

Cartas:bit

Programamos para jugar

En esta propuesta, los estudiantes construyen un juego automático con las placas micro:bit en el que deberán voltear, sacudir o dejar boca arriba solo las que cumplan una determinada condición permitiendo hacer prácticas de cálculo numérico.

El juego está relacionado con el área del conocimiento matemático, por lo cual es necesario investigar o revisar el contenido elegido con el docente de aula. Si bien, en este proyecto se construye y programa un juego automático que permite trabajar de forma divertida y diferente con fracciones, la propuesta se puede adaptar a otros contenidos.

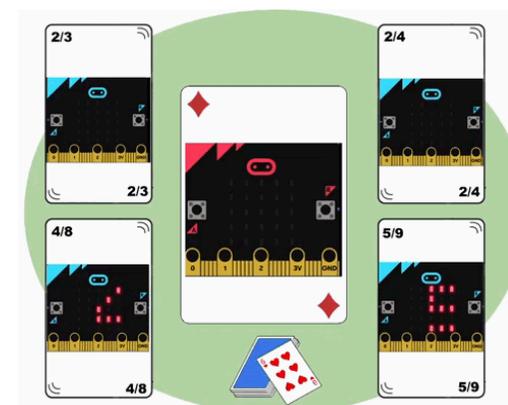
En relación a los conceptos computacionales, se trabaja con procedimientos para expresar estrategias, se utilizan listas y se organiza un desarrollo incremental. La experiencia con las placas micro:bit se centra en el uso de la radio para la comunicación entre placas y el uso de sensores.

Duración: 8 a 10 semanas.

Materiales: Placa micro:bit.

¿Cómo es el juego?

La placa organizadora (**Super:bit**) envía una señal de Inicio de partida, las placas jugadoras (**Cartas:bit**) muestran una fracción aleatoria. Los jugadores deben identificar, si la fracción que muestra su Carta: bit es equivalente a, por ejemplo, $1/2$ y dejarla boca arriba o boca abajo según corresponda. La misma dinámica puede adaptarse para trabajar con otras consignas como identificar las que son mayores o menores a la fracción mostrada, mayores o menores que 1, etc.



Índice

Ficha Curricular ↓	2
Objetivos de aprendizaje de 3° año de Pensamiento Computacional	2
Posibles vinculaciones a otros espacios y unidades curriculares.	3
Síntesis del recorrido en Etapas	4
Acuerdos iniciales de coordinación	5
Micro:bit	6
ETAPA 1 ↓	7
ETAPA 2 ↓	11
ETAPA 3 ↓	15
ETAPA 4 ↓	20
ETAPA 5 ↓	25
ETAPA 6 ↓	30
ETAPA 7 ↓	34
ETAPA 8 ↓ (opcional)	39
ETAPA 9 ↓	45
 ANEXO 1	49
 ANEXO 2	50
 ANEXO 3	51
 ANEXO 4	53

Ficha Curricular ↓

Objetivos de aprendizaje de 3° año de Pensamiento Computacional

Comunicación y Colaboración

- Colaborar y construir conjuntamente un proyecto grupal.

Resolución de problemas computacionales

-Definir diferentes soluciones computacionales que pueden resolver un problema dado.

-Incorporar la estrategia de división de un problema en subproblemas.

-Seleccionar las herramientas de programación adecuadas para el subproblema a resolver.

-Recuperar estrategias de solución y herramientas de programación construidas en experiencias anteriores para adaptarlas a nuevos problemas.

Análisis de datos, desarrollo y uso de Abstracciones

-Analizar los datos importantes y la información relevante de un problema dado.

-Reconocer las generalidades de los comportamientos de sus programas, en términos abstractos, para que puedan ser adaptados, combinados y/o reutilizados.

-Explorar distintas formas de representación y sistematización de los datos para modelar un problema.

Creación de algoritmos, programas y dispositivos

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

-Utilizar las herramientas de programación adecuadas para implementar sus programas.

-Utilizar el método de desarrollo incremental.

-Comprender la posibilidad de reutilizar partes de programas ya creados en la construcción de programas nuevos.

Evaluación

-Identificar y corregir, con ayuda del docente, errores mediante un proceso sistemático.

-Comparar las diferencias entre versiones y resultados de dispositivos y/o programas.

Contenidos PC:

•Procedimientos • Listas • Plan de desarrollo • Método de desarrollo incremental

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

Posibles vinculaciones a otros espacios y unidades curriculares.

A definir por maestro/a de aula

Competencias generales

Comunicación, Pensamiento Computacional, Pensamiento Científico, Metacognitiva, Relación con los otros, Iniciativa y orientación a la acción.

Es importante que el contenido puesto en juego durante el proyecto pueda adaptarse a los objetivos de aprendizaje previstos por el DA. Se identifican algunos contenidos del 2do ciclo que podrían trabajarse:

Área de Conocimiento Matemático:

NUMERACIÓN NATURAL. Otras regularidades: divisibilidad. Números divisibles entre 3, 6 y 1000. Números primos y compuestos. Múltiplos.

OPERACIONES. Adición y sustracción, multiplicación y división: operaciones combinadas, propiedades, cálculo pensado.

Competencias específicas

CE.5 Explora, de forma colaborativa, problemas computacionales simples, siguiendo secuencias, en situaciones lúdicas y cotidianas, para dar respuestas a interrogantes planteadas.

CE.6 Explora y utiliza lenguajes de programación con algunos símbolos para comprender y crear algoritmos sencillos como una configuración de instrucciones que producen acciones.

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.

Materiales complementarios sugeridos

- Video ¿Cómo programar mi micro:bit? <https://youtu.be/pKt5k1wXSg> y otros videos en el [Canal micro:bit](#) o [Mis primeros pasos](#) o en pdf la [Guía básica en Youtube](#)
- Video [Sitio micro:bit del Plan Ceibal](#) <https://micro.bit.ceibal.edu.uy/> En la sección [Recursos](#) se encuentran Fichas para docentes con más ideas para articular el uso de las placas con contenidos programáticos.

Criterios de logro

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

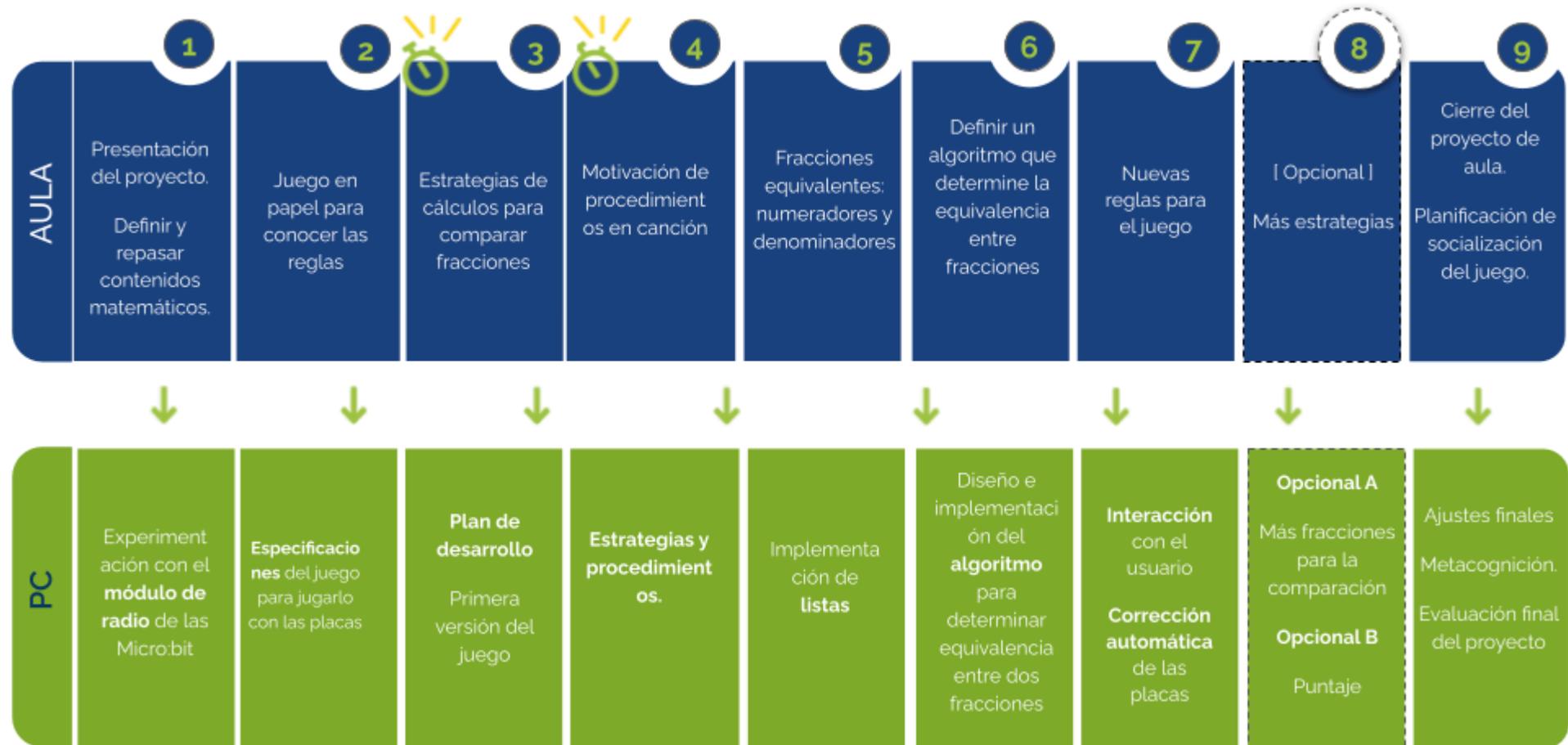
Divide un problema en subproblemas, analizando datos importantes e información relevante.

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.

Síntesis del recorrido en Etapas



🕒 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 de **docentes remotos (DR)** y **docentes de aula (DA)** es fundamental para llevar adelante esta propuesta.

Decisiones del DA → comunicar a DR :

- Definir el modo en que se trabajará con las fracciones equivalentes durante el proyecto.
- **Coordinar en qué momento los estudiantes probarán los avances del juego, utilizando las Microbit.**

Decisiones DR → comunicar a DA:

- Explicitar al DA semanalmente los objetivos de cada VC y establecer acuerdos en torno a la dinámica de las clases remotas, la organización espacial necesaria y la participación del DA.

Información que necesita tener el DR:

- Los conocimientos del grupo en relación a las fracciones y su modo de abordaje.
- Experiencias previas en el trabajo con placas micro:bit. tanto de los estudiantes como del DA.

Juego de ejemplo:

Se dispone de un juego completo realizado exclusivamente para consulta de los docentes: [Proyecto Cartas:bit](#). En este ejemplo, se decidió trabajar con equivalencias de fracciones.

Rol del DA durante las VC

- En las actividades de **inicio** organiza el intercambio para que los estudiantes relaten al DR lo realizado en el aula.
- En 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.
- En 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.
- Durante todo el proyecto serán valiosas las acciones del DA que favorezcan el **vínculo** de los estudiantes con el proyecto y el DR.
- Durante los **intercambios**, facilitar la circulación de la palabra, permitirá que todos los estudiantes tengan oportunidad para expresarse.

Rol del DR durante el proyecto

- Anticipar al DA el modo y el contenido planificado para cada VC.
- Indagar los contenidos programáticos que el DA elige para acompañar la propuesta pedagógica y resignificarlos durante la 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 que correspondan.

Micro:bit

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

- 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.

Disponibilidad de placas micro:bit entre los estudiantes

Se recomienda que cada estudiante y DA dispongan de 1 (una) placa o, de no ser posible, la cantidad suficiente de placas para que la dinámica del juego pueda implementarse .

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, que contiene una estructura similar a la de esta guía. Este espacio virtual ofrece herramientas de trabajo que servirán al DR a llevar adelante distintos momentos en la VC.



Actividades Interactivas

Las actividades interactivas están pensadas para ser realizadas en distintos momentos en cada etapa. En alguna oportunidad pueden ser una instancia de aprendizaje de los contenidos, en otras pueden formar parte del cierre del desarrollo de la clase. Lo importante es recuperar la resolución de las mismas para realizar una puesta en común.

Foro de evidencias

Los avances del proyecto se comparten en los foros de evidencias. Se puede elegir uno o varios aportes, así como proyectos de programación para analizarlos entre todos durante la VC. Lo importante es socializar la programación y enriquecer el intercambio de ideas.

Reflexión y registro de cierre

A lo largo de toda esta propuesta se propone plasmar los intercambios del cierre en **un registro común** para toda la clase que se va enriqueciendo en cada etapa. Cada pareja de docentes considerará la herramienta más adecuada que permita compartir un enlace con los estudiantes en la plataforma. Puede utilizarse un documento compartido para tomar el registro, una página creada en Crea o incluso mapas conceptuales realizados a partir de los intercambios grupales. Las dinámicas para la escritura en este archivo podrán ir variando entre una etapa y otra. Algunas veces se puede recurrir a la **escritura por parte de los docentes**, otras veces se puede **recopilar respuestas de un foro**, compilar imágenes de **capturas de pantalla** o solicitar **escrituras parciales** a subgrupos.

ETAPA 1 ↓**Las Micro:bit se comunican**

En esta etapa se presenta el proyecto, se elige el contenido matemático a utilizar en el juego y se retoma la práctica con la placa Micro:bit y el entorno de programación.

En el aula, una vez presentado el proyecto general, se aborda el contenido matemático que van a utilizar para la realización del juego.

En la VC, se experimenta con el módulo de radio de la Micro:bit para enviar información de una placa a otra.

Objetivos

Se espera que los estudiantes sean capaces de:

- Explorar y reconocer los bloques que permiten la comunicación entre placas.
- Experimentar el envío y recepción de distintos tipos de datos entre dispositivos y las limitaciones que se presentan.

Coordinación dupla pedagógica**Decisiones conjuntas entre DA y DR:**

- La forma que adoptará el registro común para todo el proyecto y la herramienta a utilizar.
- La participación del DA para gestionar las primeras experiencias de los estudiantes con el nuevo entorno será muy valiosa (ingresar al entorno, subir un archivo, ejecutar en el emulador). Con la guía del DR, el DA puede acompañar a los estudiantes que necesitan más ayuda.

Decisiones del DA

- El recorrido didáctico que se llevará a cabo para alcanzar los propósitos de la instancia de aula.
- El nivel de profundidad o alcance en el abordaje del contenido matemático elegido.

Información que necesita tener el DR:

- La cantidad de placas disponibles entre los estudiantes.
- Cómo se organizará el grupo para llevar a cabo el proyecto.
- El modo en que el DA propone abordar el contenido matemático.

[Avance Etapa 1](#)

AULA ↓**Introducción al contenido matemático****Notas para el DA ↓****Propósitos mínimos**

- Acompañar a los estudiantes en la elección del contenido matemático a utilizar en el desarrollo del juego que van a programar.

Propósitos óptimos

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

Presentación del desafío global de PC:

Programar un juego automático con las placas Micro:bit para hacer prácticas de cálculo numérico.

Contenido matemático:

Este espacio es propicio para debatir con los estudiantes acerca de qué contenido matemático pueden seleccionar para utilizar en el juego que programen. Se sugiere elegir el contenido matemático que el DA considere necesario repasar y practicar.

Por ejemplo:

- Números racionales
 - Fracciones equivalentes, menores y mayores que la unidad
 - Fracciones de distinto denominador e igual numerador, comparación y ordenación

VC ↓ Radio

💡 Desafío

Programar una Micro:bit para que al presionar un botón, envíe un mensaje a otras Micro:bit y muestren un ícono.

1. Inicio (10 min)

¿Qué características tienen en común los dispositivos que muestran las imágenes? ¿Cómo es posible que puedan establecer una comunicación o que realicen un intercambio de información entre ellos?



Esta introducción busca que los estudiantes identifiquen, como característica común de estos ejemplos, la comunicación que se establece entre los dispositivos y el intercambio de información que se produce. Uno de sus componentes reacciona en respuesta a la acción realizada por el otro componente. Por ejemplo: se presiona una tecla del control remoto, y la televisión cambia de canal. Las placas Micro:bit también pueden establecer un intercambio de información entre ellas a través de señales de radio.

2. Desarrollo (20 min)

Parte 1: Enviar un número o cadena de una placa a otra

Al explorar la categoría **Radio** ¿qué bloques pueden elegir para enviar un número o cadena de una Micro:bit a otras? ¿Qué bloques posibilitan que otras placas reciban el número o cadena enviada? ¿Cómo pueden mostrar el dato recibido?

El DR presenta la *primera parte del desafío* y habilita el tiempo necesario para que los estudiantes exploren MakeCode. En esta actividad se trabaja sobre una consigna sencilla que permite reconocer cada sector de MakeCode y retomar el uso de distintos bloques de *Entrada* (al presionar el botón, si

agitado, etc), así como los *Básicos* (mostrar número, mostrar cadena). Los estudiantes prueban los bloques de la categoría *Radio* que permiten establecer una comunicación e intercambio de información entre las placas.

Los estudiantes programan para completar la Parte 1 y realizan las pruebas necesarias en el simulador. El DR debe intervenir lo menos posible, para permitir el aprendizaje por medio del descubrimiento. De ser necesario, puede guiar en el proceso de descarga del programa a las Micro:bit.

👉 Sugerencia

Tener en cuenta que en el entorno MakeCode, al usar los bloques de radio, el simulador muestra la placa que envía la señal y la placa que la recibe. Si el programa se descarga íntegramente a dos placas que cumplirán roles distintos, pueden surgir errores operativos y ser confundidos con errores de programación.

Una alternativa, para solucionar estos inconvenientes, es usar la opción <https://makecode.com/multi#> de Makecode. La misma divide la pantalla y permite realizar dos programas en simultáneo, favoreciendo el desarrollo de la programación y prueba de los programas para la placa emisora y la receptora por separado. Luego, descargar cada uno de los programas con un nombre distinto según la placa que corresponda.

Puesta en común:

¿Cómo resolvieron el problema? ¿Qué bloque usaron para enviar un número y cuál para enviar una cadena?

Una vez resuelta la primera parte se realiza una puesta en común breve, que permita un intercambio de experiencias sobre el modo en que cada grupo resolvió la tarea.

Parte 2: Completar el desafío de la etapa

Ya consiguieron que las placas se comuniquen y se produzca un intercambio de información. Ahora deben programarlas para que la placa receptora muestre

un ícono en el momento de recibir un mensaje.

Se habilita tiempo para que los estudiantes resuelvan el desafío de manera autónoma recuperando la experiencia anterior.

Sugerencia

En función del tiempo disponible y la cantidad de placas, la actividad prevista en la VC puede realizarse íntegramente en el simulador de MakeCode, aunque se recomienda el uso de las Micro:bit para fortalecer la práctica de la descarga de los programas y la verificación del funcionamiento de los mismos.

A la hora de ejecutarlo en las micro:bit, será importante que cada grupo se asigne un **número de grupo de radio diferente** para no interferir con las pruebas de los demás.

Variante de complejidad

Programar las placas para que al presionar el Botón A en la placa emisora, aparezca en las otras una Cara feliz. Y al presionar el Botón B, aparezca una Cara triste.

Esta variante puede resolverse combinando las estrategias con mensajes de tipo numéricos (para un botón) y de tipo cadena (para el otro botón) o utilizando estructuras condicionales.

3. Cierre (10 min)

¿Por qué una placa no puede enviar un ícono de manera directa a otra placa?

¿Cómo resolvieron el problema? ¿Todos lo solucionaron de la misma forma?

¿Qué diferencias y similitudes encuentran? ¿Qué otra reacción podría tener la placa receptora en vez de mostrar un ícono?

Se espera que los estudiantes reflexionen sobre la posibilidad que tienen las placas de enviar y recibir datos de tipo numérico y de tipo cadena, y reconocer la limitación al probar enviar un ícono. Por tal motivo, cuando la placa receptora recibe la señal (sin importar el tipo de mensaje -numérico o cadena-), ejecuta la instrucción que se haya programado, en este caso muestra un ícono. La placa receptora puede realizar cualquier acción que esté programada dentro del bloque *al recibir radio*, por ejemplo: mostrar la temperatura, reproducir un sonido, indicar dónde está el Norte, entre otros.

Registro en Crea

El DR publica el registro con las notas y reflexiones de los intercambios en el **Registro Común**. Es importante conservar los archivos .hex para continuar trabajando sobre ellos en las próximas VC. A modo de resguardo se pueden publicar como Actualización en el curso o en el foro de la etapa. Pueden incorporarse capturas de pantalla de las actividades de los estudiantes.



Invitar a los estudiantes a resolver la [actividad interactiva](#) de la etapa en las que se repasen el uso de los bloques de radio.

La Yapa: Propuestas para seguir en casa

¿Puedes programar un intercambio de mensajes con otro compañero usando dos placas? ¿Cómo lo harías?
¿Qué preguntas y respuestas pueden enviarse?



<p>ETAPA 2 ↓ El juego</p>	<p>Coordinación dupla pedagógica</p>
<p>En esta etapa, los estudiantes juegan en formato papel y realizan la especificación del juego automático.</p> <p>En el aula, se familiarizan con las reglas del juego que van a crear a partir de jugarlo en formato “papel”.</p> <p>Durante la VC, describen el comportamiento que deben tener las Micro:bits para crear la versión automática.</p> <p>Objetivos</p> <hr/> <p>Se espera que los estudiantes sean capaces de:</p> <ul style="list-style-type: none"> • Comprender la dinámica del juego, similar al que van a programar con las Micro:bit. • Reconocer y registrar el comportamiento de las placas teniendo en cuenta la interacción con el usuario y la circulación de información. • Recuperar la noción de sensores como información de entrada a la Micro:bit. 	<p>Decisiones conjuntas entre DA y DR:</p> <ul style="list-style-type: none"> • En caso de trabajar en grupo, definir cantidad de integrantes considerando la disponibilidad de placas e intereses particulares que manifieste el DA. • Cómo incentivar la participación de los estudiantes en los foros. • Acordar instancias y formas de moderación de las puestas en común y los diálogos. <p>Decisiones del DA</p> <ul style="list-style-type: none"> • El modo de llevar adelante el juego en papel y el contenido matemático implicado. <p>Información que necesita tener el DR:</p> <ul style="list-style-type: none"> • Cómo se desarrolló el juego en la instancia de aula. • El contenido matemático con el que se trabajará durante todo el proyecto y los conocimientos previos de los estudiantes.

AULA ↓ Juego en papel

Notas para el DA ↓



Propósitos mínimos

- Brindar un espacio para reproducir un juego en formato papel, que permita a los estudiantes apropiarse de la dinámica del juego que van a programar en el proyecto.
- Facilitar la recuperación de los saberes matemáticos necesarios para jugarlo.

Propósitos óptimos

- Promover la creación de otras dinámicas de juego similares a la presentada en el proyecto que permitan fortalecer los contenidos programáticos.

Juego en formato papel

Brindar un espacio lúdico en el aula ofrece la oportunidad para que los estudiantes se familiaricen con la dinámica del juego e identifiquen qué saberes matemáticos necesitan para jugarlo.

Si bien, durante el desarrollo de la guía, el contenido elegido (a modo de ejemplo) es **fracciones equivalentes**, el DA puede adecuar el juego al contenido programático que prefiera.

Dinámica del juego

El juego comienza cuando el DA (o un estudiante por grupo que ella designe) reparte una fracción a cada estudiante o grupo y luego, muestra otra fracción a todos. Los estudiantes determinan si la fracción que les tocó es equivalente a la que se muestra para todos. Termina la partida cuando dejen la fracción en la posición acordada según la regla del juego. Por ejemplo, dejándola boca arriba, sólo si es equivalente a la que mostró el DA.

Se pueden jugar tantas partidas como el DA considere necesarias, incluso agregar un tiempo de resolución y puntaje por cada pregunta correcta.

En el [Anexo 1](#) se presenta una dinámica posible con el contenido elegido de ejemplo (**fracciones equivalentes**) utilizando un mazo de cartas con fracciones que pueden imprimir para jugar.

VC ↓ Juego con placas

Desafío

Describir cómo se tienen que comportar las placas para jugar el juego.

1. Inicio (5 min)

¿En qué consiste el juego de cartas que jugaron en el aula? ¿Cómo se organizaron para jugarlo? ¿Cuál es el objetivo del juego? ¿De qué modo participan los jugadores? ¿Qué sucede cuando un jugador gana una partida? ¿Cuándo termina el juego? ¿Qué determina quién gana y quién pierde?

El DR ayuda a los estudiantes a recuperar las características del juego realizado en el aula. Es importante que puedan identificar qué necesitan para jugarlo, cómo se juega y cuál su objetivo.

2. Desarrollo: especificación (20 min)

Luego de esta puesta en común el DR orienta a los estudiantes **a adaptar** el juego realizado para que pueda ser jugado con placas Micro:bit.

¿Cuándo inicia el juego? ¿Qué debe hacer la placa principal? ¿Cómo se entera la placa de cada jugador que el juego empezó? ¿Qué deben hacer estas placas? ¿Cómo interactúan los jugadores con sus placas? ¿Cuándo se gana la partida? ¿Cómo se acumulan los puntos?

Se va elaborando entre todos la especificación del juego, una descripción resumida y precisa del comportamiento de las placas. Los estudiantes deben identificar **los dos tipos de placas, momento clave de la operatoria** y la **circulación de información**. Para esto, el DR puede pedirle a los estudiantes que describan cómo se imaginan que se usarían las placas y cómo deberían responder éstas a las distintas acciones de los jugadores.

Se propicia un intercambio de ideas entre los estudiantes para que puedan identificar que el juego a construir está compuesto por varias placas en lugar de cartas, una placa toma el rol de organizadora del juego (**Super:bit**) y las otras placas el rol de jugadoras (**Carta:bit**).

La función de la Super:bit es comenzar el juego enviando un mensaje de inicio a las Carta:bit, mostrar una fracción y terminar la partida.

La función de las Carta:bit es que cada carta jugadora muestre una fracción aleatoria. El jugador decide si es equivalente a la fracción de la Super:bit. Si lo es, una opción es dejar la placa boca arriba y cada Carta:bit puede sumar un

punto si la respuesta fue correcta. En este caso, gana el estudiante jugador o equipo cuya Carta:bit muestre el mayor puntaje.

En base al conocimiento que tengan los estudiantes sobre sensores e información de entrada, pueden decidir de qué manera el jugador indicará si la fracción es equivalente o no. Puede ser orientando la placa (boca arriba / boca abajo, logotipo arriba / logotipo abajo, inclinada hacia la derecha / hacia la izquierda). También pueden aprovechar otros sensores, por ejemplo para responder exponiendo la placa a la luz o a la oscuridad. Para que los estudiantes reconozcan estas posibilidades y decidan cómo hacerlo el DR puede invitarlos a explorar la categoría "Entrada" del entorno MakeCode.

El DR y los estudiantes elaboran un esquema a modo de resumen en alguna pizarra colaborativa. Este esquema se conserva en un documento para ser consultado durante el proyecto.



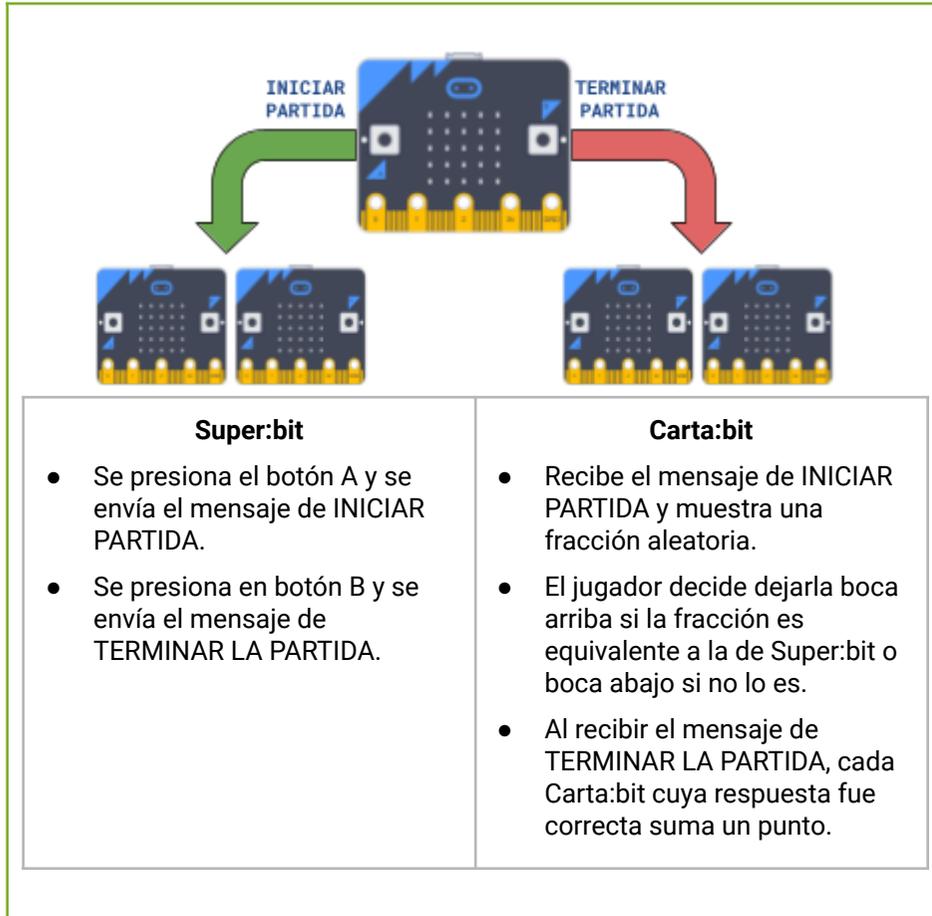
Sugerencia

Como en el desarrollo de la propuesta se van a realizar diversas anotaciones, que progresivamente se irán completando etapa por etapa, sugerimos que las mismas se realicen en un solo documento. [Anotaciones de Cartas](#), es una opción para lograrlo.



Atención

A partir del momento en que los estudiantes identifican la necesidad de contar con placas que cumplan dos roles distintos: una principal que organiza el flujo de información y otra con la que juega cada estudiante, se las llama: **Super:bit** y **Carta:bit**, respectivamente .



Un ejemplo de especificación

★ Importante
 Antes del cierre considerar las recomendaciones respecto a la importancia de realizar las **Actividades interactivas**.

3. Cierre (10 min)

La especificación y las reglas de los juegos

¿Qué necesitan saber para jugar un juego de cartas? ¿En qué se parecen y en qué se diferencian las reglas de los juegos y las especificaciones que escribimos? ¿Para qué sirve tener la especificación antes de empezar a programar?

En base a las respuestas de los estudiantes, el DR señala, por un lado que tanto las reglas como las especificaciones determinan qué sucede en un juego o programa, al decir cómo se deben comportar sus integrantes (qué debe hacer cada jugador al levantar una carta o qué debe hacer la Micro:bit al presionar un botón). Por otro lado, detallar que la especificación es una manera de tener en claro qué hace falta programar y orientarnos en ese proceso.

Registro en Crea

El DR publica el registro con las notas y reflexiones de los intercambios en el **Registro Común**. Se publica el registro de los intercambios y la **especificación** a la que arribaron.

Pueden incorporarse capturas de pantalla de las actividades de los estudiantes.

Invitar a los estudiantes a resolver la [actividad interactiva](#) de la etapa en las que se repasen especificación de un juego e información de entrada en las placas micro:bit.



La Yapa: Propuestas para seguir en casa

¿De qué otras formas puede la placa avisar que tu respuesta fue correcta?



ETAPA 3 ↓**Comenzando a programar el juego automático**

En esta etapa se recupera la estrategia de cálculo utilizada en el juego en papel y se comienza a escribir el plan de desarrollo del proyecto.

En el aula, se retoma la experiencia del juego para analizar y registrar las operaciones matemáticas realizadas al jugarlo.

En la VC, se comienza a escribir el plan de desarrollo que guiará todo el proyecto y se programa la primera etapa funcional del juego.

Objetivos

Se espera que los estudiantes sean capaces de:

- Iniciar la escritura de un plan de desarrollo que contemple la programación de los distintos componentes que intervienen en el juego (Super:bit y Carta:bit)
- Programar la primera versión funcional del mismo considerando dicho plan.
- Valorar el desarrollo incremental como una forma de programación de versiones del juego que funcionan, pero son incompletas.

Coordinación dupla pedagógica**Decisiones conjuntas entre DA y DR:**

- En función del tiempo y las características del grupo, decidir la dinámica de trabajo para la actividad de escritura del Plan de desarrollo. Tal vez se pueda, primero, trabajar en pequeños grupos y en la puesta en común, acordar un plan común. El conocimiento del DA para conformar pequeños grupos será muy valioso.

Decisiones del DA

- El modo de abordar la actividad de aula y todo trabajo previo con el contenido matemático, permitirá a los estudiantes llegar con fluidez a etapas siguientes que requieren explicitar un algoritmo para decidir cuándo una fracción es equivalente.

Información que necesita tener el DR:

- Experiencias de los estudiantes en otras tareas de planificación o escritura de acciones anticipatorias para llevar a cabo, que le permitan al DR vincularla con el Plan de desarrollo.

[Avance Etapa 3](#)

AULA ↓**Práctica con el contenido matemático****Propósitos mínimos**

- Alentar a la recuperación y registro de los cálculos que realizaron para jugar el juego en papel en la etapa anterior.

Propósitos óptimos

- Reforzar la práctica de operaciones con el contenido elegido resolviendo actividades digitales.
- Propiciar un espacio para analizar y comparar distintas estrategias de cálculo.

Notas para el DA ↓**Operaciones matemáticas para jugar.**

La experiencia de jugar en papel dejó en evidencia que cada jugador o equipo necesitó hacer alguna operación matemática para poder jugarlo. Se sugiere recuperar la revisión de las respuestas del juego de la etapa anterior para debatir y registrar las estrategias de cálculo que pensaron al tomar la decisión de dar vuelta la fracción o no hacerlo (siguiendo el caso del ejemplo, con fracciones equivalentes).

En el caso que el DA considere necesario reforzar la práctica y posterior análisis de distintas operaciones con el contenido matemático elegido, pueden acceder a resolver propuestas de Matific.

A continuación, algunos enlaces a actividades digitales sugeridas:

[Crear fracciones equivalentes](#)

[Fracciones equivalentes en la recta numérica](#)

[Números mixtos equivalentes en la recta numérica](#)

Si surgen nuevas estrategias o modificaciones a las ya registradas, se sugiere dejar evidencia en el foro.

VC ↓

Plan de desarrollo y primera versión funcional del juego

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



Desafío

Programar a Super:bit para que envíe el mensaje INICIAR PARTIDA y a Carta:bit para que al recibirlo muestre una fracción al azar.

1. Inicio (10 min)

¿Cómo ayudan las especificaciones que escribieron en la clase pasada para saber por dónde empezar a programar? ¿Es suficiente tener escrito el comportamiento de cada placa para realizar la programación del juego? ¿Por qué?

El DR recupera las especificaciones del juego de la VC anterior, en la que detallaron el comportamiento de las placas Super:bit y Carta:bit. Se brinda tiempo para leerlo y concluir que a partir de esta especificación, es necesario pensar un plan (paso a paso) para programarlas y lograr que el juego automático funcione. En este contexto, el DR argumenta y justifica la elaboración paulatina de una planificación (Plan de desarrollo) que servirá de guía durante todo el proyecto.

2. Desarrollo (25 min)

Plan de desarrollo

¿Qué es lo primero que deben programar? ¿Por qué? ¿Qué debe suceder luego? ¿Cuáles son los siguientes pasos a programar para que el juego se complete por etapas? ¿Sólo en ese orden pueden hacerlo? ¿Por qué?

El DR habilita un espacio para que los estudiantes identifiquen características que deben ir agregando a las placas. A medida que toman decisiones sobre el orden de los pasos a programar, el DR va completando el [documento](#) que contiene las especificaciones del juego.

Las preguntas que se presentan en este apartado no son definitivas. El DR puede realizar otras similares del tipo “¿Qué debe suceder cuándo...?”. Por ejemplo, “¿Qué debe hacer la Carta:bit cuando recibe el mensaje INICIAR PARTIDA?": “debe mostrar una fracción al azar”. Se puede identificar a *mostrar fracción* como una tarea o problema, y la característica *al azar*, como una

subtarea o subproblema a resolver en un siguiente paso.

De este modo, el DR guía y argumenta la elaboración de una planificación en cuyo proceso se valorará: la identificación de un **problema a resolver y la división en subproblemas**; y el **desarrollo incremental** de lo planificado; es decir, que en cada etapa se agreguen sucesivas características que hacen al producto final, pero de manera tal que al completarlas, se obtengan versiones funcionales y sucesivamente más completas del juego. En el [Anexo 2](#) pueden acceder a un ejemplo de Plan de Desarrollo.



Atención

Puede ocurrir que el plan esté incompleto, o que no siga la secuencia adecuada. Sin embargo, se debe respetar la propuesta de los estudiantes y dejar que aborden los problemas “del modo equivocado” para que puedan identificar tareas que faltan describir en su plan. De este modo se apropiarán de la planificación. Este **plan de desarrollo** está pensado para que sea **dinámico**, se puede ir modificando acorde a las dificultades que encuentren los estudiantes para llevar a cabo lo planificado. Cada cambio iterativo que se realice sobre el plan para luego continuar con el desarrollo, favorece la autonomía en la forma de pensar la resolución y permite que apliquen esta forma de trabajo en otros proyectos.

Desafío de la VC

¿Cómo deben comportarse las placas Super:bit y Carta:bit según el Plan de desarrollo? ¿Qué es lo primero que hay que programar? ¿Cuál es el segundo paso?

El DR presenta el desafío y habilita un tiempo para que los estudiantes comparen la relación entre el plan de desarrollo y el desafío que propone la

etapa para decidir si es necesario o no ajustar el plan. De ser necesario, realizan las modificaciones.

Parte 1: Carta:bit debe mostrar una fracción cuando reciba el mensaje de INICIAR PARTIDA

¿Cómo pueden programar la Super:bit para que envíe el mensaje INICIAR PARTIDA a la Carta:bit? ¿Cómo es posible que ésta lo reciba? ¿Qué bloques conocidos necesitan usar? ¿Qué bloques pueden combinar para que se muestre una fracción que ustedes decidan?

El DR realiza preguntas que guíen a los estudiantes a recuperar la programación de los ejercicios de la Etapa 1 y reconozcan la necesidad del uso de la categoría Radio para resolver la programación de esta parte. Oportunamente los estudiantes experimentaron con el envío de mensajes numéricos y de tipo cadenas.

Es posible que surja la inquietud de cómo representar la fracción en la pantalla de la Carta:bit. En este caso el DR interviene para proponer una combinación de bloques: *mostrar número* y *mostrar cadena*. Cabe aclarar que en este primer paso, son los estudiantes los que eligen qué fracción debe mostrar su placa.

Parte 2: Carta:bit debe mostrar una fracción al azar

Según el plan de desarrollo, la fracción debe ser al azar ¿qué estrategia pueden escribir para completar este desafío? ¿Qué modificación deben realizar en su programación para completar esta tarea? ¿Qué bloque pueden usar para que el programa responda a la estrategia escrita?

Al recuperar el plan de desarrollo se identifica que falta resolver un subproblema: que la fracción que muestra la Carta:bit sea al azar.

El DR incentiva a los estudiantes a analizar de qué manera consiguieron que la placa muestre una fracción, y a partir de allí, identificar qué se debe modificar. El DR propone hacer una puesta en común para explicitar la estrategia y que quede anotada y visible para la instancia de programación. Por ejemplo, lo programado hasta el momento puede representarse de la siguiente manera:

CARTA:BIT

Al recibir cadena "INICIAR PARTIDA"

Mostrar una fracción :

Mostrar un número (3)

Mostrar /

Mostrar un número (4)

Se puede modificar de esta forma:

CARTA:BIT

Al recibir cadena "INICIAR PARTIDA"

Mostrar una fracción al azar:

Mostrar un número al azar entre 1 y 10.

Mostrar /

Mostrar un número al azar entre 1 y 10.

Es recomendable que el DR deje registro de diversas anotaciones que se irán completando a medida que avancen en el desarrollo del proyecto. Se sugiere hacerlo en un documento único del estilo [Anotaciones de Cartas](#).

Programación

El DR les pide que identifiquen qué parte de la programación deben modificar para que refleje la nueva estrategia. Los anima a explorar la categoría *Matemática* en MakeCode para encontrar un bloque que permita reemplazar los números fijos de su fracción, en números aleatorios. Si es necesario, el DR guía a reconocer que, gracias a la forma de *bloque escoger al azar*, es posible incorporarlo dentro del bloque *mostrar número*.



Atención

Es recomendable que a medida que avanzan en la programación en el entorno <https://makecode.com/multi#> de Makecode, los estudiantes vayan probando el juego, previa descarga del programa en la placa que corresponda.

★ Importante

Antes del cierre, considerar las recomendaciones respecto a la importancia de compartir los avances del proyecto en el **Foro de evidencia** y realizar las **Actividades interactivas**.

3. Cierre (10 min)

¿Qué ventaja tiene escribir un Plan de Desarrollo? ¿Es necesario que terminen de programar todo lo que planificaron para poder jugar? ¿Cómo puede jugarse con lo que programaron en esta clase?

El Plan de Desarrollo permite tener noción de los pasos a seguir para programar el funcionamiento de las placas, y en qué orden debe llevarse a cabo cada tarea. El Plan de desarrollo con el que se trabaja en esta propuesta, pretende ser flexible y dinámico, permitiendo a los estudiantes modificarlo a medida que detectan errores en su lógica. Se acompaña a los estudiantes a reconocer que no es necesario programar el juego de manera completa para poder jugarlo, sino que cada etapa permitirá tener una versión funcional. Por ejemplo, en la versión de esta clase, cada jugador tiene una fracción al azar en su Carta:bit, a la que puede comparar con una fracción escrita en el pizarrón.

La particularidad de hacer que un juego (o cualquier otro programa) sea funcional al final de cada versión (etapas), implica un desarrollo incremental. La clave para este proceso consiste en empezar con la implementación sencilla de un conjunto de requisitos y mejorar iterativamente la evolución secuencial de versiones hasta que el juego esté completo. En cada iteración se realizan modificaciones en las que se añaden nuevas funcionalidades y se revisa el plan de desarrollo.

Registro en Crea

El DR publica el registro con las notas y reflexiones de los intercambios en el **Registro Común**. Se publica el registro de los intercambios y la **especificación** a la que arribaron.

Queda a disposición de todo el curso la planificación elaborada.

Se sugiere registrar las reflexiones del cierre, junto con material gráfico que el DR podría haber utilizado para ilustrarlas.

Invitar a los estudiantes a resolver la [actividad interactiva](#) de la etapa en las que se repase la noción de desarrollo incremental.

La Yapa: Propuestas para seguir en casa

¿Cómo harías para que en ambas placas, Super:bit y Carta:bit anuncien el inicio de la partida?



ETAPA 4 ↓ Optimizando la primera versión

En esta etapa del proyecto los estudiantes incorporan una nueva herramienta, procedimientos, para completar la primera versión del juego.

En el aula, analizan la letra de una canción, identifican partes que se repiten y deciden cómo nombrarlas.

En la VC, se presenta la noción de procedimiento como expresión de la estrategia del programa para que la Carta:bit muestre varias veces la fracción al azar.

Objetivos

Se espera que los estudiantes sean capaces de:

- Acercarse a la noción de estrategia como una manera de pensar una solución a un problema.
- Utilizar procedimientos como expresión de la estrategia para crear un programa.
- Valorar la legibilidad de los programas.

Coordinación dupla pedagógica

Decisiones conjuntas entre DA y DR:

- Qué canción van a elegir. El DA, además de considerar los gustos musicales de los estudiantes, debe asegurarse que el estribillo de la canción elegida se repita varias veces, ya que ésta será la clave para que el DR introduzca la noción de "procedimiento" durante la VC.

Decisiones del DA

- La dinámica de trabajo para realizar el análisis de la letra de una canción.

Información que necesita tener el DR:

- Cómo se desarrolló la actividad de aula.

[Avance Etapa 4](#)

AULA ↓ Aprendemos con canciones

Notas para el DA ↓



Propósitos mínimos

- Brindar el espacio para analizar la letra de una canción con estribillo que el DR retomará en la VC.
- Favorecer la identificación de sus partes y nombrarlas de un modo representativo.

Propósitos óptimos

- Articular con música la realización de esta actividad.

Análisis de la letra de una canción

Se brinda a los estudiantes un espacio para que, a partir de escuchar, leer o cantar la letra de una canción puedan analizarla prestando atención si aparece con cierta regularidad alguna estrofa idéntica.

Supongamos que van a cantarla por grupos. Un grupo canta la primera estrofa, otro la que sigue y así hasta terminar. *¿Qué nombre le pondrían a cada estrofa para que ningún grupo se equivoque en el orden al momento de cantarla? ¿Hay alguna parte de la canción que se repite? ¿Cuántas veces? ¿Cómo se llama en música a ese tipo de estrofas?*

El DA puede registrar en el pizarrón los nombres elegidos por los estudiantes para llamar a cada estrofa en el orden de la letra original.

¿Y si variamos la canción?

Usando la misma letra, se le pide a los estudiantes que armen una nueva canción para cantar ambas versiones. El DA utilizando los mismos nombres que eligieron para cada parte, los ordena según las sugerencias de los estudiantes, sin dejar afuera ninguna.

En el [Anexo 3](#) se sugieren letras de canciones que pueden servir para realizar esta actividad y un ejemplo de cómo resolverla.

VC ↓

Una fracción al azar, varias veces

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

💡 Desafío

Programar la Carta:bit para que al presionar un botón vuelva a mostrarla.

1. Inicio (10 min)

¿Qué canción escucharon o escribieron en el aula? ¿Cómo nombraron a cada estrofa? ¿Escribieron una nueva versión de la canción? ¿Fue necesario escribir toda la letra de nuevo?

El DR pregunta sobre la actividad del aula, recuperando el registro en el foro. En el caso de no haberse realizado la actividad, reformular estas preguntas para que los estudiantes identifiquen y nombren las distintas estrofas en una canción (ver [Anexo 2](#)).

El DR acompaña a los estudiantes a reconocer que: agrupar los versos de la canción que pertenecen a una misma estrofa, con un nombre representativo (Estrofa 1, Estribillo), trajo consigo el beneficio de mejorar la legibilidad y ahorrar escritura.

En esta instancia, el DR anticipa que en programación también se pueden realizar agrupamientos de instrucciones que estén asociadas a una tarea específica y asignarles un nombre. A esta forma se la denomina **procedimiento o función** (dependiendo el entorno que se utilice) y pueden usarse en uno o varios segmentos de un programa, como sucede con el estribillo de la canción.

Al igual que en la actividad de la canción, usar funciones mejora la legibilidad y ahorra la escritura de código.

2. Desarrollo Primera parte (30 min)

Entre los jugadores, hay uno muy distraído y no alcanza a ver la fracción que mostró su Carta:bit. ¿Cómo puede verla nuevamente?

El DR pone en discusión esta posible situación y regula el intercambio de los estudiantes. Propone retomar el Plan de desarrollo y completarlo en la sección que corresponde con el propósito de optimizar el juego. En este contexto se presenta el desafío.

Estrategia para resolver el desafío

La Carta:bit ya muestra una fracción al azar cuando recibe el mensaje INICIAR PARTIDA ¿qué se necesita hacer para volver a mostrar esa fracción cuando se presiona un botón? ¿Qué estrategia pueden escribir para resolverlo?

El DR modera los comentarios de los estudiantes y toma nota de las sugerencias para ampliar la estrategia anterior. Puede suceder que algunos estudiantes propongan lo siguiente:

Al presionar el botón A

Mostrar una fracción al azar:

Mostrar un número al azar entre 1 y 10.

Mostrar /

Mostrar un número al azar entre 1 y 10.

Aunque esta forma es equivocada, porque se estaría generando una nueva fracción al azar en vez de repetir la que ya se había mostrado, el DR puede optar por pedirles que la programen en sus proyectos para verificar que no soluciona el problema planteado. Luego, a partir del error, reflexionar con los estudiantes por qué no funciona esta estrategia.

Se busca poner de manifiesto, si no hubiera surgido espontáneamente (en base a experiencias de propuestas anteriores), la necesidad de almacenar los valores generados al azar para poder recuperarlos luego. Al reconocer la necesidad de almacenar el numerador y el denominador, se recupera el concepto de **variable**. Se emplea una estrategia similar a esta:

CARTA:BIT

Al recibir cadena "INICIAR PARTIDA"

Almacenar una fracción al azar:

Almacenar un número al azar en *numerador*

Almacenar un número al azar en *denominador*

Mostrar fracción almacenada:

Mostrar numerador

Mostrar /

Mostrar denominador

Al presionar el botón A

Mostrar fracción almacenada:

Mostrar *numerador*

Mostrar /

Mostrar *denominador*

Funciones en la estrategia

¿Cómo se puede aplicar funciones para este caso? ¿Cuál es el beneficio de implementarlas?

El DR anima a los estudiantes a modificar y simplificar la estrategia anterior, utilizando funciones. Por ejemplo:

CARTA:BIT

Al recibir cadena "INICIAR PARTIDA"

Almacenar fracción al azar:

Mostrar fracción almacenada:

Al presionar el botón A

Mostrar fracción almacenada:

Función Almacenar fracción al azar

Almacenar un número al azar en *numerador*

Almacenar un número al azar en *denominador*

Función Mostrar fracción almacenada

Mostrar *numerador*

Mostrar /

Mostrar *denominador*

Recordar que es conveniente que el DR unifique el registro de anotaciones, etapa por etapa, en un único documento del estilo [Anotaciones de Cartas](#).

Programación

Una vez escrita la estrategia y analizada entre DR y estudiantes, se continúa con la exploración del entorno para reconocer la categoría Funciones. Se habilita tiempo para que los estudiantes prueben cómo definir funciones. De ser necesario el DR interviene indicando los pasos para este procedimiento.

Los estudiantes avanzan en la programación, **trabajando sobre el programa que elaboraron en la etapa anterior y definiendo las funciones que especificaron en su estrategia**. Si los estudiantes propusieron sólo la función *Mostrar fracción almacenada*, es válido permitirles definir sólo ésta, ya que es la que pueden observar en dos partes de la estrategia.

**Atención**

Los números que se generan al azar deben almacenarse en variables inmediatamente después de recibir la señal de inicio de partida. Se deben crear dos variables (para el numerador y para el denominador). En el entorno de MakeCode se almacenan los números combinando los bloques de la siguiente manera:

**Procedimientos en Pilas Bloques**

Se propone realizar un desafío en Pilas Bloques: [Campeón desordenado](#) para reforzar la relación entre estrategias y el uso de procedimientos.

Para resolverlo, con guía del DR los estudiantes hacen una puesta en común de la estrategia de solución que consideran apropiada (antes de empezar a programar). Luego, la representan creando los procedimientos en el entorno. Por último, incorporan las primitivas a los procedimientos definidos. ([Anexo 4](#)).

**Procedimientos y Funciones**

Dado que MakeCode llama "Funciones" tanto a funciones como a procedimientos, no nos interesa hacer esta distinción de nombre en la VC.

**Atención**

Recordar usar la opción <https://makecode.com/multi#> de Makecode, para evitar errores operativos.

★ Importante

Antes del cierre considerar las recomendaciones respecto a la importancia de compartir los avances del proyecto en el **Foro de evidencias** y realizar las **Actividades interactivas**.

3. Cierre (10 min)

Estrategia - Funciones/procedimientos - Programas más fáciles de leer y modificar

¿Con qué dificultades se encontraron al resolver el desafío? ¿Qué herramientas conocidas utilizaron para resolverlo? ¿Cómo mejoraron el programa? Comparando la estrategia que resuelve el desafío y la estrategia que utiliza funciones ¿a qué conclusiones llegan?

¿Probaron el juego? ¿Se imaginan cómo continuará el desarrollo? ¿Pueden prever alguna parte del programa que habrá que modificar en el futuro? ¿Y algún procedimiento/función nueva?

Con estas preguntas, se busca reforzar la importancia del diseño de la **estrategia** antes de comenzar a programar, para **pensar en el problema** y **cómo resolverlo**, pero sin preocuparse todavía por detalles de programación o del entorno. En este sentido, es muy útil identificar subtareas o subproblemas más simples que al combinarlos, constituyan la solución final.

Las funciones (o procedimientos) permiten expresar la estrategia dentro del programa. Esto facilita la escritura y la legibilidad del código. De esta manera, si es necesario corregir el programa, es mucho más fácil identificar dónde se encuentra el error y realizar los ajustes pertinentes dentro de la función que corresponda. En aquellos programas donde una función se utiliza varias veces, es suficiente corregir las instrucciones de la misma, para observar la mejora en todo el programa.

Los estudiantes reconocen que la programación de esta etapa permitió optimizar el juego, sin que haya un cambio en la funcionalidad del mismo. Es decir, se juega de la misma manera que en la etapa anterior: cada jugador tiene una fracción al azar en su Carta:bit, a la que puede comparar con una fracción fija escrita en el pizarrón. El docente incentiva a que jueguen con las placas, en la VC o en la hora de aula, dependiendo del acuerdo que haya realizado con el DA.

Registro en Crea

El DR publica las notas y reflexiones de los intercambios en el **Registro Común** y solicita a los estudiantes a compartir en el foro de la etapa, las modificaciones realizadas al Plan de desarrollo y los avances de proyecto realizados en Make.code



Se invita a los estudiantes a resolver la [actividad interactiva](#) de la etapa en las que se repase la noción de procedimiento.

La Yapa: Propuestas para seguir en casa

Para seguir en casa te proponemos que resuelvas este nuevo desafío de Pilas Bloques [Tito enciende las luces](#)
¿Cuál fue tu estrategia para resolverlo?



ETAPA 5 ↓**Controlando las fracciones al azar**

En esta etapa, se limita el rango de valores posibles para el numerador y el denominador, aumentando la probabilidad que aparezca una fracción equivalente en las placas.

En el aula, se analizan conjuntos de valores posibles para el numerador y el denominador que generen más fácilmente una fracción equivalente a una fracción fija.

En la VC, los estudiantes incorporan estos conjuntos al programa utilizando listas.

Objetivos

Se espera que los estudiantes sean capaces de:

- Aproximarse a la noción de listas como una herramienta para representar un conjunto de valores en un programa.
- Utilizar una lista para almacenar y representar un conjunto de valores.

Coordinación dupla pedagógica**Decisiones conjuntas entre DA y DR:**

- En qué espacio jugarán con las placas para probar la última versión programada.

Decisiones del DA

- La decisión acerca de cuál es la fracción más conveniente para comparar y hacer cálculos.

Información que necesita tener el DR:

- Cómo se abordó la clase de aula para llegar al resultado de la lista de numeradores y denominadores que permitan generar fracciones equivalentes a la fracción elegida.

[Avance Etapa 5](#)

AULA ↓**Numeradores y denominadores: ¿cuáles?****Propósitos mínimos**

- Promover la prueba del juego automático para comprobar que la mayoría de las fracciones que se obtienen no son equivalentes a la escrita en el pizarrón.
- Generar el espacio para que los estudiantes calculen y registren qué numeradores y fracciones son equivalentes a una fracción que elija el DA.

Propósitos óptimos

- Alentar la estimación de la probabilidad de obtener una fracción equivalente a la fracción elegida.
- Promover la problematización con fracciones equivalentes.

Notas para el DA ↓**Listado de numeradores y denominadores para generar fracciones equivalentes**

Se brinda a los estudiantes un espacio para que, a partir de una fracción escrita por el DA en la pizarra, comiencen a jugar una partida utilizando las placas.

Se evidencia que la mayoría de las fracciones generadas al azar por cada Cartas:bit no es equivalente. Recuperando las estrategias de cálculo registradas en la Etapa 3, anotan en una lista qué numeradores y denominadores necesitan para crear fracciones equivalentes a la elegida por el DA. Por ejemplo, $\frac{1}{2}$.

En este caso, los listados serían:

Numeradores: 1-2-3-4

Denominadores: 2-4-6-8

Si el DA considera necesario, puede repetir la práctica varias veces, eligiendo otras fracciones para reforzar la práctica de los cálculos que necesitan realizar. Están disponibles los siguientes recursos:

[Equivalencia de fracciones](#)

[REA Ceibal](#)

Es recomendable que los listados de numeradores y denominadores, y las conclusiones a las que abordaron, se compartan en el foro de evidencias de esta etapa.

VC ↓ Creando listas

En toda la VC tomaremos $\frac{1}{2}$ como ejemplo de fracción para comparar con las fracciones generadas por las Cartas:bit.

1. Inicio (15 min)

¿Cómo resultó la experiencia de jugar la última versión programada? ¿Cuántas veces lo probaron? ¿Cuántas fracciones equivalentes les mostró en esas pruebas? ¿Les parece un juego desafiante? ¿Por qué?

Se realiza un breve intercambio sobre la evidencia de que la mayoría de las fracciones generadas al azar por Carta:bit, no son equivalentes a $\frac{1}{2}$. El DR invita a los estudiantes a suponer las razones por las cuales esto sucede, y luego presentarles un juego que los ayudará a identificar una herramienta necesaria para resolver el desafío de esta etapa.

Crear, recuperar, modificar listas

El DR comienza la actividad dividiendo al curso en dos equipos para empezar a jugar. Solo necesitan una hoja y un lápiz para anotar.

Primer momento:

- 1- El DR pregunta: ¿Qué **elementos** llevan a la escuela todos los días?
- 2- Mientras los estudiantes responden, el DR va registrando las respuestas (en forma clara, ordenada y con un color) en un documento, compartiendo pantalla para que todos puedan verlo. (Se recomienda brindar un tiempo acotado para generar un ambiente lúdico).
- 3- A la voz de "Listo, preparado, ya", durante un breve tiempo, el DR oculta el listado y comienza la competencia.
- 4- Cada equipo debe recordar y escribir en una hoja la mayor cantidad de elementos posible en el tiempo que los docentes determinen.

Gana la partida el equipo que recuerde y escriba la mayor cantidad de elementos de los registrados por el DR en el inicio del juego.

Desafío

Programar la Carta:bit para qué, a partir de una lista de números, elija el numerador y denominador que va a mostrar.

Segundo momento:

- 1- El DR cambia la pregunta: ¿Qué **útiles escolares** llevan a la escuela todos los días?
- 2- Luego muestra la lista de elementos y marca los útiles escolares ya nombrados por los estudiantes, pero con otro color; o agrega otros nuevos que ellos le sugieren en esta instancia del juego.
- 3- Se repite la dinámica del primer momento.

Gana la partida el equipo que recuerde y escriba la mayor cantidad de útiles escolares de los registrados por el DR.

Al finalizar el juego, el DR muestra su listado, solicitando a cada equipo que observen sus propios listados escritos en la hoja y junto al DA, los guían a reconocer que:

1. Al comienzo del juego cada equipo solo tenía su hoja en blanco. Luego, fueron escribiendo los elementos que recordaron **creando** una lista.
2. Para crear esta lista tuvieron que **recuperar** elementos anotados por el DR.
3. Al seleccionar y marcar cuáles de esos elementos eran útiles escolares y agregar otros nuevos, **modificaron la lista** original.

De esta manera, el DR introduce la noción de listas como una herramienta de programación que se utiliza para almacenar, recuperar y modificar datos homogéneos.

2. Desarrollo (30 min)

Estrategia

Comprobaron que es muy difícil que las fracciones generadas al azar por la Carta:bit sean equivalentes a $\frac{1}{2}$. ¿Escribieron listas de valores para el numerador y para el denominador en el aula? ¿Cómo pueden utilizar estas listas para que la Carta:bit muestre, con mayor probabilidad, una fracción equivalente? ¿Qué parte de la estrategia deben modificar para que tanto el

numerador como el denominador tomen los valores de sus listas?

Los estudiantes recuperan las listas que generaron para el numerador y el denominador en la clase de aula. El DR muestra la estrategia que elaboraron en la VC anterior (guardada en el documento de Notas) y ayuda a los estudiantes a identificar que deben modificar *Función Almacenar fracción al azar*. Es un buen momento para señalar que el hecho de tener en claro la estrategia, hizo que esta modificación se haya podido identificar fácilmente.

Función Almacenar fracción al azar:

Almacenar ~~número al azar~~ número al azar entre {1, 2, 3, 4} en numerador
Almacenar ~~número al azar~~ número al azar entre {2, 4, 6, 8} en denominador

Tener en cuenta que para facilitar la tarea de registro del DR de las distintas anotaciones mientras avanza el desarrollo del proyecto, es recomendable hacerlo en un documento común del estilo [Anotaciones de Cartas](#).

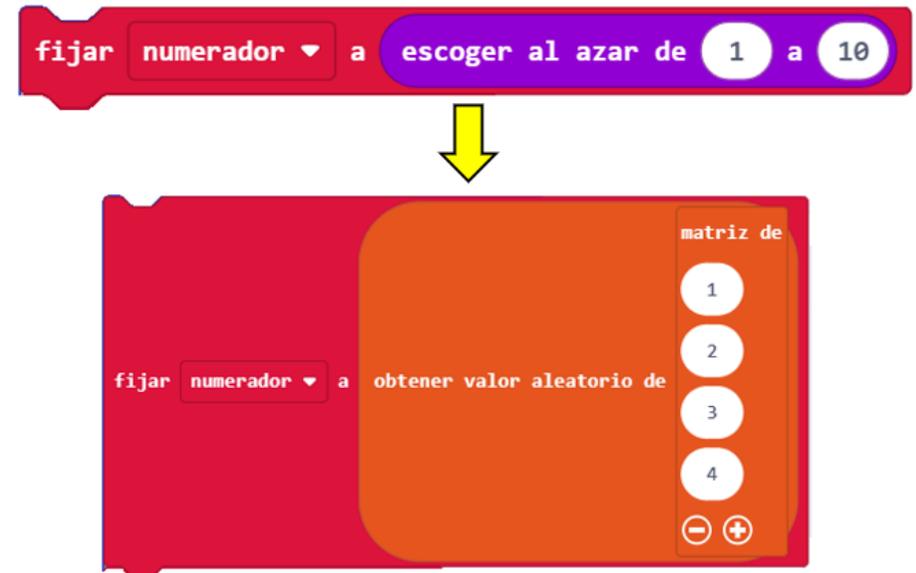
Programación

Los estudiantes abren sus proyectos e identifican qué parte de la programación deben modificar utilizando la nueva estrategia. Si fuera necesario, el DR los orienta a identificar la función que contiene las instrucciones para almacenar el numerador y el denominador.

Se orienta a los estudiantes a observar los bloques de la categoría “Arreglos”. Debido a las complicaciones en la traducción, el DR debe guiarlos activamente para que incorporen los bloques necesarios y construyan la lista correspondiente para cada variable (numerador y denominador).

El DR muestra cómo modificar el bloque *Fijar numerador*, quitándole el *escoger al azar de 1 a 10*, e incorporando el bloque de la categoría “Arreglos” *obtener valor aleatorio*. Al final de éste, debe incorporarse el bloque *matriz vacía*, el cual permitirá crear la lista de valores para el numerador. También deberán usar un bloque de *número*, de la categoría matemática, para rellenar el primer valor de la matriz (lista).

Los estudiantes pueden completar los valores restantes con los que trabajaron en el aula y continuar por su cuenta para modificar los bloques relacionados al denominador.



Al finalizar la programación que resuelve el desafío, el DR propone a los estudiantes jugarlo y dejar por escrito en el Plan de desarrollo que la inclusión de esta nueva herramienta de programación (listas) ha permitido la optimización de la primera versión del juego.

★ Importante

Antes del cierre considerar las recomendaciones respecto a la importancia de compartir los avances del proyecto en el **Foro de evidencias** y realizar las **Actividades interactivas**.

3. Cierre (10 min)

Listas como conjuntos

¿Cuál es la función de las listas en sus programas? ¿Si quisieran agregar un valor nuevo para el denominador, cómo lo harían? ¿Esto provocaría algún cambio importante en el programa?

La función que cumplen las listas en esta etapa, es permitir que las variables *numerador* y *denominador* tomen un valor específico de un conjunto limitado de

datos. La cantidad de elementos de una lista no es importante, por eso, si quisieran agregar un valor nuevo para el denominador, bastaría con incorporarlo a la lista y el resto del programa no sufriría ningún cambio.

En este intercambio, se busca aproximarse a una noción de lista como una manera de representar en los programas un conjunto de valores. Las listas pueden crearse y cargarse con distintos valores homogéneos, para recuperarlos y modificarlos de acuerdo a la necesidad del programa.

Listas en otros programas

De los programas que conocen, ¿se les ocurre cuáles usan listas? ¿Por qué consideran que estos programas las usan? ¿Qué función cumplen en ellos?

El DR alienta a que los estudiantes identifiquen situaciones de uso de otros programas en los que haya un conjunto de valores en juego. Por ejemplo, las personas que dieron like en una publicación; el estado en una aplicación de mensajería (“en línea”, “ausente”, “ocupado”...); etc.

En el caso de los likes, las listas sirven como una retroalimentación para el autor de la publicación, le permite saber a quiénes les resultó interesante o relevante lo que publicó. En el caso de la mensajería, elegir un estado de la lista, permite que sus contactos conozcan la disponibilidad del usuario.

Registro en Crea

El DR publica las notas y reflexiones de los intercambios en el **Registro Común** y solicita a los estudiantes a compartir en el foro de la etapa, las modificaciones realizadas al Plan de desarrollo y los avances de proyecto realizados en Make.code



Invitar a los estudiantes a resolver la [actividad interactiva](#) de la etapa en las que se repase la noción de listas.

La Yapa: Propuestas para seguir en casa

¿Se te ocurre cómo usar la herramienta **lista** para que en el momento en que se enciende la placa aparezca una frase al azar?



ETAPA 6 ↓ Decisión de equivalencia

En esta etapa, se le agrega a la Carta:bit la capacidad de reconocer si la fracción generada es equivalente a $1/2$. Se cuenta con una nueva versión del juego, más automatizada que la anterior, en la que el usuario solo compara su respuesta con la de la Carta:bit.

En el aula, se recuperan con el DA las nociones matemáticas necesarias para construir un algoritmo que determine si una fracción es equivalente a otra.

En la VC, los estudiantes implementan este algoritmo en las placas.

Objetivos

Se espera que los estudiantes sean capaces de:

- Implementar el algoritmo diseñado utilizando los bloques disponibles en MakeCode.
- Utilizar funciones para agregar legibilidad al programa y simplificar la estructura de bloques anidados.

Coordinación dupla pedagógica

Decisiones conjuntas entre DA y DR:

- Dudas o aportes de los niños que hayan surgido y puedan ser relevantes a la hora de trabajar en la VC.
- El método para decidir la equivalencia de fracciones a implementar, en función de los bloques disponibles en MakeCode y las operaciones con las que estén más familiarizados los estudiantes. Si conocen más de uno, puede ser una decisión de cada grupo.

Decisiones del DA

- El recorrido didáctico que se llevará a cabo para alcanzar los propósitos de la instancia de aula.

Información que necesita tener el DR:

- El grado de fluidez con el que los estudiantes manejan las estrategias para decidir la equivalencia de fracciones.

[Avance Etapa 6](#)

AULA ↓ Equivalencia de fracciones

Notas para el DA ↓



Propósitos mínimos

- Motivar la recuperación de las estrategias de cálculo matemático realizadas en la etapa 3 para definir un algoritmo que determine cuándo una fracción es equivalente a $\frac{1}{2}$.

Propósitos óptimos

- Propiciar un espacio para definir un nuevo algoritmo a partir de otra estrategia de cálculo matemático.

¿Cómo decidir si dos fracciones son equivalentes?

En esta instancia los estudiantes deberían poder resolver con claridad cuándo dos fracciones son equivalentes y dejar registro del paso a paso (algoritmo) realizado para demostrarlo.

Pueden proponer un algoritmo general para decidir si dos fracciones son equivalentes, o un algoritmo específico para el caso de fracciones equivalentes a $\frac{1}{2}$.

Si los estudiantes crean distintos algoritmos para determinar la equivalencia de fracciones, se sugiere dejar evidencia en el foro de esta etapa, ya que el proceso realizado puede ser de utilidad para el desarrollo de la VC.

VC ↓

Estrategia para decidir la equivalencia

En toda la VC tomaremos $\frac{1}{2}$ como ejemplo de fracción para comparar con las fracciones generadas por las Cartas:bit.

1. Inicio (5 min)

¿Cuál es el paso a paso que realizaron para determinar la equivalencia entre dos fracciones?

Se recuperan las notas del foro realizadas en el aula y se brinda un tiempo para que los estudiantes comenten los cálculos y el algoritmo que desarrollaron para determinar si una fracción es equivalente a $\frac{1}{2}$.



Atención

Algunas opciones para definir la equivalencia:

- Dividir el numerador por el denominador y comparar con 0.5.
- Dividir el denominador por el numerador y comparar con 2.
- Multiplicar el numerador por 2 y comparar con el denominador.

2. Desarrollo (30 min)

Estrategia

¿Cómo deben modificar la estrategia para que la Carta:bit interprete que la partida finalizó? ¿Qué estrategia pueden escribir para comprobar si la fracción al azar es equivalente a $\frac{1}{2}$? ¿Cómo puede informar la Carta:bit si son equivalentes o no lo son?

Se recupera el Plan de desarrollo para comenzar a abordar el desafío y organizar cómo lograr la nueva versión del juego. El DR señala que, para que el juego sea **completamente automatizado**, el jugador debe interactuar con la Carta:bit. El DR presenta el desafío y modera los comentarios y sugerencias de los estudiantes. Es importante escribir la estrategia en términos de

Desafío

Programar a Super:bit para que envíe el mensaje TERMINAR PARTIDA, y a Carta:bit para que al recibirlo muestre un ícono, si la fracción es equivalente; y otro, si no lo es.

operaciones que sean factibles de programar en las placas. Para esto, se pueden encontrar analogías entre: los términos matemáticos que usaron en el aula y los bloques disponibles en MakeCode para expresar estos cálculos.

También es importante recordar la necesidad de definir funciones para encapsular los distintos subproblemas que forman parte de la solución. Si fuera necesario, el DR orienta a los estudiantes para que expresen sus ideas en términos de la estructura "SI ENTONCES".

SUPER:BIT

Al presionar A:
Enviar "INICIAR PARTIDA"

Al presionar B:
Enviar "TERMINAR PARTIDA"

CARTA:BIT

Al recibir cadena
Si cadena = "INICIAR PARTIDA"
Almacenar fracción al azar:
Mostrar fracción almacenada:
Si cadena = "TERMINAR PARTIDA"
Mostrar si es equivalente a $\frac{1}{2}$

Función Mostrar si es equivalente a $\frac{1}{2}$

Si numerador * 2 = denominador:

Mostrar cara feliz

si no

Mostrar cara triste

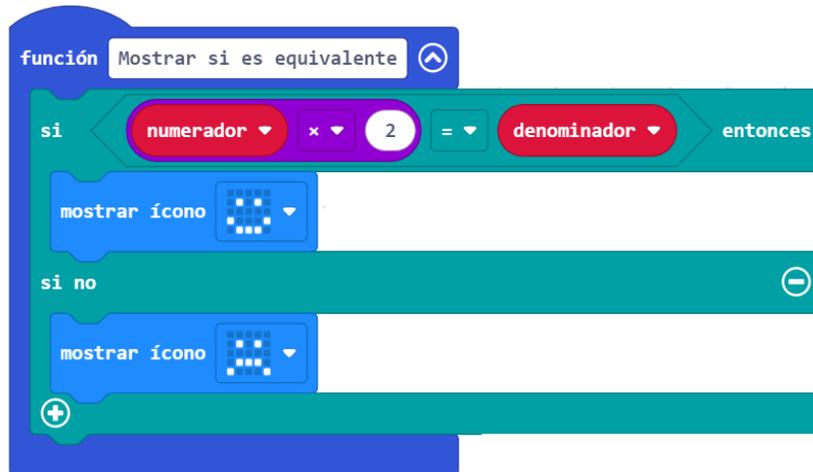
Recordar que el DR irá dejando registro de las diversas anotaciones a medida que avancen en el desarrollo del proyecto. Se sugiere hacerlo en un documento único del estilo [Anotaciones de Cartas](#).

Programación

El DR habilita el espacio de programación. Recuerda a los estudiantes que partiendo de la estrategia elaborada deben pensar **qué parte del programa deberán modificar y qué deberán agregar**.

En esta oportunidad, se debe modificar el programa de la Super:bit para que, además de enviar el mensaje de *Iniciar partida*, envíe el de *Terminar partida*. Por otro lado, en Carta:bit, es necesario utilizar un bloque condicional que permita diferenciar los mensajes que recibe de la Super:bit. Puede utilizarse un bloque condicional con un *si no*, o dos condicionales simples.

El DR retoma la herramienta de función para guiar a los estudiantes a crear una que contenga las instrucciones necesarias para realizar los cálculos que comprueben la equivalencia entre las fracciones. En esta sección es necesario combinar bloques de la categoría lógica (el condicional y uno de comparación), con uno matemático (para la multiplicación), e incorporar las variables según corresponda. Para la estrategia que se presenta en esta guía, la programación quedaría de la siguiente manera:



★ Importante

Antes del cierre considerar las recomendaciones respecto a la importancia de compartir los avances del proyecto en el **Foro de evidencias** y realizar las **Actividades interactivas**.

3. Cierre (10 min)

Los programas como herramienta de comunicación - Funciones y legibilidad

¿Qué se entiende con más facilidad: los programas o las estrategias? ¿Por qué? ¿A quién está dirigida cada una? ¿Para qué usaron las funciones al programar la solución? ¿De qué manera les facilitó el trabajo? ¿Y si quisieran analizar los programas de sus compañeros?

Se señala el rol de las funciones como un puente entre la estrategia (escrita y pensada en el lenguaje de las personas) y el programa (expresado en el lenguaje de las computadoras). Es importante resaltar el aspecto comunicacional de los programas, por esta razón es recomendable compartir las soluciones que construimos y analizar las construidas por otras personas para seguir aprendiendo. La creación de funciones y el modo de nombrarlas son un aporte fundamental a la legibilidad de los mismos. Aportan claridad y orden al programa, permitiendo que otras personas puedan entender el código cuando lo leen.

Registro en Crea

El DR publica las notas y reflexiones de los intercambios en el **Registro Común** y solicita a los estudiantes compartir en el foro de la etapa, las modificaciones realizadas al Plan de desarrollo y los avances de proyecto realizados en Make.code.



Invitar a los estudiantes a resolver la [actividad interactiva](#) de la etapa en las que se repase la noción de condicional.

La Yapa: Propuestas para seguir en casa

¿Cómo harías para que en ambas placas, Super:bit y Carta:bit anuncien el inicio de la partida?



ETAPA 7 ↓**Carta:bit corrige automáticamente**

La motivación de esta etapa es agregar una de las características fundamentales del juego: que el jugador interactúe con la Carta:bit y ésta indique si la respuesta fue correcta o incorrecta. De esta manera se obtiene una versión completamente automática del juego.

En el aula, se escribe una nueva versión de las reglas del juego.

En la VC, los estudiantes agregan a sus programas la interacción con el jugador para que indique su respuesta.

Objetivos

Se espera que los estudiantes sean capaces de:

- Escribir una regla general que describa el comportamiento de Carta:bit como respuesta a la intervención del usuario.
- Escribir una estrategia general que contemple todos los casos posibles.
- Programar Carta:bit para que su comportamiento dependa de la fracción al azar y de la interacción con el jugador.

Coordinación dupla pedagógica**Decisiones conjuntas entre DA y DR:**

- Dudas o aportes de los estudiantes que hayan surgido y puedan ser relevantes a la hora de trabajar en la VC.

Decisiones del DA

- El recorrido didáctico que se llevará a cabo para alcanzar los propósitos de la instancia de aula.

Información que necesita tener el DR:

- La forma en que se aborda el contenido del proyecto en el Aula, para relacionar el procedimiento utilizado con la programación en la VC.

[Avance Etapa 7](#)

AULA ↓ Nuevas reglas del juego

Notas para el DA ↓



Propósitos mínimos

- Promover la escritura de las reglas de la nueva versión del juego y compararlas con las anteriores.

Propósitos óptimos

- Alentar a la definición y práctica de estrategias de cálculo pensado para decidir rápidamente si una fracción es equivalente a $\frac{1}{2}$.

Nueva versión del juego, nuevas reglas.

La escritura de las reglas de la nueva versión del juego es una oportunidad para reforzar la noción de incrementalidad en el desarrollo, para que luego los estudiantes puedan identificar sus avances en programación con las mejoras en la dinámica del juego.

Pensando cálculos matemáticos

Con la motivación de resolver acertadamente el desafío que plantea Cartas:bit y en el menor tiempo posible, los estudiantes pueden desarrollar y poner en práctica estrategias para decidir si una fracción es equivalente a $\frac{1}{2}$ o no.

VC ↓ Interacción del jugador

Desafío

Programar la interacción del jugador con la Carta:bit para que ésta indique si la respuesta es correcta o incorrecta.

En toda la VC tomaremos $\frac{1}{2}$ como ejemplo de fracción para comparar con las fracciones generadas por las Cartas:bit.

1. Inicio (5 min)

¿Cómo debe continuar la programación del juego según el plan de desarrollo?
¿Qué componentes conocen de las placas que permitan detectar la interacción del usuario?

Se recupera el Plan de desarrollo para presentar el desafío. Se señala la importancia de que el jugador pueda interactuar con la Carta:bit para que sea un **juego completamente automatizado**. El DR recupera conceptos relacionados a los sensores como aquellos componentes que permiten el ingreso de información a la Carta:bit para que se comporte según lo programado.

2. Desarrollo (30 min)

Estrategia

Supongamos que el jugador decide que la fracción que observa en su Carta:bit es equivalente a $\frac{1}{2}$. ¿De qué manera se lo informa a su Carta:bit? ¿Y si decide que no es equivalente? ¿Cómo determina la Carta:bit si la respuesta es correcta o incorrecta? ¿Qué hay que tener en cuenta para que el programa evalúe la respuesta? ¿De qué manera la Carta:bit da a conocer al jugador si respondió correcta o incorrectamente? ¿Cómo pueden escribir una estrategia para que la Carta:bit responda al jugador si su respuesta es correcta?

El DR presenta el desafío y recupera las reglas del juego escritas en el aula. Los estudiantes deben decidir qué interacción usarán para responder si la fracción mostrada por la Carta:bit es equivalente a $\frac{1}{2}$ o no. A modo de ejemplo para esta guía se toma como interacción: *pantalla hacia arriba* para indicar que hay equivalencia - *pantalla hacia abajo* para indicar que no hay equivalencia.

Luego, para que la Carta:bit indique si la respuesta que dieron es correcta o

incorrecta, los estudiantes deben tener en cuenta tanto si la fracción es equivalente a $\frac{1}{2}$ como si la pantalla está hacia arriba o hacia abajo. El DR los puede guiar de distintas maneras para identificar esta particularidad que permitirá resolver el desafío.

Una opción es pedirle a los estudiantes que mencionen ejemplos de fracciones equivalentes a $\frac{1}{2}$ y anotarlas en un cuadro similar al que se muestra abajo. El DR le pide a los estudiantes que reconozcan, en función de la posición de la Carta:bit, si la respuesta es correcta o incorrecta. Por ejemplo:

Fracción que muestra Carta:bit	Interacción del jugador	Carta:bit corrige
$\frac{2}{4}$ (hay equivalencia con $\frac{1}{2}$)	Pantalla hacia arriba	Respuesta correcta
	Pantalla hacia abajo	Respuesta incorrecta.
$\frac{1}{3}$ (no hay equivalencia con $\frac{1}{2}$)	Pantalla hacia arriba	Respuesta incorrecta.
	Pantalla hacia abajo	Respuesta correcta.

Recordando la experiencia de jugar el juego, los estudiantes identifican que no pueden conocer cuál es la fracción que hay que evaluar, porque ya programaron anteriormente a Carta:bit para que genere la fracción al azar. Por esta razón, deben generalizar los ejemplos trabajados en el cuadro anterior, identificando que para corregir la respuesta no importa de qué fracción se trata. Lo **relevante** es determinar si es equivalente a $\frac{1}{2}$ o no y qué interacción realizó el jugador. Puede expresarse en una tabla similar a la anterior y servir como referencia. El DR irá dejando registro de las diversas anotaciones a medida que avancen en el desarrollo del proyecto. Se sugiere hacerlo en un

documento único del estilo [Anotaciones de Cartas](#).

La fracción es equivalente a $\frac{1}{2}$.	Pantalla hacia arriba	Respuesta correcta.
	Pantalla hacia abajo	Respuesta incorrecta.
La fracción no es equivalente a $\frac{1}{2}$.	Pantalla hacia arriba	Respuesta incorrecta.
	Pantalla hacia abajo	Respuesta correcta.

 **Funciones, condiciones y alternativas anidadas**
 La forma que adopte la estrategia dependerá de cómo los estudiantes elijan combinar las condiciones y el uso de las alternativas. En cualquier caso, el DR debe alentar a la definición de funciones para evitar combinaciones o anidaciones demasiado complejas.

CARTA:BIT
 Al recibir radio cadena
 Si cadena = "TERMINAR PARTIDA"
 Mostrar **respuesta**

Función Mostrar **respuesta**:
 Si numerador * 2 = denominador:
~~Mostrar cara feliz~~
Respuesta correcta con pantalla arriba
 si no
~~Mostrar cara triste~~
Respuesta correcta con pantalla abajo

Función Respuesta correcta con pantalla arriba:
 Si está pantalla hacia arriba:
 Mostrar cara feliz
 Si no:
 Mostrar cara triste

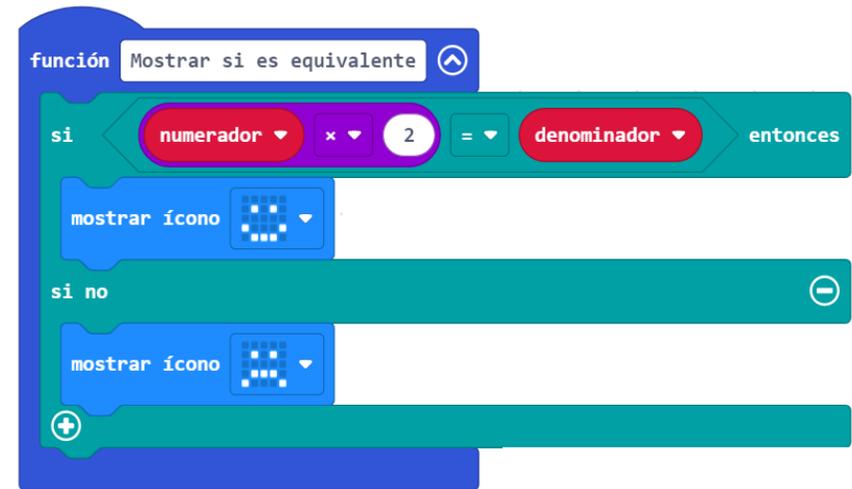
Función Respuesta correcta con pantalla abajo:
 Si está pantalla hacia abajo:
 Mostrar cara feliz
 Si no
 Mostrar cara triste

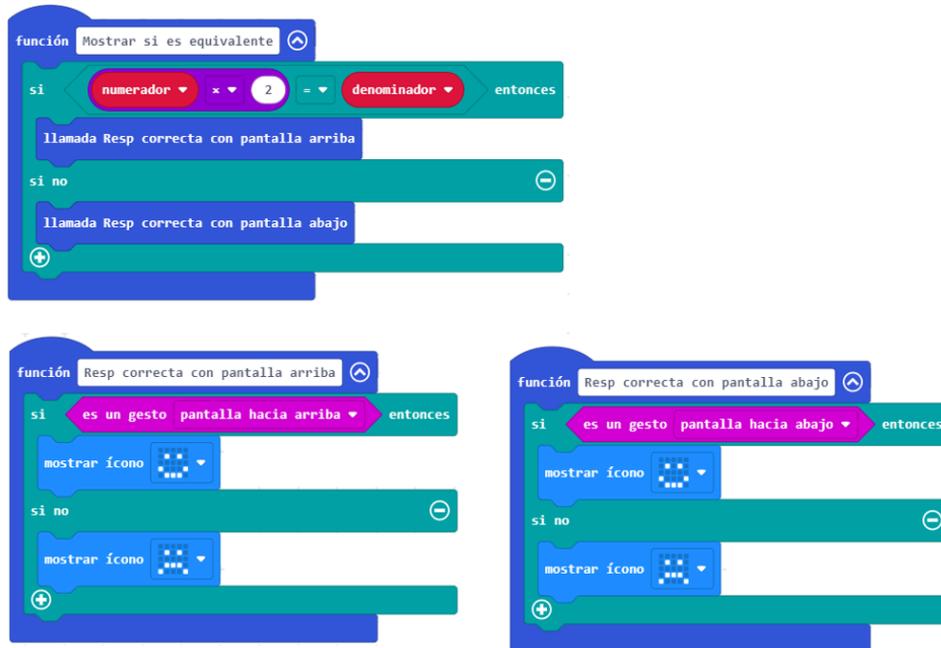
En este ejemplo de estrategia se definen funciones y se evitan alternativas anidadas

Programación

Al igual que en la etapa anterior, durante la programación se estimula a los estudiantes a que se basen en la estrategia elaborada para modificar sus proyectos.

Siguiendo con el ejemplo de la estrategia que propone esta guía, se deben realizar los siguientes cambios en el código: donde se mostraba una cara feliz (porque la fracción obtenida era equivalente a $\frac{1}{2}$) debe incorporarse una función que agrupe instrucciones relacionadas a la interacción del usuario (combinando los bloques de alternativa condicional y de entrada para detectar el movimiento de la placa). Luego, se incorporan los bloques que muestran un ícono para la corrección automática. Se debe realizar un proceso similar para la sección donde antes se mostraba una cara triste, es decir cuando la fracción no es equivalente a $\frac{1}{2}$.





3. Cierre (5 min)

Abstracción y generalización

¿De qué depende que la placa muestre una cara feliz o una triste? ¿Cómo hicieron para descubrir la regla general que incorporaron a la estrategia? ¿Qué cosas que hicieron en la VC los ayudaron a descubrir esta regla? ¿Hubieran podido programar directamente sin conocer esta regla?

En esta experiencia, los estudiantes tuvieron que armar un programa cuyo comportamiento depende de variables que son desconocidas hasta el momento

del juego. Para eso, tuvieron que expresar reglas **generales**, es decir, que funcionan para cualquiera de los valores posibles (tanto para la orientación de la placa como para los componentes de la fracción). Para descubrir esas reglas, propusieron casos puntuales y analizaron los resultados esperados para identificar regularidades. El DR agrega que este proceso de encontrar reglas generales a partir de ejemplos particulares se llama **abstracción**. Es, además, un buen ejemplo para reforzar la importancia de analizar el problema y pensar una estrategia antes de empezar a programar.

Registro en Crea

El DR publica las notas y reflexiones de los intercambios en el **Registro Común** y solicita a los estudiantes compartir en el foro de la etapa, modificaciones que hayan realizado en el Plan de desarrollo y los avances de proyecto realizados en Make.code.

Invitar a los estudiantes a resolver la [actividad interactiva](#) de la etapa en las que se repasen condiciones y alternativas anidadas.

La Yapa: Propuestas para seguir en casa

¿Te imaginas de qué otro modo puedes programar la interacción del jugador y la Carta:bit? ¿Qué sensor usarías?



<p>ETAPA 8 ↓ (opcional) Más fracciones para comparar</p>	<p>Coordinación dupla pedagógica</p>
<p>El objetivo de esta etapa es que los estudiantes piensen una modificación de la versión automática del juego: la fracción aleatoria a comparar será elegida por Super:bit</p> <p>En el aula, se motivan estrategias de cálculo para decidir la equivalencia con distintas fracciones.</p> <p>En la VC, los estudiantes planifican y programan esta nueva versión considerando la posibilidad de reutilizar, generalizar y adaptar lo que hicieron hasta ahora en el proyecto.</p> <p>Objetivos</p> <hr/> <p>Se espera que los estudiantes sean capaces de:</p> <ul style="list-style-type: none"> • Elaborar una estrategia general en base a una particular trabajada en la etapa anterior. • Valorar los procedimientos o funciones como una herramienta para modificar más fácilmente los programas. • Recuperar la noción de variable como herramienta que permite almacenar información. 	<p>Decisiones conjuntas entre DA y DR:</p> <ul style="list-style-type: none"> • En base al avance del grupo clase y la fluidez de los estudiantes, tanto en programación como en matemática, decidir si se realiza esta etapa y con qué nivel de guía. • El DR puede sugerir al DA algunas ideas para trabajar la equivalencia de fracciones que resulten fáciles de adaptar a los programas que ya hicieron los estudiantes en etapas previas. <p>Decisiones del DA</p> <ul style="list-style-type: none"> • Definir la estrategia didáctica para trabajar la equivalencia de fracciones y elaborar el algoritmo. <p>Información que necesita tener el DR:</p> <ul style="list-style-type: none"> • Cómo se abordó en el aula la equivalencia de fracciones.

AULA ↓ Cálculos pensados

Notas para el DA ↓



Propósitos mínimos

- Recuperar las estrategias de decisión de equivalencia con $\frac{1}{2}$ para generalizarlas a dos fracciones cualesquiera.

Propósitos óptimos

- Recuperar los saberes matemáticos necesarios para que los estudiantes planteen un método de decisión de equivalencia entre dos fracciones cualesquiera.

¿Cómo decidir si dos fracciones son equivalentes?

Los estudiantes necesitarán una manera de decidir si dos fracciones cualesquiera son equivalentes. Para esto, se sugiere generalizar o adaptar el algoritmo propuesto en las etapas previas para comparar contra $\frac{1}{2}$.

Pensando cálculos matemáticos

De la misma manera que cuando jugaban comparando contra $\frac{1}{2}$, las nuevas reglas del juego requerirán de los estudiantes estrategias veloces para decidir mentalmente la equivalencia de fracciones cualesquiera.

VC 8 - A ↓ Opcional

Nueva versión del juego

Desafío

Programar a Super:bit para que elija una fracción al azar y la envíe a Carta:bit. La misma será usada para compararla con la fracción al azar que ella genere.

1. Inicio (5 min)

Presentación: ¿Qué le falta a su programa para cumplir con este desafío? ¿Qué pueden aprovechar de lo ya programado? ¿Cómo van a organizar el trabajo?

Se presenta el desafío y se establecen algunas pautas de trabajo. Al igual que hicieron en las etapas anteriores, a partir de la especificación, los estudiantes deberán **modificar el Plan de desarrollo** y luego una **estrategia**, prestando especial atención a la **reutilización y adaptación** de lo que ya programaron.

Es una oportunidad para que, basándose en la experiencia de las etapas previas, transiten este recorrido con más autonomía.

2. Desarrollo (30 min)

Análisis previo:

Se puede rectificar la especificación para agregar esta característica nueva y abordar la planificación de esta etapa modificando el Plan de desarrollo, basándose en el hito anterior.

Desafío: La Super:bit genera una fracción al azar y se la envía a Carta:bit para que la use como fracción contra la cual comparar.

Es la versión completamente automatizada del juego: las fracciones de la Carta:bit, la fracción contra la cual comparar y la evaluación de la respuesta se generan automáticamente.

Se parte de: La Carta:bit juega contra una "fracción fija" (por ejemplo, $\frac{1}{2}$)

Se modifica: la fracción contra la que comparan la Carta:bit.

Se agrega: La Super:bit genera una fracción al azar y la envía a Carta:bit.

Los estudiantes deben considerar que:

- la Super:bit, además de generar la fracción, debe enviarla a Carta:bit.
- la Carta:bit, además de recibir los mensajes, debe recibir la fracción que envía la Super:bit.

- la nueva fracción cumple el mismo rol que antes cumplía $\frac{1}{2}$.

Estrategia

Los estudiantes pueden ampliar la estrategia de la etapa anterior. Deben identificar **qué partes del programa hace falta modificar** y qué fragmentos de programa o qué problemas que ya resolvieron pueden servirles.

SUPER:BIT

Al presionar A:

Almacenar fracción al azar

Enviar fracción almacenada

Enviar "Empezar"

Al presionar B:

Enviar "Terminar"

CARTA:BIT

Al recibir radio cadena

Si cadena = "Empezar"

Almacenar fracción al azar

Mostrar fracción almacenada

Si cadena = "Terminar"

Mostrar respuesta

Función Mostrar respuesta:

Si $\text{numerador} * \text{denominadorRecibido}$

= $\text{denominador} * \text{numeradorRecibido}$:

Respuesta correcta

con pantalla arriba.

si no:

Respuesta correcta

con pantalla abajo

Programación

Para programar el envío y la recepción de la fracción, los estudiantes necesitarán el bloque *enviar valor "name" = value* y *al recibir name, value*. El DR puede explicar el funcionamiento de este bloque para que los estudiantes se concentren en cómo incorporarlo en la programación de su estrategia.

Enviar fracción:

Enviar valor "numerador" =
numerador

Enviar valor "denominador" =
denominador

Al recibir *name, value*:

Si *name* = "numerador"
Almacenar *value* en *numerador recibido*

Si *name* = "denominador"
Almacenar *value* en *denominador recibido*

★ Importante

Antes del cierre considerar las recomendaciones respecto a la importancia de compartir los avances del proyecto en el **Foro de evidencias** y realizar las **Actividades interactivas**.

3. Cierre (10 min)

Modificación de programas: Importancia de las funciones para aprovechar más fácilmente, legibilidad.

Comparen la estrategia y el programa que hicieron en esta etapa con el que hicieron en la etapa 6. ¿En qué se parecen y en qué se diferencian? ¿Por qué? ¿Cuántos bloques tuvieron que cambiar o agregar? ¿Dirían que son muchos o pocos, teniendo en cuenta lo que tiene que hacer el nuevo programa? ¿Les resultó difícil pensar la estrategia? ¿Y modificar el programa?

Los estudiantes realizan esta comparación para identificar que las diferencias son pocas y están concentradas en las funciones. Como ya observaron en reflexiones anteriores, el hecho de tener funciones definidas permite que las modificaciones sean más fáciles de hacer, tanto porque se concentran en pocos lugares como porque es fácil identificar qué hay que cambiar gracias a los nombres de las funciones y la división en problemas pequeños.

Por otro lado, también es momento de valorar el esfuerzo de generalización realizado para modificar la estrategia (se pueden retomar nociones de generalización y abstracción trabajadas en otras etapas) y cómo este análisis previo también facilitó la programación.

Registro en Crea

El DR publica las notas y reflexiones de los intercambios en el **Registro Común** y solicita a los estudiantes compartir en el foro de la etapa, modificaciones que hayan realizado en el Plan de desarrollo y los avances de proyecto realizados en Make.code.



Invitar a los estudiantes a resolver la [actividad interactiva](#) de la etapa en las que se repase la posibilidad de reutilizar, generalizar y adaptar partes de un programa.

La Yapa: Propuestas para seguir en casa

¿Qué otros juegos puedes imaginar con las placas micro:bit? ¿Algún otro juego de cartas? ¿Cómo podría ser un juego usando otros sensores? Y en lugar de fracciones, ¿con qué otros datos podrías jugar?



VC 8-B ↓ Opcional Puntaje

Desafío

En cada partida cuando el jugador acierta, suma 10 puntos; de lo contrario, resta 5 puntos.

1. Inicio (15 min)

¿Qué debemos incorporar para que nuestro juego tenga puntaje? ¿En el Plan de desarrollo contemplamos esta opción? ¿Qué herramienta de programación se necesita para poder hacerlo?

Se retoma lo trabajado en propuestas anteriores sobre el uso de variables. En el Plan de desarrollo se describen las acciones asociadas para programar el puntaje en el juego.

El DR que considere necesario reforzar el concepto de variable puede llevar a cabo el juego realizado en la Etapa 2 en la clase de aula, pero ahora suma puntos el jugador que acierte en una partida. Luego de jugar unas partidas en grupo, el DR indaga de qué manera llevaron el control de los puntos. Pueden surgir distintas formas, por ejemplo: anotar en la pizarra, anotar en papel, sumar en voz alta, etc. Todas estas herramientas sirven para asociar con la herramienta de programación Variable y su función de recordar un valor.

2. Desarrollo (20 min)

Parte 1: Cada vez que el estudiante acierta la fracción equivalente suma 10 puntos en el juego, en caso contrario resta 5 puntos.

*Observen el programa, ¿en qué momento se controla si hay un acierto? ¿Cómo sabemos hasta ahora que se acertó? ¿Cómo sabemos hasta ahora que **NO** se acertó?*

El DR brinda el tiempo a los estudiantes para que recuerden y exploren la categoría **Variables** y reconozcan que allí se encuentran los bloques que permiten crear, modificar y conocer el valor de una variable. El estudiante debe incorporar una variable **Puntaje** y la programan para que sume 10 puntos en cada acierto y resta 5 en el caso contrario.



Parte 2: Iniciar el valor del puntaje en cero cuando se arranca el juego y mostrar el puntaje al finalizar cada partida.

Observen el programa, ¿en qué momento la variable Puntaje se debe iniciar en cero? ¿Cómo vemos los puntos acumulados cuando finaliza una partida?

Los estudiantes resuelven esta segunda parte de manera autónoma. El DR puede realizar preguntas para que detecten que la variable tiene que iniciar en cero cuando se comienza el juego y que deben utilizar uno de los bloques de la categoría Básico para mostrar el valor cuando la Carta:bit recibe Finalizar de parte de la Super:bit.

Importante

Antes del cierre considerar las recomendaciones respecto a la importancia de compartir los avances del proyecto en el **Foro de evidencias** y realizar las **Actividades interactivas**.

3. Cierre (10 min)

Los programas recuerdan información durante su ejecución. ¿Qué programas o plataformas conocen donde se almacena información y qué se almacena? Si es necesario el DR ofrece algunos ejemplos:

- Otros juegos: cantidad de vidas disponibles, número de nivel alcanzado, nombre del jugador, equipo del jugador, datos de configuración

- Editores de texto: se recuerda el texto escrito hasta el momento y si estamos usando algún formato particular, como negrita o color.
- Aplicaciones de dibujo: recuerda el tipo de lápiz seleccionado para dibujar.

Registro en Crea

El DR publica el registro con las notas y reflexiones de los intercambios en el **Registro Común**. Pueden incorporarse capturas de pantalla de las actividades de los estudiantes.

Invitar a los estudiantes a resolver la [actividad interactiva](#) de la etapa.



La Yapa: Propuestas para seguir en casa

¡Felicidades! Lograste tener una nueva versión de tu juego. Ahora, puedes incorporar *un sonido para cuando se pierden o aciertan puntos, ¿Qué componente necesitas? ¿En qué parte incorporas la programación de este nuevo componente?*



<p>ETAPA 9↓ Cierre</p>	<p>Coordinación dupla pedagógica</p>
<p>En esta última etapa, se planifica la difusión del proyecto en la comunidad.</p> <p>En la VC, se realizan los últimos ajustes y la metacognición del trayecto de PC.</p> <p>Se realiza en Crea la actividad de evaluación final del proyecto.</p> <p>Objetivos</p> <hr/> <p>Se espera que los estudiantes sean capaces de:</p> <ul style="list-style-type: none"> ● Reconocer y reflexionar sobre su propio proceso de aprendizaje durante todo el proyecto. ● Compartir sus producciones con sus compañeros, familias y la comunidad educativa. 	<p>Decisiones conjuntas entre DA y DR:</p> <ul style="list-style-type: none"> ● 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 esta actividad es evocar los aspectos de la experiencia que se vuelven relevantes con las sucesivas preguntas del DR y que tal vez los estudiantes no identifican o valoran a priori. También, puede transmitirle al DR situaciones que haya notado como particularmente significativas para ellos, de modo tal que el DR las considere al momento de hacer la reflexión final. <p>Decisiones del DA</p> <ul style="list-style-type: none"> ● De qué manera compartirán las aplicaciones con las familias y comunidad educativa.

AULA ↓

Cierre de proyecto de aula

Notas para el DA ↓



Propósitos mínimos

- Propiciar una instancia de cierre del proyecto de aula

Propósitos óptimos

- Planificar junto a los estudiantes la forma en que compartirán el juego creado con la comunidad educativa.

Planificar el día o la forma en que se mostrará el juego programado, invitando a los compañeros, las familias, personal de la escuela y otros estudiantes y maestros, a jugarlo.

Esta etapa de cierre, es una oportunidad para reconocer, evaluar y dialogar sobre los aprendizajes curriculares que les dejó el proyecto.

VC ↓

Producto final y metacognición

1. Inicio (5 min)

El DR invita a los estudiantes a que recuperen los programas de la etapa anterior y que cuenten brevemente en qué momento del proyecto se encuentran y qué creen que les falta a su programa.

2. Desarrollo (15 min)

Preparación del programa final

Los estudiantes ajustan cuestiones que hayan quedado pendientes y preparan los programas en su versión final. Si programaron la Super :bit y la Carta:bit en el mismo archivo, es momento de recordar que es fundamental separar y borrar lo que no corresponde, para generar un archivo para cada una según el rol asignado..

3. Cierre (25 min)

Reflexión final y metacognición

El desarrollo incremental, reutilización y automatización creciente

¿Podrían haber programado de una vez, la versión final? ¿Les resultó de utilidad escribir el Plan de desarrollo? ¿Por qué creen que se escribe antes de comenzar a programar?

¿Qué herramientas del lenguaje les facilitaron la tarea de ir haciendo crecer el programa? ¿Les parece importante poder modificar los programas fácilmente?

¿Fue necesario contar con una versión completamente automática del juego? Durante todo el proyecto, ¿pudieron suplantar las características que faltaban programar?

Funciones y Estrategia

¿Qué les pareció más difícil? ¿La parte de la programación o pensar la estrategia?

¿En qué momentos les ayudó tener la estrategia definida o tener una estrategia anterior contra la que comparar? ¿Por qué?

¿Para qué les sirvió ir definiendo funciones? ¿Se imaginan cómo serían las versiones del programa sin las funciones? ¿Es estrictamente necesario definir funciones? ¿Valió la pena hacerlo?

Funciones y modificabilidad

¿En qué momento les pareció que haber definido las funciones les estaba simplificando las tareas de modificación? ¿Por qué? ¿Qué hubiera pasado si no las definían? ¿Les parece que es frecuente o infrecuente que los programas se modifiquen? ¿Será, entonces, una buena o mala idea definir funciones frecuentemente?

Proceso de aprendizaje

¿Qué cosas de las que hicimos no conocían? (qué aprendimos) ¿Cómo le contarían 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

¿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 reflejan 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?

3. Cierre (10 min)

Evaluación

El DR invita a los estudiantes a resolver la evaluación del proyecto.

Registro en Crea

Los estudiantes comparten sus proyectos finales en CREA.



ANEXO 1

Mazo de cartas con fracciones

Juego en formato papel

Objetivo: Cada jugador o equipo debe dar vuelta su carta sólo si la fracción que le tocó es equivalente a la que muestra el DA. Si no lo es, queda boca abajo.

Recursos: [Mazo de cartas con fracciones](#) para imprimir y recortar.

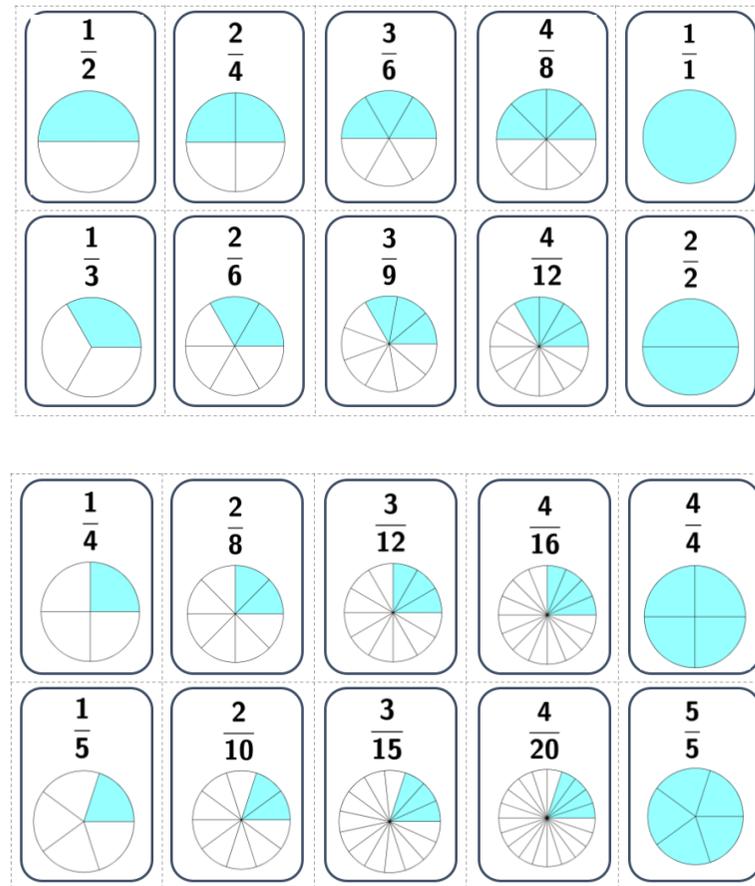
Participantes: De manera individual, equipos de juego (entre 3 y 5 integrantes) o parejas. El DA (o quien él designe) será quien reparta las cartas con fracciones y elija qué carta con fracción mostrar a los jugadores.

Dinámica: Comienza cuando el DA o un estudiante reparte una carta con fracciones a cada estudiante o equipo. Luego, elige al azar una carta con fracción del mismo mazo, la muestra a todos al mismo tiempo e indica que deben dejarla boca arriba si la carta es una **fracción equivalente** a la que el DA está mostrando o boca abajo si no lo es.

Ganador: El jugador o equipo que lo resuelve correctamente en el menor tiempo o quien sume más puntos (en caso de jugar varias partidas) es el ganador del juego.

[Volver a Etapa 2](#)

Mazo de cartas con fracciones



 ANEXO 2

Plan de desarrollo de ejemplo

Plan de desarrollo de ejemplo:

1. Presionar el botón A en Super:bit para que envíe INICIAR PARTIDA
2. Programar a Carta:bit para que muestre una fracción al azar cuando recibe la señal de INICIAR PARTIDA
3. Programar a Carta:bit para ingresar una respuesta con el movimiento de la placa
4. Hacer que Carta:bit muestre si la respuesta del jugador fue correcta con una carita feliz
5. Programar el puntaje

[Volver a Etapa 3](#)


ANEXO 3
Canciones con estrofas repetidas - Ejemplo de actividad de aula
[Gorrión - Tabaré Cardozo](#)

Soy el novio de la luna
 Esta noche bacanal
 Brilla el cielo constelado
 Vivo enamorado para enamorar

[ESTRIBILLO]

Tengo estrellas en las manos
 Y en tus ojos claridad
 Soy el hijo de los vientos
 Que abre el firmamento cada carnaval

[ESTRIBILLO]

Traigo pájaros azules
 Y una flor en el ojal
 Robo besos de princesas
 Tengo una promesa para regresar

[ESTRIBILLO]

ESTRIBILLO:
 Soy un gorrión
 Soy un bufón
 Que cantando alegre va
 De polizón
 En un camión
 Que atraviesa la ciudad

[Vayan pelando las chauchas - Canario Luna](#)

(CHAUCHAS:)
 Vayan pelando las chauchas
 Vayan pelando las chauchas
 Aunque les cueste trabajo

(LA CELESTE:)
 Donde juega la celeste
 Donde juega la celeste
 Todo el mundo boca abajo

[CHAUCHAS]
 [LA CELESTE]

(URUGUAYOS:)
 Uruguayos, sangre de campeones
 Uruguayos, garra y calidad
 Uruguayos, sangre de campeones
 Uruguayos, garra y calidad

[CHAUCHAS]
 [LA CELESTE]
 [CHAUCHAS]
 [LA CELESTE]
 [URUGUAYOS]
 [URUGUAYOS]

[Río de la Plata - Natalia Oreiro](#)

Con lágrimas en los ojos me fui alejando.
 Vi lo duro que fue dejar todo atrás, mi gente y mi barrio.
 Tener que marcharme sola de mi tierra un día cuando era una niña dejándolo todo por mis fantasías.

Soy del Río de la Plata, corazón latino, soy bien candombera.
 Llevo siempre una sonrisa con mi sueño a cuestas no tengo fronteras.
 Soy del Río de la Plata, que viva el candombe de sangre caliente, ritmo que me enciende el alma, que brilla en los ojos de toda mi gente.

→ ESTRIBILLO

Las lágrimas de mis ojos se fueron yendo.
 Volé por tantos caminos que sin darme cuenta casi toco el cielo.
 Hoy yo le canto al mundo con toda la fuerza que llevo aquí dentro:
 que vale la pena luchar por un sueño

[ESTRIBILLO]

Cumplir un sueño te puede cambiar la vida.
 Crecí buscando mi suerte, soñando despierta y deshojando heridas.
 Hoy sé que aunque fui tan lejos buscando un destino, soy siempre la misma que llevo en el alma mi tierra querida.

[ESTRIBILLO]

Por ejemplo, analizando esta canción:

Gorrión - Tabaré Cardozo

Soy el novio de la luna
Esta noche bacanal
Brilla el cielo constelado
Vivo enamorado para enamorar

Soy un gorrión
Soy un bufón
Que cantando alegre va
De polizón
En un camión
Que atraviesa la ciudad

Tengo estrellas en las manos
Y en tus ojos claridad
Soy el hijo de los vientos
Que abre el firmamento cada carnaval

Soy un gorrión
Soy un bufón
Que cantando alegre va
De polizón
En un camión
Que atraviesa la ciudad

Traigo pájaros azules
Y una flor en el ojal
Robo besos de princesas
Tengo una promesa para regresar

Soy un gorrión
Soy un bufón
Que cantando alegre va
De polizón
En un camión
Que atraviesa la ciudad

La letra de esta canción se podría esquematizar así:

[ESTROFA 1]
[ESTRIBILLO]
[ESTROFA 2]
[ESTRIBILLO]
[ESTROFA 3]
[ESTRIBILLO]

¿Y si variamos la canción?

[ESTRIBILLO]
[ESTRIBILLO]
[ESTROFA 3]
[ESTROFA 2]
[ESTROFA 1]
[ESTRIBILLO]

[Vover a Etapa 4](#)

ANEXO 4

Actividad en Pilas Bloques



[Pilas Bloques-Campeón desordenado](#)

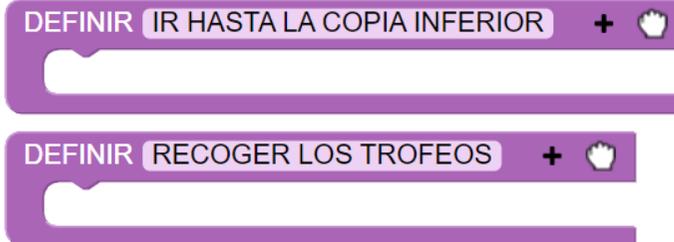
Un ejemplo del desafío resuelto

- **Pensar estrategia de solución:**
 1. Ir hasta la copa inferior
 2. Recoger los trofeos
- **Se representa la estrategia propuesta con procedimientos:**

AL EMPEZAR A EJECUTAR

IR HASTA LA COPIA INFERIOR

RECOGER LOS TROFEOS



Primitivas

Por último se incorporan las primitivas necesarias a cada procedimiento para cumplir con la acción que designa el nombre.

Este desafío admite otras estrategias de solución, por lo cual si los estudiantes las proponen, se recomienda mostrarlas, y analizarlas.

[Volver a Etapa 4](#)