema en esta instancia de la propuesta de

(o recuerda) que existe un bloque *para siempre* cuya función es ejecutar el contenido una y otra vez mientras la placa esté encendida. Además, debe conocerse el bloque *pausa (ms)* (que se encuentra en la segunda pestaña de la categoría *Básico*), para generar un programa que se detenga un segundo, decremente el contador y siempre repita esta acción. **Esta parte no es el tema central de la actividad y puede presentarse de manera guiada, dependiendo del tiempo disponible, el dominio del entorno, la familiaridad con el bloque *para siempre* y la noción de temporalidad en los programas.**

Una vez completo el desafío, los grupos verán que la cuenta regresiva funciona al principio, pero después de llegar a 0 sigue decreciendo. Nuevamente, si no están familiarizados con los números negativos, no podrán darse las precisiones matemáticas, pero basta con la intuición de que el bloque *para siempre* sigue trabajando. Cómo hacer para que esto no suceda da pie a la próxima parte.

Desafío 2. Detener la cuenta regresiva al llegar a cero (20 min)



Variante de complejidad

Si los estudiantes ya están familiarizados con la noción de alternativa condicional, puede aprovecharse este momento para que ellos mismos propongan la solución al desafío. Se esboza en conjunto un modelo de solución y se avanza directamente a la instancia de programación.

Este desafío busca motivar e introducir la **alternativa condicional**, a partir del problema: la cuenta regresiva no se detiene.

¿Siempre debemos restar uno a la variable contadora? ¿Cuándo sí y cuándo no?

El DR modela las respuestas para que los estudiantes describan las situaciones utilizando la palabra SI (Ej: “SI la variable ya vale cero, no debemos restar”).

trabajo, sin embargo, si se quisiera una medición precisa puede utilizarse el bloque *plot bar graph of*, en la categoría *LED* (no se visualiza el número, pero la cuenta regresiva se expresa como los LEDs de la pantalla apagándose sucesivamente).

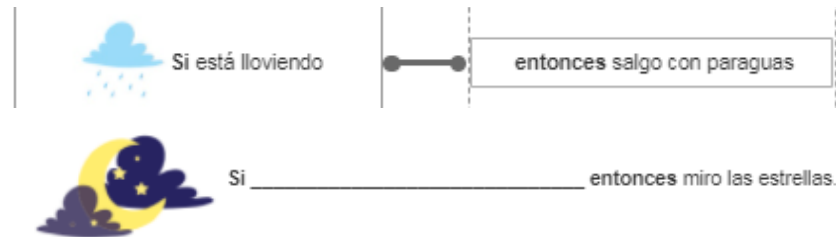
Deben identificar, también, que a veces no debe ejecutarse el bloque que decrementa la variable.

Queremos que a veces no se ejecute el bloque cambiar por -1. ¿Pero podemos hacer que no siempre se ejecuten los mismos bloques en nuestro programa?

La conclusión de este intercambio apunta a identificar la necesidad de una nueva herramienta que permita cambiar el comportamiento del programa (o qué bloques se ejecutan) durante su ejecución. Se presenta la **alternativa condicional** como una **herramienta de programación** que permite ejecutar o no ciertos bloques según si se cumple o no una condición. Para fortalecer esta definición, se propone a los estudiantes buscar ejemplos de la vida cotidiana en la que exista un comportamiento o acción atada a una condición.

Recursos: para profundizar sobre alternativa condicional

- El video [P Ctubers 6 - Condicionales](#) El desafío del huevo.
- La actividad interactiva que remite a ejemplos de la vida cotidiana expresados como alternativas condicionales.³



Trabajo en MakeCode: elaboración de la condición y agregado del bloque SI

Se retoma la discusión sobre la cuenta regresiva, ahora para expresarla con esta nueva estructura (por ejemplo, se puede decir “SI la variable es mayor a 0, cambiar la variable por -1”) y se propone a los estudiantes incorporarla a sus programas. Se señalan los bloques para expresar alternativas y comparaciones y se invita a explorarlos para detener la cuenta regresiva. Si fuera la primera vez

³ Adaptación de la actividad del Manual para docentes “[Ciencias de la Computación para el Aula, 1er Ciclo de Primaria](#)” (pág 167) disponible en <http://program.ar/manual-primer-ciclo-primaria>.

que los utilizan, puede ser necesaria una guía más intensa para combinar adecuadamente el bloque de la condición, la alternativa y los bloques del interior.

Atención

Según lo establecido en la coordinación, el DR debe motivar una solución que utilice los operadores de comparación con los que los estudiantes estén más familiarizados. Será valiosa la ayuda del DA para orientar a los grupos.

Al finalizar este desafío, los grupos cuentan con una cuenta regresiva que pueden usar para otras actividades escolares, en particular, de Ed. Física.

Desafío 3. Agregamos la capacidad de contar saltos con los botones. (15 min)

Se repite la experiencia “cuántas veces saltas en 10 segundos”, pero ahora se usa una micro:bit con el programa de la cuenta regresiva y otra con el contador de saltos.

¿Hacen falta dos placas? ¿No podremos usar la misma para mostrar el tiempo y contar? Con estas preguntas, queda planteado el desafío. Parte fundamental de la tarea de programación será el análisis de los programas previamente confeccionados y la identificación de algunos de sus aspectos claves, ya sea para reproducirlos, extraerlos o adaptarlos para construir los nuevos programas.

Atención: metodología de trabajo

La estrategia propuesta para este desafío es una introducción a la forma en la que se trabajará en las próximas VC: **obtener el nuevo programa a partir de analizar y adaptar los anteriores**. Se sugiere trabajar siempre sobre copias de los programas a adaptar o modificar para conservar una versión de los programas originales que funcione correctamente.

¿Hace falta empezar de cero? ¿Dónde pueden encontrar los programas o los bloques que necesitamos?

Para incorporar al programa de la cuenta regresiva los bloques del contador construido en la etapa anterior, los grupos deben revisar sus programas e identificar los bloques necesarios para esta tarea (los eventos de los botones, pero también la variable que cuenta los saltos).

Atención

No es posible copiar bloques de una ventana de MakeCode a otra, ni es fluido el trabajo con MakeCode en más de una pestaña del navegador al mismo tiempo. Se sugiere que la identificación de qué bloques hay que agregar al programa se haga entre todos y que el DR deje visible en su pantalla los bloques a agregar para que los estudiantes los reproduzcan.

Desafío 4. Los botones dejan de contar una vez que se agota el tiempo. (20 min)

Se repite la experiencia “cuántas veces saltas en 10 segundos”, para que los estudiantes descubran que si siguen presionando los botones, la placa sigue contando aún cuando la cuenta regresiva terminó. Se enfrenta a los estudiantes a este problema:

¿Está bien que la placa siga contando saltos aunque la cuenta regresiva haya terminado? ¿A qué se debe el problema? ¿Cómo se imaginan que se puede resolver? ¿Qué problema similar surgió en esta etapa?

Al igual que con los bloques de la cuenta regresiva que modificaban el número en pantalla, los bloques que modifican la cuenta de los números, deben dejar de ejecutarse en un momento dado.

¿Cómo lo resolvimos? ¿Por qué funcionó esa solución? ¿Qué debe suceder cuando apretamos un botón? ¿Siempre? ¿Cuándo sí y cuándo no?

La alternativa condicional es una solución posible, ya que permite cambiar el comportamiento de un programa durante su ejecución. Y la condición para esta alternativa debe expresar “Si no se terminó el tiempo”, para lo que se puede recordar que se definió una variable especialmente para llevar esta cuenta.

Luego de estas observaciones, se avanza con la tarea de programación.

Atención

La intervención docente más o menos específica en estos razonamientos estará determinada por las respuestas que vaya obteniendo de los estudiantes.

Variantes de complejidad

Reutilización de soluciones Para que no se mezclen los números en la pantalla puede hacerse que el botón A+B esté desactivado mientras se está ejecutando la cuenta regresiva. Esta es una buena oportunidad para que los estudiantes construyan desde cero una solución a un problema muy similar al que acaban de resolver en la VC con la guía del DR.

Reiniciar Al presionar A+B, además de mostrarse la cantidad de saltos, se reinician los contadores y la cuenta regresiva. Pueden establecer analogías con los bloques para reiniciar la cuenta en la etapa anterior.

Alternativa condicional si no quienes estén familiarizados con la alternativa condicional con su rama “si no”, puede hacerse que, si la cuenta regresiva no es cero, la pantalla muestre el número de segundos restantes, mientras que si ya se agotó el tiempo, informe el resultado de la cuenta (la cantidad de clics registrados sobre cada botón y cuál fue mayor). En este caso, puede dedicarse el botón A+B exclusivamente para reiniciar la cuenta. En el caso de haber definido funciones propias, puede evaluarse cómo se pueden aprovechar para realizar estas modificaciones más fácilmente.

3. Cierre (10 minutos)

Reforzar el concepto de alternativa condicional, poniéndolo de manifiesto en situaciones computacionales cotidianas. Explicitar la práctica de reutilización y adaptación de programas.

¿Con qué problema nos enfrentamos en la primera versión de la cuenta regresiva? ¿Qué herramienta de programación usamos para resolverlo?

Los estudiantes recuperan el uso de la alternativa condicional. Para generalizar esta nueva noción se pide a los estudiantes que piensen ejemplos de usos de computadoras en su vida cotidiana en los que pueda haber una alternativa condicional involucrada. Por ejemplo: "SI la contraseña es válida, permitir el ingreso". Es importante que los estudiantes puedan concluir que la **alternativa condicional es una manera de permitir que un programa tenga más de un comportamiento, en base a si sucede o no una condición.**

¿Es parecido el programa del contador al de la cuenta regresiva? ¿Por qué? ¿Cómo lo aprovechamos en la etapa de programación?

Como el problema de contar y de hacer una cuenta regresiva son parecidos, cuando se definió la estrategia de programación para la cuenta regresiva, fue de utilidad analizar el programa del contador.

¿Fue muy distinto programar el contador en esta etapa que en la anterior? ¿Por qué? ¿Son parecidos los problemas? ¿Y los programas? ¿A qué otros programa se parece el resultado de esta etapa? ¿Por qué?

Para el último programa se reutilizó uno previo: se agregó el comportamiento del contador (tomándolo de otro programa) y, luego, se combinó con la cuenta regresiva para que se desactiven los botones al finalizar el tiempo. **Esta es una práctica muy usual durante la programación: tomar algo existente, modificarlo y combinarlo con otras cosas para construir un comportamiento diferente.**

¿Les pareció fácil cada desafío? ¿Se imaginan otra forma para conseguir el desafío general? ¿Habría sido más fácil o más difícil resolverlo sin las etapas intermedias?

Se recupera de la experiencia que con esta manera de trabajo, se acotan los problemas, pues **en cada paso se resuelve un problema pequeño** que se basa en otros que ya fueron resueltos completamente. Es más efectivo resolver problemas de a uno, aún cuando el resultado siga teniendo problemas, que atacar un problema grande de entrada, cuya solución puede ser tan compleja que no podamos resolverla ni parcialmente.

Foro y bitácora

Se completa entre todos el mapa conceptual identificando el programa construido (cuenta regresiva), el concepto computacional abordado (alternativa condicional) y con ayuda del DA los conceptos curriculares o experiencias de Ed. Física vistos en la semana.

Los estudiantes pueden publicar sus reflexiones en el foro de evidencias y adjuntar sus programas para que el DR pueda realizar un seguimiento.

Se invita a realizar la actividad interactiva para repasar la noción de alternativas.



La Yapa: Propuestas para seguir en casa

¿Qué otra cosa vieron que sucede cuando se acaban las cuentas regresivas? Puedes agregar alguna animación que ilustre que se terminó el tiempo y, además, informar automáticamente cuántos saltos se hicieron (sin tener que apretar A+B).



ETAPA 4↓ Sensores

En esta etapa se introducen los desafíos deportivos que consisten en repeticiones de ejercicios.

En el aula los estudiantes recuperan estas situaciones de su experiencia o práctica en la clase de Ed. Física.

En la VC se trabaja continuando la idea de usar programas existentes para construir otros con comportamientos similares. Se adapta el dispositivo contador de saltos para que detecte los saltos automáticamente utilizando el sensor de movimiento.

En la reflexión final, se explicitan los pasos para la reutilización de programas como una manera de organizar la elaboración de soluciones computacionales.

Objetivos

Se espera que los estudiantes sean capaces de:

- Asociar los sensores con la entrada de información a la micro:bit.
- Describir la solución a un problema en términos de soluciones existentes para otros problemas.
- Reconocer una secuencia organizada de pasos para la reutilización de programas al momento de resolver problemas computacionales.

Coordinación dupla pedagógica

Información que necesita tener el DR:

- Consulta al DA del avance en su proyecto áulico para contextualizar la VC.
- Breve intercambio a propósito de cómo resultaron las clases en la Etapa 3.

Decisiones del DA:

- Organización de la etapa: la actividad sugerida en el aula es el disparador para las actividades de las VC, pero dado que estas llevarán más de una semana, el trabajo con el eje vinculado a la salud será una buena forma para el DA de acompañar la propuesta durante las siguientes clases de aula.

Decisiones conjuntas entre DA y DR:

- Esta etapa incluye una VC opcional, en la que se construye un cronómetro y, además, se adapta para medir automáticamente el tiempo requerido en completar una rutina de 10 saltos. DA y DR deben decidir si incluirla o no, teniendo en cuenta la pertinencia de este tipo de mediciones en los dispositivos diseñados por los estudiantes, sus habilidades de programación y de diseño de estrategias más complejas.

AULA ↓

Prácticas físicas repetitivas

Notas para el DA ↓



Propósitos mínimos

- Brindar espacio para identificar las diferentes situaciones de la práctica física en las que se cuentan movimientos repetitivos.

Propósitos óptimos

- Brindar espacio para vivenciar las diferentes situaciones de la práctica física en las que se cuentan movimientos repetitivos o velocidad.
- Propiciar la utilización del dispositivo en eventos de clase con el DA o con el profesor de Educación Física.
- Abordar el proyecto de aula y articularlo con los avances realizados en los dispositivos de medición.
- Repasar las unidades de medida del tiempo: milisegundos y microsegundos (en caso que se construya el cronómetro).

Durante esta etapa, en las VC se irán construyendo programas que permitirán registrar movimientos. En la medida en que los estudiantes estén familiarizados con las prácticas físicas asociadas a estas mediciones (ejercicios repetitivos) más sentido encontrarán a la construcción de sus dispositivos. Se sugiere utilizar las placas en el aula y las clases de Ed. Física a medida que los van construyendo. Estas experiencias de uso, pueden ser una oportunidad para encontrarse con errores o con ideas para mejorar sus programas.

En caso que se decida abordar el desarrollo del cronómetro (VC opcional), será valioso trabajar con los estudiantes la distinción entre contar el tiempo que lleva una determinada cantidad de movimientos (ej: llegar una meta, completar un ejercicio) y contar cuántos movimientos se realizan en una determinada cantidad de tiempo (abdominales, pasos, saltos). Para ello, con la ayuda del docente de Ed. Física, se sugiere enfrentar a los estudiantes con la necesidad de contar movimientos y tiempo, de forma independiente o combinada.

VC 4↓ Sensores y sus eventos

🕒 Actividades previstas para requerir más de una VC de 45 min

💡 Desafío

Programar una placa para que cuente saltos automáticamente. Y hacer que avise cuando se completaron 10 saltos.

1. Inicio

Recuperación y presentación del desafío (10 min)

La motivación de esta actividad es continuar con la automatización del dispositivo contador de saltos para liberar al usuario de la responsabilidad de presionar el botón cada vez que se produce un salto. Luego de una breve reflexión sobre ello, el DR introduce el desafío de la clase: programar una placa para que cuente saltos automáticamente. y que avise cuando se completaron 10 saltos. La presentación se puede ilustrar con la primera parte del siguiente video (omitiendo la parte de programación).

▶ JUEGOS CON MICRO:BIT - VINCHA



<https://youtu.be/zNquRh4b5eY> hasta 1:22 min (previo a la programación)

2. Desarrollo

Desafío 1: Contar saltos automáticamente (30 min)

El DR habilita una instancia de **planificación**, en la que los grupos continúan y profundizan la práctica de reutilizar y reinventar.

¿A qué problema que ya resolvimos se parece el que tenemos que resolver ahora? ¿Qué cambia con respecto al que ya está hecho?

Una buena manera de empezar a resolver un problema es identificar problemas similares que ya se hayan resuelto. En este caso, los estudiantes deben descubrir que contar saltos automáticamente es muy similar a contarlos con el botón y que la diferencia está en el motivo que hace avanzar el contador (la placa detecta el salto vs el usuario presiona el botón).

¿Qué programa tenemos para contar? ¿Qué tan distinto será el programa que cuenta saltos del programa que cuenta con los botones? ¿Qué parte de ese programa es la encargada de contar?

Habiendo identificado el problema que servirá de base, se procede a analizar su solución: el programa. Los estudiantes observan el programa que construyeron y especulan sobre cómo debería ser el nuevo. Para esto, deben identificar qué partes del programa que ya construyeron son responsables del comportamiento que se quiere conservar (contar) y cuáles del que se quiere modificar (responder al botón).

Trabajo en MakeCode

¿Qué utilizamos para contar con los botones? ¿Será muy diferente contar saltos automáticamente?

Los estudiantes deben identificar que deben conservar la variable contadora pero modificar el evento *al presionar botón*. Esto motiva la exploración del entorno en busca de un nuevo bloque que proporcione información de saltos. Si los grupos están familiarizados con la noción de evento, puede usarse como pista para orientar la exploración.

¿Qué herramientas usamos hasta ahora para detectar que debíamos incrementar la cuenta? ¿Y para detectar saltos? ¿Serán muy diferentes?

Si fuera necesario, para acompañar la exploración del entorno en busca del evento de agitación de la placa, el DR retoma la conceptualización de computadora construida en la Etapa 1, para precisar que tanto en el caso del uso de los botones, como el sensor de movimiento existe **información de entrada** a la micro:bit. En el programa anterior, el bloque de evento *al presionarse el botón A* era el responsable de detectar esta entrada (el clic sobre el botón) e incrementar la variable.

En el nuevo programa, *¿cuál será la información de entrada? ¿Qué bloques de entrada ofrece el entorno? ¿cuál puede servir para este desafío? ¿Qué tipo de bloques son? ¿Qué nos permiten expresar? ¿Es muy diferente la forma en que se usa cada uno?*

Los estudiantes ingresan a la pestaña *Entrada* e identifican el bloque *si agitado* como candidato. Es importante que identifiquen que es un bloque de evento (permite establecer la respuesta de la placa ante un estímulo externo) y, por lo tanto, se utiliza de la misma manera que *al presionarse el botón A*. **También es importante que asocien este sensor (y los otros de la pestaña) con el ingreso de información a la micro:bit.**

Atención: problema de traducción

Puede ser necesario aclarar que, debido a un problema de traducción, el bloque que responde al evento de agitar la placa se llama "si agitado" y no "al agitarse", como corresponde para continuar en la línea de los eventos de los botones ("al presionarse el botón..."). Se puede aprovechar esta aclaración para señalar que si este entorno hubiera sido programado por personas hispanohablantes, este error no habría ocurrido.

Atención: sensibilidad del sensor

Anticipar a los estudiantes que el sensor no es "muy sensible" y, por lo tanto, pueden ocurrir algunos errores menores en la cuenta, que se deben al funcionamiento y anclaje de la micro:bit y no a la programación.

Variante de complejidad

Estrategias más complejas Los cursos que tengan más experiencia en programación y en diseñar soluciones con cierta complejidad, pueden abordar el problema solos, simplemente con la sugerencia de pensar cómo modificar el programa anterior y alguna pista hacia la sección *Entrada*. En este caso, es importante hacer una puesta en común de las estrategias de programación propuestas, para compararlas y ajustarlas de ser necesario.

Desafío 2: Programar la micro:bit para que nos avise cuando completamos una rutina de diez saltos (30 min)

Para este desafío, se puede proponer una nueva dramatización, con la consigna: "Saltar 10 veces". Para contar los saltos se usa el programa que los cuenta automáticamente. Además, debe haber otro estudiante como supervisor para avisar cuándo el compañero llegó a diez.

Se plantea entonces el desafío 2 y se habilita una primera instancia de análisis del problema.

¿Cómo podemos automatizar el aviso sin necesidad de que un compañero detecte cuando se llegó al final del ejercicio? ¿Qué tiene que hacer el nuevo programa? ¿Se parece a alguno/s que ya hicimos? ¿A cuál/es? ¿Qué podemos aprovechar de cada uno?

Tanto el programa para contar saltos automáticamente como la cuenta regresiva son útiles. En caso de que los estudiantes identifiquen solo el primero, se avanza en la precisión del análisis del programa, para identificar que este problema es similar al de detectar cuándo la cuenta regresiva alcanzaba el 0 (resuelto en el segundo desafío de la Etapa 3).

En conclusión: el nuevo programa debe **combinar** un contador de saltos con un mecanismo para detectar que se ha alcanzado un límite, similar al de la cuenta regresiva.

Importante: intervención de los docentes

Esta es la **primera vez que los grupos programan un dispositivo complejo**, similar a los que van a diseñar y programar para su circuito personalizado. Por este motivo son **fundamentales tanto el espacio de exploración autónoma como la guía del DR para encauzar los esfuerzos.**

El próximo paso implica tomar ciertas partes de estos programas y adaptarlos y combinarlos para lograr el programa esperado. Para eso, el DR comparte un programa de cuenta regresiva y otro de contador de saltos construidos por los estudiantes para **identificar** entre todos, las partes responsables de los comportamientos deseados.

En el contador de saltos: ¿Qué partes del programa se encargan de contar saltos?

Los estudiantes identifican la variable contadora y el evento de agitación.

En la cuenta regresiva: ¿Qué herramienta de programación usamos para detectar el límite? ¿Cómo la usamos?

Los estudiantes **rescatan el uso de la alternativa condicional** sobre la cantidad de saltos (o segundos).

Conociendo las herramientas de las que se dispone, se procede a **adaptar el programa contador de saltos**. Dependiendo de las habilidades de cada grupo para leer, interpretar y abstraer código, la guía del docente será más o menos intensa durante la programación.

Recordemos: ¿qué modificación queremos agregar? El aviso al alcanzar 10 saltos.

¿Qué herramienta vamos a utilizar para detectarlo? ¿Cómo la usábamos en la cuenta regresiva? ¿Qué cambia en este caso? En la cuenta regresiva se utilizaba una **alternativa condicional** en cuya **condición** se comparaban los **segundos restantes con 0** para detectar que se había agotado la cuenta. En este caso, hay que **comparar los saltos realizados con 10** para detectar que se ha alcanzado el objetivo de la rutina.

¿Dónde vamos a incorporar la modificación? ¿En qué momento de la rutina (o la dramatización) debe identificarse que se cumplió el objetivo y mostrar el aviso? ¿Con qué parte del programa se corresponde este momento? Después de cada salto, es momento de preguntarse si se ha alcanzado el objetivo, para que en caso de que sí, se muestre un aviso y el usuario no siga saltando. Esto sucede al final del bloque del evento de agitación.



Variante de complejidad

Más información Los grupos que estén más familiarizados con la alternativa condicional pueden preocuparse por mostrar la cantidad de saltos realizados hasta el momento o, incluso, mostrar la cantidad de saltos restantes en la pantalla. Una dificultad de este desafío recae en que no cambie el número en la pantalla una vez que se alcanzó la cantidad de saltos deseada y se sigue agitando la placa.

3. Cierre (10 minutos)

Los docentes acompañan a los grupos para que **identifiquen en su propia experiencia de programación cómo llevaron a cabo la combinación y adaptación de programas existentes**.

¿A qué programa de los que hicimos dirían que se parece el contador automático de saltos? ¿Por qué? ¿Y el que nos avisa cuando saltamos 10 veces? ¿Nos ayudó a programar, haber identificado estos parecidos? ¿De qué manera?

Programas aparentemente distintos, pueden tener un comportamiento similar y ser capaces de identificar estas similitudes sirve de punto de partida para el diseño del programa final.

¿Un programa siempre tiene una única función? ¿Cuántas cosas hace el programa de los diez saltos? ¿Podemos identificar qué parte del programa hace cada una? ¿Por qué es importante poder hacer esta identificación?

Un mismo programa puede combinar comportamientos y se puede identificar qué sector del programa es responsable de cada uno. Establecer esta identificación con facilidad nos sirve para analizar mejor los programas. Por ejemplo: si queremos modificar alguna parte de su comportamiento, es necesario identificar los bloques que son responsables de ella. También es necesario si queremos reutilizar alguna parte del programa para resolver un nuevo problema.

El DR suma a la discusión que **adaptar y reutilizar** programas es sumamente frecuente en la programación, ya que es una manera de evitar resolver de nuevo problemas que ya están resueltos y utilizar estas soluciones para construir programas que resuelvan problemas nuevos. Para finalizar la reflexión, se elabora una lista a modo de resumen de cuatro pasos fundamentales para la reutilización y adaptación de programas.

- 1) *¿Qué fue lo primero que hicimos?* **Describir** el comportamiento del nuevo programa. Es fundamental tener muy claro *qué* queremos construir.
- 2) *Una vez que establecimos el comportamiento que queríamos, ¿qué buscamos en él?* **Identificamos** comportamientos conocidos.
- 3) *¿Cómo llegamos a los otros programas? ¿Qué cosas buscábamos?* **Buscamos programas** conocidos con comportamientos parecidos al que necesitábamos.
- 4) *Una vez que terminamos el análisis y la exploración, ¿en qué consistió la programación?* Finalmente, tuvimos que **adaptar** algunas partes de los programas viejos para el nuevo programa.

La VC termina adelantando a los estudiantes que trabajarán con esta dinámica para el diseño y programación de los dispositivos que elijan para su circuito.

La Yapa: Propuestas para seguir en casa

¡Ya tienen un asistente deportivo computacional! ¿Cómo se puede cambiar la cantidad de saltos que hay que hacer? ¿Hace falta cambiar todo el programa? ¿Se puede elegir al comienzo?

¿Y si queremos repetir el ejercicio? ¿Cómo se puede hacer para que no haga falta apagar y volver a encender la placa? ¿Cómo se puede adaptar a distintos ejercicios motrices? ¿De qué otras maneras nos puede avisar que ya cumplimos el objetivo?



Bitácora y Foro

Se completa entre todos el mapa conceptual con los programas, conceptos y prácticas puestos en juego. Se incluyen los pasos para la reutilización de programas para que los estudiantes ejemplifiquen cada uno de los pasos con sus experiencias durante esta etapa.

- 1) Describir el comportamiento del nuevo programa.
- 2) Identificar comportamientos conocidos.
- 3) Buscar otros programas que contengan estos comportamientos.
- 4) Adaptar al nuevo programa.

Los estudiantes pueden publicar sus reflexiones en el foro de evidencias y adjuntar sus programas para que el DR pueda realizar un seguimiento.

Se invita a realizar la actividad interactiva para repasar la noción de alternativas.



VC Opcional↓ Temporización

🕒 Actividades previstas para requerir más de una VC de 45 min

💡 Desafío

Programar la micro:bit para medir tiempos. Primero, programar un cronómetro (el botón A inicia la medición de tiempo y el botón B la detiene). Luego, programar la placa para medir cuánto demora una persona en saltar 10 veces.

1. Inicio

Recuperación y presentación del desafío (10 min)

- *¿En qué situaciones, al hacer ejercicio, medimos el tiempo?*
Medir el tiempo transcurrido entre el inicio y el fin de un ejercicio es una necesidad frecuente en las prácticas motrices. Por ejemplo: cuánto tardas en saltar 10 veces con 2 pies, con 1 pie, en hacer 5 sentadillas, en dar una vuelta corriendo al patio.
- *¿Nos puede asistir la micro:bit para medir el tiempo de inicio y fin de un ejercicio motriz? ¿Tiene Makecode bloques para trabajar con tiempo? ¿En qué categoría estarán?*
Los estudiantes comparten sus ideas y hallazgos. El DR orienta a los grupos señalando que la información de tiempo es información de entrada, y explica que los bloques *tiempo de ejecución*, en la categoría *Entrada* permiten conocer, al momento de ser ejecutados en el programa, cuánto tiempo transcurrió desde que se encendió la placa (ya sea en microsegundos -micros- o milisegundos -ms-). Se puede pensar como un reloj que comienza en 00.00 cada vez que se enciende la placa y al que se le puede preguntar qué hora es. Esta hora no está expresada en horas y minutos sino como la cantidad de milisegundos (o microsegundos) transcurridos desde que se encendió la placa. A esto se le dice *marca de tiempo* (o *timestamp* en inglés).

Luego de la puesta en común, se presenta el desafío: programar la micro:bit para medir tiempo. Dicho desafío se divide en dos partes: la programación de un cronómetro (el botón A inicia la medición de tiempo y el botón B la detiene) y la programación para medir cuánto demora una persona en saltar 10 veces.

2. Desarrollo

Desafío 1. Cronómetro (30 min)

- *¿Cómo se puede utilizar una marca de tiempo para medir tiempo transcurrido? ¿Siempre que medimos tiempos usamos un cronómetro o también podemos usar un reloj? ¿Si volvemos del recreo 10.45 y habíamos salido 10.20, podemos saber cuánto duró el recreo?*
Como conclusión de esta discusión, se explicita la estrategia: calcular la diferencia (mediante una resta) entre la hora del momento del final (expresada como horas y minutos, en el caso del recreo, y como una marca de tiempo en el caso de la micro:bit) y la hora del momento del inicio. Se puede esquematizar una línea de tiempo, que explicita la temporalidad con la que se suceden los eventos, las marcas de tiempo de cada uno y cómo se pueden aprovechar para calcular tiempo transcurrido.
- *¿Qué debe suceder cuando presionamos cada uno de los dos botones?*
Se propone a los estudiantes que pongan por escrito la estrategia.

Cuando clic en A:

- tomar marca de tiempo de inicio

Cuando clic en B:

- tomar marca de tiempo de fin
- informar la diferencia entre las marcas de tiempo

- *¿Cómo hacemos para programar esta estrategia? ¿Qué información necesitamos a la hora de informar el resultado? ¿Está disponible todavía? ¿Qué herramienta de programación usamos para almacenar información? ¿Servirá para guardar marcas de tiempo? ¿Por qué?*
Los grupos deben identificar que necesitan dos variables para almacenar las marcas de tiempo. Se recupera la conceptualización de las variables

como herramientas para almacenar información. Si bien hasta ahora las habían utilizado para almacenar cantidades, dado que las marcas de tiempo también son números, también pueden utilizarlas.

Importante

Medir tiempo transcurrido utilizando una marca de tiempo es uno de los **primeros problemas más complejos a los que se enfrentan los estudiantes**. Por lo tanto, es crucial que exista el **espacio en la VC para que puedan pensar y proponer soluciones**. Además, la necesidad de una variable para almacenar la marca inicial es un ejemplo de **aplicación de un concepto ya trabajado para resolver un problema computacional distinto** para el que fue presentado.

Construir soluciones y aplicar las herramientas y conceptos de programación conocidos en contextos novedosos es uno de los objetivos fundamentales del aprendizaje de la programación y, por lo tanto, constituye el núcleo de esta actividad.

Atención

Es probable que sea la primera vez que los grupos necesiten usar un **bloque de operaciones aritméticas** para calcular la resta. Si fuera necesario, el DR explicará cómo utilizarlos, ya que la dinámica de encastrado puede no resultar intuitiva y no representa el foco de la actividad.

También puede proponer **dividir el valor de la diferencia por 1000 para expresar el resultado en segundos** en vez de en milisegundos.

Cuando los grupos finalizan la programación, se retoma la experiencia para que los estudiantes identifiquen algunos puntos clave:

¿Habían resuelto hasta ahora algún problema parecido a este? ¿Les sirvió lo que ya aprendieron? ¿Por qué?

El uso de herramientas conocidas para nuevos problemas. Utilizaron los

eventos sobre los botones, pero no para contar sino para marcar el inicio y el final de un período de tiempo. También utilizaron variables para almacenar información, no con cantidades de un conteo sino con información temporal.

¿Pudieron medir tiempo transcurrido aun cuando no encontraron un bloque que sirva por sí solo para eso? ¿Qué pasó cuando quisieron resolver un problema para el que no hay exactamente un bloque que lo resuelva?

La construcción de una solución a partir de un **uso ingenioso de una herramienta específica**: A diferencia de los desafíos anteriores que estaban más asociados a los bloques provistos, en el entorno de programación no existe ningún bloque que se encargue exactamente de la medición de tiempo. Sin embargo, a pesar de no contar con un bloque “tiempo transcurrido entre”, aprovecharon las marcas de tiempo provistas por la placa para resolverlo. Para esto, fue necesario plantear una **estrategia** para conseguir este comportamiento combinando los bloques existentes de una manera particular.

Desafío 2. ¿Cuánto demoras en saltar 10 veces? (30 min)

Se plantea el nuevo desafío como un avance en la automatización. Para motivarlo, puede realizarse el ejercicio con dos estudiantes: alguien que salta y tiene la placa programada como en la Etapa 4, que detecta cuándo se cumplen 10 saltos, y alguien que utiliza el cronómetro del desafío anterior para medir el tiempo.

Atención

Este desafío requiere poner en práctica, de manera más compleja, **la secuencia de pasos para la reutilización de programas que se trabajó en la etapa anterior**:

- 1) Describir el comportamiento del nuevo programa.
- 2) Identificar comportamientos conocidos.
- 3) Buscar otros programas que contengan estos comportamientos.
- 4) Adaptarlos para el nuevo programa.

Es importante explicitar estos pasos durante la resolución del desafío, para que los grupos se familiaricen con **la estrategia que utilizarán en las clases posteriores para diseñar y programar sus propios dispositivos**.

- *¿Qué debe hacer el nuevo dispositivo? ¿Cuántas cosas dirían ustedes que tiene que hacer? ¿Son cosas nuevas?*

El primer paso para programar algo complejo es comprender y describir el comportamiento que queremos conseguir, en base a otros comportamientos, que resulten más simples de comprender (ya sea porque son más simples o porque son más familiares). Este nuevo programa combina dos comportamientos: medir tiempo y contar saltos (hasta 10) y ya se resolvieron estos problemas por separado. Con esto se completa la instancia de **identificar comportamientos conocidos**.

- *¿Qué programas hicieron para resolver estos problemas? ¿Qué bloques usaba cada uno?*

Este es el segundo momento en la construcción de la solución: **buscar en otros programas** ya realizados comportamientos similares a los deseados. Los estudiantes deben descubrir que los programas utilizados en la dramatización inicial (el cronómetro del desafío anterior y el contador de 10 saltos de la Etapa 4), les servirán de referencia para adaptarlos al nuevo programa.

- *Pensando en el resultado que queremos: ¿qué nos sirve de cada uno de esos programas? ¿Qué adaptaciones habrá que hacer? ¿Qué aspectos de cada uno queremos conservar? ¿Cuáles queremos modificar?*

Los grupos deben identificar que como el problema consiste en **medir tiempo transcurrido** necesitarán **utilizar marcas de tiempo con una estrategia similar** a la utilizada en el cronómetro. Además, deben utilizar el contador de saltos y la alternativa condicional para detectar cuando se alcanzaron los 10 saltos, con una estrategia similar a la de la etapa anterior.

Se pueden recuperar y sintetizar estas estrategias, con especial atención a las marcas de tiempo, **señalando particularmente los eventos que dan inicio y fin** al intervalo que se quiere medir (en el caso anterior: los clics sobre cada uno de los botones) y **la necesidad de variables** para almacenar las marcas. Pensando en las **adaptaciones al nuevo programa**, se espera que los grupos identifiquen que en este caso (en comparación con el cronómetro) los eventos que dan inicio y fin a la medición de tiempo son distintos: si bien el inicio es un clic en un botón, **el final debe producirse automáticamente cuando se alcanza la cantidad de saltos requerida**.

Trabajo en MakeCode

A partir de la discusión anterior, los grupos avanzan a la programación en

MakeCode. Dependiendo de sus habilidades para leer, interpretar y abstraer código, la guía del docente puede ser más o menos intensa durante esta etapa. Las siguientes preguntas pueden ayudar a los grupos a identificar algunos puntos claves de la solución.

- *¿Conviene comenzar un programa nuevo o trabajar sobre alguno existente? ¿Cuál? ¿Por qué?*

Es conveniente trabajar sobre una **copia** de alguno de los dos programas elegidos como base. En principio no hay un mejor candidato para esto. Los grupos pueden ponerse de acuerdo en cuál van a usar, para que el DR deje en su pantalla el otro programa, para que puedan consultarlo durante la adaptación.

- *¿Qué información necesita almacenar este programa? ¿Con qué herramientas lo hará?*

El programa resultado debe almacenar la cantidad de saltos, y las marcas de tiempo inicial y final. Para cada una de estas cosas, deberá existir una variable. Es importante insistir en el nombre que elijan los grupos para cada una de ellas.

- *¿Cuándo empieza la medición de tiempo? ¿Cómo se inicia una medición de tiempo con las herramientas que conocemos? ¿Qué programa podemos mirar para ayudarnos? ¿Qué bloques usamos en ese programa para iniciar la medición?*

La medición comienza al presionar un botón. Luego, en el evento correspondiente debe tomarse la marca de tiempo inicial y almacenarse en la variable, tal como se hizo en el programa del cronómetro.

- *¿Cuándo debe detenerse la medición de tiempo? ¿Cómo sabemos que debemos detener la medición? ¿Qué programa podemos mirar para ayudarnos?*

La medición debe detenerse cuando se alcanzan los 10 saltos. En el programa que contaba los 10 saltos, esto se consigue con la alternativa condicional.

- *¿Qué debemos hacer para detener la medición de tiempo? ¿Qué programa podemos mirar para ayudarnos? ¿Qué bloques vamos a usar? ¿Dónde los vamos a colocar? ¿Por qué? ¿A qué se parece?*

Para terminar la medición de tiempo, debe tomarse la marca de tiempo final. En el programa del cronómetro (al presionar el botón B), pueden verse los bloques necesarios para esto. Estos bloques deben colocarse dentro de la alternativa condicional del programa que cuenta 10 saltos, pues allí se habían colocado los bloques que queríamos ejecutar cuando se alcanzaban los 10 saltos. Antes solo informábamos que se

había completado la rutina, ahora debemos, además, tomar la marca de tiempo para finalizar la medición.



Variante de complejidad

Visualización Se pueden mostrar los saltos restantes y, al completarse el ejercicio, informar automáticamente el tiempo medido.

3. Cierre (10 minutos)

Este cierre refuerza las reflexiones de la etapa anterior, ahora sobre una nueva experiencia de programación.

Los docentes acompañan a los grupos para que identifiquen **cómo llevaron a cabo la combinación y adaptación de programas previos**, para poner de manifiesto y **encontrar en su propia experiencia** (tanto en esta etapa como en la anterior) aspectos fundamentales de la práctica de reutilización y adaptación.

- *¿A qué programa/s se parece el programa que acaban de construir?
¿Cómo les ayudó a programar haber identificado estos parecidos?
¿Tuvieron que volver a pensar cómo usar las marcas de tiempo? ¿O qué condición poner en la alternativa?*
Programas aparentemente distintos, pueden tener un comportamiento similar. Identificar estas similitudes allana el diseño del programa final y por eso vale la pena analizar los programas desde su comportamiento, para encontrar estos parecidos.
- *¿Un programa siempre tiene una única función? ¿Cuántas cosas hace el programa "Cuánto demorás en saltar 10 veces"? ¿Podemos identificar qué parte del programa hace cada una? ¿Cada una de estas partes, se parece a algún programa que ya hicimos? ¿Por qué?*

Un mismo programa puede combinar comportamientos y se puede identificar qué sector del programa es responsable de cada uno.

- *¿Fue fácil identificar los bloques responsables de cada una de las funciones que queríamos tomar de los programas originales? ¿Por qué?*
Es importante construir programas de manera que la relación entre bloques (código fuente), comportamiento y problema que resuelven sea lo más clara posible. Esto no solo facilita la elaboración de esos programas en particular, sino que también simplifica la creación de programas en el futuro para resolver nuevas necesidades.

El DR explicita que de ahora en adelante se trabajará siempre bajo estas premisas.

ETAPA 5↓ Diseño y construcción

En esta etapa, se define el circuito de prácticas motrices por estaciones y se construyen dispositivos que incluyen a la micro:bit para medir, fiscalizar o asistir el ejercicio de cada estación.

En el aula, se construye el circuito entre todos y cada grupo se hace responsable de una estación, para la que diseña un dispositivo, construye su parte física y describe la función del programa que controlará la micro:bit.

En la VC, a partir de las elaboraciones en el aula, cada grupo diseña e implementa el programa para su dispositivo.

Objetivos

Se espera que los estudiantes sean capaces de:

- Diseñar los programas de cada dispositivo a construir e identificar qué características trabajadas anteriormente necesita cada uno.
- Construir los programas necesarios para los dispositivos que idearon.

Coordinación dupla pedagógica

Información que necesita tener el DR:

- Consulta al DA del avance en su proyecto áulico para contextualizar la VC.
- Breve intercambio a propósito de cómo resultaron las clases en la Etapa 4. Además, se repasan todas las mediciones y sensores (y sus combinaciones) trabajados para contribuir a las ideas de los estudiantes.

Decisiones del DA:

- Organización de la etapa: la actividad sugerida en el aula es el disparador para las actividades de las VC, pero dado que estas llevarán más de una semana, el diseño de los dispositivos físicos y su soporte se puede ir realizando de forma paralela en las sucesivas semanas que lleve la programación.
- Para que durante la VC el DR pueda acompañar a cada grupo en la programación, es importante que en la clase de aula, en conjunto con el DA y el prof. de Ed. Física, se defina el circuito de prácticas motrices, las estaciones que lo conforman y los grupos de trabajo a cargo de cada una.

AULA ↓

Circuito de prácticas motrices

Notas para el DA ↓



Propósitos mínimos

- Fomentar que cada grupo de trabajo asuma la responsabilidad de una estación que forma parte de un circuito de prácticas motrices para el cuidado de la salud.
- Acompañar a los estudiantes en la creación de sus propios diseños que contemplen el ejercicio a realizar y la placa micro:bit.
- Fomentar la creación de un documento por grupo de trabajo para reflejar las decisiones de diseño, ya sea en papel o en formato digital.

Propósitos óptimos

- Planificar el uso a nivel institucional del circuito de prácticas motrices.
- Impulsar el registro del proceso de construcción de los dispositivos con fotografías, videos o pequeños relatos.

Todo el grupo clase acuerda un circuito de prácticas motrices compuesto por diferentes estaciones -a modo de Gincana- y cada grupo de trabajo elige una de ellas para hacerse cargo. La actividad de cada estación se debe poder medir, contar, sensor con la micro:bit, retomando las herramientas aprendidas en las etapas anteriores. Las estaciones pueden identificarse con un nombre y una breve descripción de la función de su dispositivo. Ej: "Abdominador: cuántos abdominales se hacen en 10 segundos". Pulsera o Vincha.

En las instancias de aula que suceden a medida que se avanza con la programación de las placas, se puede trabajar sobre el **diseño** de los dispositivos (su funcionalidad, su aspecto físico y un soporte acorde a su uso), prever materiales necesarios y organizar el proceso de **construcción**.

Se sugiere alentar la **documentación** de todo el proceso de construcción, las decisiones iniciales, las ideas para el diseño con dibujos o esquemas y la planificación de la estrategia de programación.

VC ↓

Diseño y construcción de los programas

🕒 Actividades previstas para requerir más de una VC de 45 min

💡 Desafío

Diseñar e implementar los programas necesarios para los dispositivos ideados para el circuito.

1. Inicio

Diseño y planificación del programa

Cada grupo pone en común el diseño del dispositivo que imaginó y la descripción de su función (**Paso 1: describir el comportamiento del nuevo programa**). Ej: *Queremos contar cuántos abdominales se hacen en 10 segundos.*

El DR escucha y plantea interrogantes o brinda pistas para que entre todos identifiquen qué características de las trabajadas anteriormente requiere cada dispositivo (**Paso 2: identificar comportamientos conocidos**).

Ej. *El dispositivo para contar cuántos abdominales se hacen en 10 segundos requiere: contador, cuenta regresiva, sensor de movimiento.*

También puede producirse alguna referencia más directa para la programación, como la especificación de una estrategia. Ej. *Incrementar el contador de abdominales solo si la cuenta regresiva no llegó a cero.*

Cada grupo completa su documento de diseño con la información obtenida en esta instancia.

2. Desarrollo

Construcción de los programas

Cada grupo avanza en sus programas de forma autónoma conforme el diseño previsto. El DR está atento a las preguntas que pudieran surgir y completa con preguntas de chequeo, mientras que el DA recorre los grupos, detecta

obstáculos y recoge inquietudes

La estrategia sugerida para los estudiantes es la misma a la trabajada en las últimas etapas: a partir de las características de comportamiento identificadas en el documento de diseño, recuperan los programas que las contienen (**Paso 3: Buscar otros programas que contengan estos comportamientos**, Ej.: contador de saltos, cuenta regresiva), se identifican las partes pertinentes y se combinan en el nuevo programa, con las adaptaciones necesarias (**Paso 4: Adaptar al nuevo programa**).

3. Cierre

Prácticas computacionales (5 min al finalizar cada VC)

Al finalizar cada VC será importante dedicar un momento de reflexión para subrayar las prácticas computacionales que se pusieron en juego durante las actividades realizadas. El DR puede elegir alguna práctica que haya sido más recurrente y significativa para destacar.

- *¿Qué herramientas de programación usaron para sus programas? ¿Para resolver qué problema? ¿Se acuerdan qué problema resolvieron con ellas cuando las aprendimos? ¿Se imaginan otros problemas que puedan resolverse con ellas? ¿Alguno de su vida cotidiana?*

Reflexión sobre las **nuevas herramientas de programación aprendidas** (variables, alternativas) y la variedad de usos que se le pueden dar. Se pueden retomar algunos de los ejemplos de situaciones cotidianas en las que una computadora almacena un dato y lo utiliza más adelante en la ejecución de un programa, o cómo un mismo programa se comporta de diferentes maneras en base a si se cumplen o no ciertas condiciones.

- *¿Qué ventajas les parece que tuvo empezar pensando en el diseño del programa y cómo programarlo, antes de pasar a la programación en sí?*

¿Qué cosas descubrieron cuando describieron sus programas?
 ¿Encontraron cosas conocidas o repetidas?

Reflexión sobre la **importancia del diseño y la planificación en la programación**. Explicitación de cómo, a partir de algunos pocos comportamientos básicos, se pueden construir numerosos dispositivos distintos. Recuperación de la experiencia de diseño de la última etapa y su relación con la implementación de los dispositivos. Se puede presentar como una manera de resolver un nuevo problema en términos de otros que ya se saben resolver. Luego, para construir la solución no hace falta empezar de cero, solo concentrarse en cómo aprovechar las soluciones existentes.

- ¿Se acuerdan de los dispositivos que hicimos en las primeras VCs? ¿Por qué les parece que les propusimos empezar por esos? ¿Les sirvieron después para programar sus dispositivos? ¿Cómo los aprovecharon?
 Metacognición sobre **reutilización y reinención**: en las primeras VCs se trabajó con pequeños programas con las características deseadas de los dispositivos finales, como una manera controlada de programar un aspecto clave del comportamiento. Recién después, se recuperaron esos programas para adaptarlos, combinarlos y usarlos para resolver desafíos más complejos.
- ¿Por qué les parece que es importante poder entender fácilmente qué hace un programa? ¿Alguna vez tuvieron que ir a buscar cuáles eran los bloques responsables de algo en particular (como incrementar una cuenta, mostrar un mensaje, desactivar un botón, o incluso algún error)? ¿Les fue fácil? ¿Podría haber sido más fácil?
 Metacognición sobre **lectura, análisis y corrección de código**: cuando

fue necesario identificar distintas partes de un programa para reproducir alguna característica particular en otro, modificarlo para que hiciera algo sutilmente distinto de lo que hacía o, incluso, corregir un error, fue muy importante poder leer e interpretar el código fácilmente.

Bitácora y Foro

Se completa entre todos el mapa conceptual con los programas, conceptos y prácticas puestos en juego. Se pueden enriquecer los pasos para la reutilización de programas con las nuevas experiencias surgidas en esta etapa.

Los estudiantes pueden publicar sus reflexiones en el foro ¿Qué aprendimos? y adjuntar sus programas para que el DR pueda realizar un seguimiento.

Se invita a realizar la actividad interactiva de esta etapa.



ETAPA 6↓

Cierre: Ensayo y reflexión.

En el aula se trabaja sobre la documentación que los grupos fueron generando a lo largo de su trabajo para planificar su presentación y generar los discursos y materiales necesarios. También, el DA puede aprovechar esta instancia para dar cierre a su proyecto de aula.

En la VC se simula un ensayo general de la presentación del circuito que genera un intercambio entre estudiantes y docentes para plantear inquietudes, evaluar la apropiación de los conceptos y mejorar las explicaciones de los grupos. También puede incorporarse una actividad de coevaluación.

Para terminar, se realizan reflexiones generales sobre la propuesta de trabajo y la metacognición sobre los conceptos involucrados y las prácticas ejercidas. Se realiza en Crea, la actividad de evaluación final del proyecto.

Objetivos

Se espera que los estudiantes sean capaces de:

- Exponer el funcionamiento de sus dispositivos dentro del circuito de prácticas motrices y describir el proceso de elaboración, mediante una comunicación clara y completa de los procesos realizados.
- Explicitar y tomar conciencia de su propio proceso de aprendizaje durante todo el proyecto.

Coordinación dupla pedagógica**Información que necesita tener el DR:**

- Modo en que se prevé la instancia de socialización, preparativos necesarios, tiempos y contexto para que el DR pueda acompañar y si es posible participar en forma virtual.

Trabajo conjunto entre DA y DR:

- El ensayo general propuesto para la VC es una oportunidad para realizar ajustes y mejoras a la programación, pero también a la exposición planificada en el aula, tanto para expresión oral como para las precisiones sobre el proceso transitado. Se puede prever una dinámica para que contemple las inquietudes de ambos docentes.
- El DA fue quien presenció todo el trabajo de los estudiantes, acompañó y recibió sus inquietudes más urgentes y pudo observar directamente el desarrollo de cada grupo. Por lo tanto, cuenta con el insumo fundamental para el ejercicio de metacognición propuesto. Su rol en la actividad de metacognición es evocar los aspectos de la experiencia que se vuelven relevantes con las sucesivas preguntas del DR y que tal vez los niños no identifican o valoran a priori.

AULA ↓**Preparativos y cierre del proyecto de aula****Notas para el DA ↓****Propósitos mínimos**

- Acompañar a los estudiantes en la preparación de los materiales para la socialización: organización de sus bitácoras, documento de diseño y exposición oral de la estación a su cargo, elaboración de afiches o gráficos explicativos.

Esta instancia puede convertirse en una oportunidad para abordar contenidos vinculados a la comunicación escrita y oral, para afinar algunos conceptos clave, evaluar la comprensión y propiciar la metacognición sobre los aprendizajes curriculares que dejó el proyecto.

Propósitos óptimos

- Preparar el cierre del proyecto de aula.
- Acompañar a los estudiantes en la organización de su exposición oral para exponer los desafíos que enfrentaron, cómo los resolvieron y qué aprendieron en ese recorrido vinculando el deporte, la salud y la tecnología.

**Bitácora**

En lo posible subir al foro en Crea fotografías o enlaces al documento de diseño para conservar un registro digital del proyecto terminado.

VC 6↓

Ensayo general y reflexión final

1. Desarrollo**Ensayo general (20 min)**

Se realiza la disposición de las distintas estaciones con todos los elementos necesarios y las placas programadas. El DR puede oficiar de visitante virtual de la exposición y hacer preguntas a los grupos para que amplíen sus explicaciones.

¿Qué nombre tiene la estación y el dispositivo? ¿El soporte pensando es adecuado? ¿La explicación permite entender qué se mide o contabiliza con la placa y cómo funciona? ¿Qué función cumple dentro del circuito global?

La dinámica de esta actividad está planteada como un ensayo general de la socialización del circuito y los dispositivos. Es una oportunidad para realizar ajustes en la exposición oral o la documentación creada, evaluar la apropiación de los nuevos conceptos de pensamiento computacional abordados, intervenir en caso de ser necesario, socializar con sus pares para confrontar opiniones. Si ambos docentes lo consideran oportuno, se sugiere realizar una **evaluación entre pares** mientras cada grupo expone las características de su estación. Esto permitiría además favorecer la escucha atenta y la colaboración.

2. Cierre**Metacognición (20 min)**

A partir de la dinámica de cierre que los docentes acuerden, el propósito de la metacognición es que los estudiantes reflexionen sobre:

- **Los ejes planteados para los cierres de las VCs de la etapa anterior que no hayan podido ser abordados.**

- **La conceptualización de programa como responsable del comportamiento de una computadora**

¿Cuántas cosas se pueden hacer con una micro:bit? ¿Qué hay que cambiar para que la micro:bit haga cosas diferentes? ¿Esto es parecido a lo que sucede con las computadoras que usamos todos los días? ¿En qué se parecen y en qué se diferencian una computadora-micro:bit y una computadora-teléfono? ¿Cómo se comporta el teléfono si apretamos el botón "cámara" del celular? ¿y si apretamos el botón del whatsapp?

Se vio que exactamente las mismas placas hacían cosas muy diferentes en las distintas etapas cuando se le cargaba un programa u otro. Para generalizar esta noción, se puede ampliar a muchas de las computadoras que usamos en nuestra vida cotidiana (tomaremos el ejemplo de un celular): dependiendo del programa que estamos usando cambia su comportamiento (nos permite sacar fotos o enviar mensajes, por ejemplo). Así se ve cómo el comportamiento de nuestras computadoras también está determinado por un programa. A diferencia de la micro:bit que sólo puede almacenar un programa, nuestras computadoras, en general, pueden tener varios almacenados y nos permiten elegir cuál queremos usar sin necesidad de pasar por un proceso de instalación. De todas maneras, también podemos instalar programas nuevos para hacer cosas nuevas.

- **El proceso de aprendizaje**

¿Qué cosas de las que hicimos no conocían? (qué aprendimos) ¿Cómo relatarían a otra persona el camino recorrido durante este proyecto? (cómo lo aprendimos), ¿qué lograron construir con sus nuevos conocimientos? (para qué les sirvió), ¿en qué otras situaciones creen que pueden utilizar lo aprendido? (generalización).

- **Aspectos emocionales**

*¿Recuerdan qué fue lo que los ayudó a destrabar un problema o encontrar una solución? ¿Distinguen alguna actitud propia o de los compañeros que los ayudó a avanzar con el proyecto? ¿Se escucharon todas las voces y se buscó que se reflejen en el producto final? ¿Se tuvieron en cuenta las ideas y fortalezas de cada miembro del equipo? ¿Cómo enfrentaron el error? ¿Lo tomaron como una oportunidad de aprendizaje o los frustró? ¿Por qué?
¿Recuerdan cómo se sintieron frente al desafío al iniciar este proyecto? ¿Y cómo se sienten ahora con lo logrado? ¿Cómo se sintieron a lo largo de las distintas etapas?*

Para cerrar y valorar los esfuerzos que requirieron estos aprendizajes, identificamos entre todos que lo que los estudiantes hicieron fue crear distintos programas de computadora para resolver una variedad de

problemas. De esta manera, podemos entender un programa como solución a un problema y podemos reconocer prácticas valiosas en su elaboración (descomposición, reutilización, reinención, etc.).



Evaluación final del proyecto




Los estudiantes completan la actividad de evaluación del proyecto.

ANEXO 1

¿Es una computadora?

Ejemplo de registro del análisis conjunto de cada artefacto.

Artefacto	¿Toma datos, acepta entradas, recibe estímulos?		A partir de los datos de entrada... ¿Toma decisiones o calcula algo?		¿Genera una salida o mueve alguna cosa?		Conclusión: ¿Es probable que tenga una computadora en su interior?
<p><u>Nintendo Switch</u></p> <ul style="list-style-type: none"> - Pantalla táctil. - Procesador y memoria. - Botón POWER / Botón de volumen. - Acelerómetro, giroscopio y sensor de brillo. - Batería 	Sí	Recibe los estímulos de los botones. Recibe información de brillo.	Sí	Al presionar los botones, mueve los personajes del juego.	Sí	Muestra el juego en la pantalla, emite sonidos.	Sí.
<p><u>Inodoro inteligente</u></p> <ul style="list-style-type: none"> - Botonera para elegir función. - Calefacción y secado - Control remoto. - Limpieza automática. - Presión y temperatura de chorro de agua ajustables. 	Sí	Recibe la información de la botonera o del control remoto.	Sí	Enciende o apaga la calefacción, modifica la presión o temperatura del agua.	Sí	Enciende o apaga el chorro y el calentador de agua.	Sí

<p>Tabla de cortar inteligente</p> <ul style="list-style-type: none"> -Permite cortar ingredientes y agregarlos luego a una olla o sartén sin el riesgo de que se caigan en el camino. -Diseño simple e innovador está disponible en una amplia gama de colores y tamaños. 		<p>No</p>		<p>No</p>		<p>No</p>		<p>No</p>
<p>Almohada inteligente</p> <ul style="list-style-type: none"> - Autoamoldable: se adapta al contorno del cuerpo. - Retorno lento que sirve como amortiguador. - Recupera su forma original. - Protección antiácaros, hongos y bacterias. 		<p>Sí</p>	<p>Recibe el estímulo de la cabeza apoyada.</p>	<p>No</p>	<p>(Solo conserva la forma de la cabeza, pero no realiza ningún cálculo para eso)</p>	<p>No</p>	<p>(Solo se queda quieta en la forma que adquirió)</p>	<p>No</p>
<p>Micro:bit</p> 	<p>Sí</p>	<p>Recibe información de los botones.</p>	<p>Sí</p>	<p>El texto que se muestra depende del botón presionado.</p>	<p>Sí</p>	<p>Enciende luces en la pantalla.</p>	<p>Sí</p>	

[Volver a la etapa 1](#)