



SECUENCIA DIDÁCTICA POSTPRIMARIA RURAL



I. INFORMACIÓN GENERAL			
NOMBRE DOCENTE: DANIEL FELIPE GUZMAN	SEDE: VALLEDUPAR	GRADO: 6°-9°	PERODO: 4°
ASIGNATURA: TECNOLOGÍA E INFORMÁTICA	FECHA INICIO: 25 SEPTIEMBRE	F. FINALIZACIÓN: 06 OCTUBRE	

COMPONENTE PEDAGOGICO

Temática	<i>Programación B Condicionales y Repeticiones</i>		
Competencia	Selecciona correctamente bloques, los combina en el área de programación siendo capaz de reconocer patrones para elaborar instrucciones que permitan resolver repetidamente un mismo problema y que tomen decisiones cuyo comportamiento no sea siempre el mismo.	DBA	6°-7° Presento diversas alternativas para la satisfacción de necesidades y solución de problemas tecnológicos e informáticos en diferentes contextos. 8°-9° Soluciono problemas tecnológicos e informáticos dando cumplimiento a restricciones, condiciones y especificaciones técnicas y contextuales. 10°-11° Propongo innovaciones tecnológicas e informáticas para la solución de problemas dando cumplimiento a restricciones, condiciones y especificaciones técnicas y contextuales.
Evidencias de aprendizajes y desempeños esperados	6°-7° Descompongo un problema en secuencia de pasos proponiendo o desarrollando probables soluciones a los problemas planteados. 8°-9° Reconozco que no hay soluciones perfectas, y que pueden existir varias soluciones a un mismo problema según los criterios utilizados y su ponderación. 10°-11° Propongo, analizo y comparo diferentes soluciones a un mismo problema de la tecnología o la informática, explicando su origen, ventajas y dificultades.	Justificación	El Área de Tecnología e Informática (T&I) favorece el desarrollo de competencias tecnológicas de niños y jóvenes de Educación Básica y Media para, principalmente, estudiar, reflexionar y comprender la naturaleza, evolución e implicaciones ético-políticas de la tecnología y la informática en la vida cotidiana; así como, para resolver problemas, necesidades y deseos de orden tecnológico asociados a la mejora de la calidad de vida de las personas y demás especies que habitan el planeta, procurando la conservación de un mundo sustentable y sostenible para las generaciones actuales y futuras.
Elementos didácticos	<i>Tablero, Proyector, PC, Internet, impresora,</i>		

ESTRUCTURA

ACTIVIDADES INICIALES

Exploración:	<p>Clase 1: Como ya conocemos el entorno de programación y el manejo de los bloques, vamos a aprender lo que se debe hacer cuando los algoritmos son muy largos y requieren usar los mismos bloques, con las mismas instrucciones muchas veces.</p> <p>Analicemos este caso: Gato debe saltar hacia arriba 100 veces seguidas. Piensa ¿cómo quedaría el algoritmo construido usando los bloques? ¿No sería un algoritmo muy largo y con muchos bloques?</p>	 
---------------------	--	--

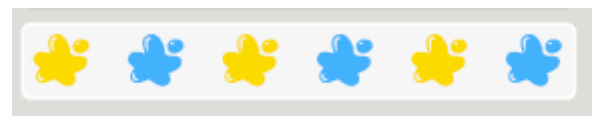
Clase 1:

Conozcamos los patrones, patrones son secuencias que se repiten cada cierto tiempo, un ejemplo de patrón son los meses del año o los días de la semana, se repiten cada semana o cada mes.

Analicemos la siguiente secuencia de imágenes y pensemos ¿cuáles características u objetos se repiten y cada cuanto ocurre la repetición?



Analizando la secuencia anterior se identifican dos patrones, el primer de forma y el segundo de forma,



El primer patrón es de figura inicia con una estrella y finaliza con la luna, este patrón se repite siempre después de la luna 3 veces; el segundo patrón es el color primero va el amarillo y luego el azul.

Miremos los siguientes patrones, busquemos y analicemos las secuencias que se repiten y cada cuanto ocurre la repetición, puede ser de forma, color y ocurrir cada dos o tres figuras.

➤ **AB:** se repiten dos elementos alternadamente.



➤ **ABC:** se repiten tres elementos alternadamente.



➤ **AAB:** se repite dos veces un mismo elemento y a continuación otro.



➤ **AABB:** se repite dos veces un elemento y a continuación dos veces otro.



Clase 2:

Estructuración /
Transferencia

E

Engineering
Ingeniería


REPETICIONES

Cuando estamos codificando algoritmos y según la situación que estamos trabajando, a menudo tenemos que llevar a cabo las mismas instrucciones más de una vez (así como el ejemplo del gato que debía saltar 100 veces que vimos al inicio de la unidad). Para evitar tener que poner los mismos bloques muchas veces, podemos incluirlos en un bloque que tiene una instrucción de bucle que repite los bloques las veces que necesitamos. ¡Ya conoces los bloques Por siempre y Repetir del tipo bucle!

Bucles o Ciclos

Nos permiten repetir una serie de acciones:

- x's cantidad de veces
- hasta que ocurra algo
- mientras está sucediendo algo.



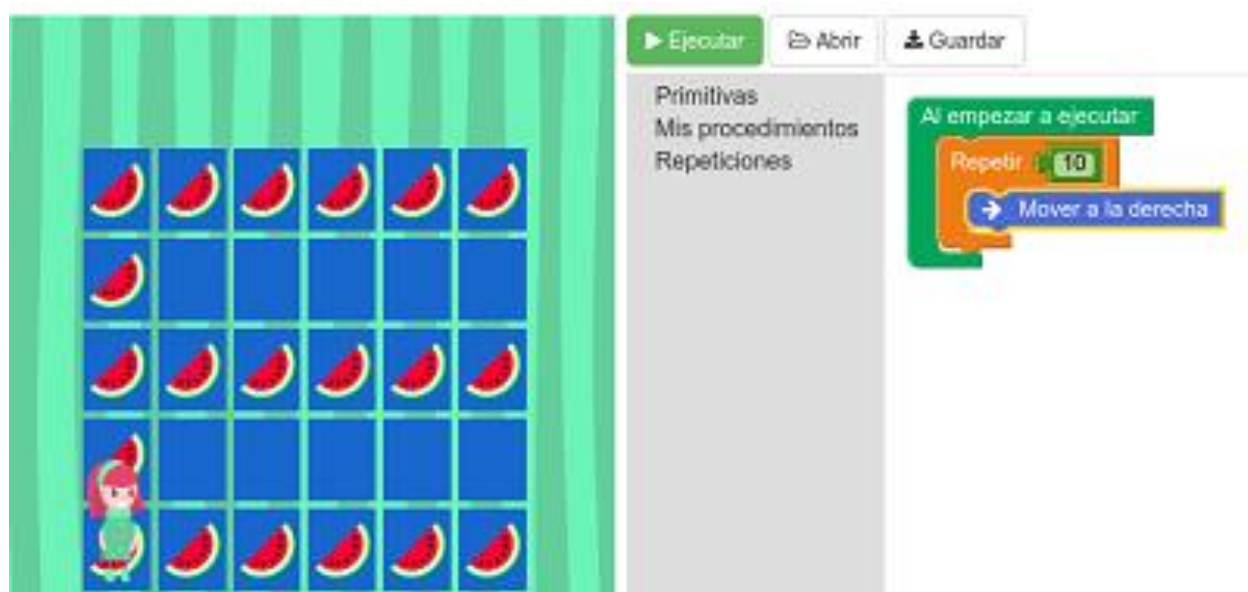
>playkodo



Para el caso de Pilas Bloques, el bloque para repetir varias veces una instrucción es representado con un bloque naranja llamado repetir, en el que se ingresa el número de veces que se repite la serie de bloques que encajen dentro de este bloque.

Miremos con atención el uso del bloque de repetición, para no insertar los mismos bloques muchas veces, se ingresa primero el número de veces que se debe realizar la acción o acciones, luego dentro del bloque se ingresan los bloques con las instrucciones o algoritmos que necesitamos repetir.

Procuremos completar con dos bloques que faltan el algoritmo de la situación problema usando el bucle repetir.



Evaluación Formativa Verificar que el concepto de bucle y patrón sea claro; al igual que el análisis de patrones.

ACTIVIDADES DE DESARROLLO

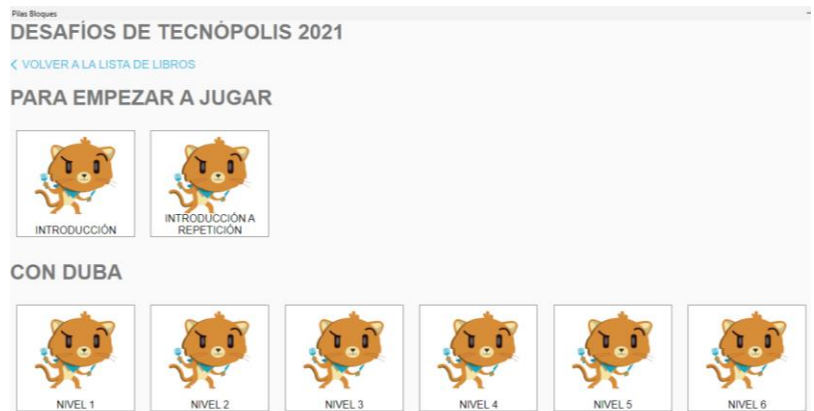
Práctica / Ejecución

Clase 1:

Analicemos este patrón, tratemos de establecer la secuencia de formas, colores y el número de veces que se repite y cada cuánto se repite; selecciona la secuencia de color y formas que corresponde y que están ubicadas en la parte inferior del patrón.

Clase 2:

Continuando con el proceso de práctica de programación por bloques realizar todos los desafíos que incluyen a los personajes del programa en la sección Tecnópolis dentro de la aplicación Pilas Bloques.



Recuerda que los bloques se arrastran y se encajan del área de instrucciones al espacio de trabajo.

Evaluación Formativa Verificar que se apropia del proceso de análisis de los patrones por color, forma y finalización de la secuencia. Al igual que el manejo del bloque repetir en la construcción de algoritmos dentro del entorno de programación.

ACTIVIDADES DE CIERRE

TAREA 1:

Hagamos el análisis de patrones como lo hemos hecho en clases anteriores, primero completa el patrón de las formas, recuerda que debes pensar antes en:

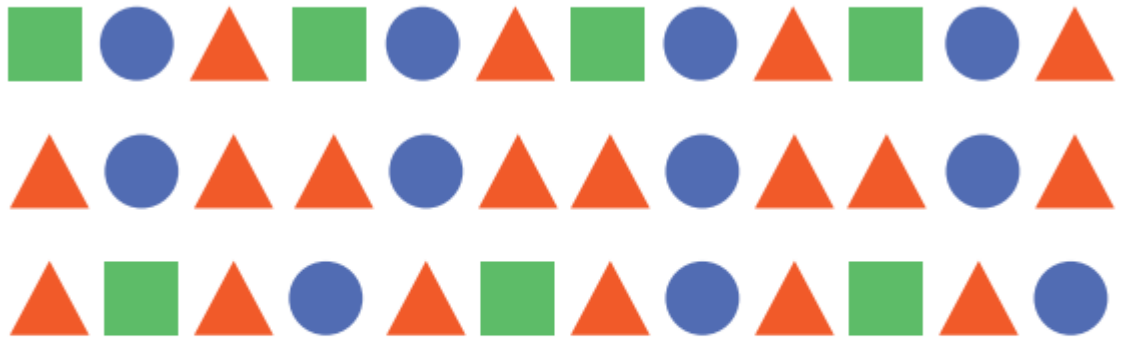
- ¿Cuál o cuáles formas son las que se repiten?
- ¿Cuántas veces se repiten las figuras para completar el patrón?
-

1. ¿AYUDAS A COTY A COMPLETAR LOS MURALES? DIBUJA LOS ELEMENTOS QUE FALTAN EN CADA UNA DE LAS TRES SECUENCIAS DE FIGURAS.



Evaluación

2. ENCERRÁ CON UN RECUADRO EL PATRÓN QUE SE REPITE EN CADA UNA DE LAS SIGUIENTES SECUENCIAS.



¿CUÁNTAS VECES SE REPITE CADA PATRÓN?

EN LA FILA 1 SE REPITE VECES

EN LA FILA 2 SE REPITE VECES

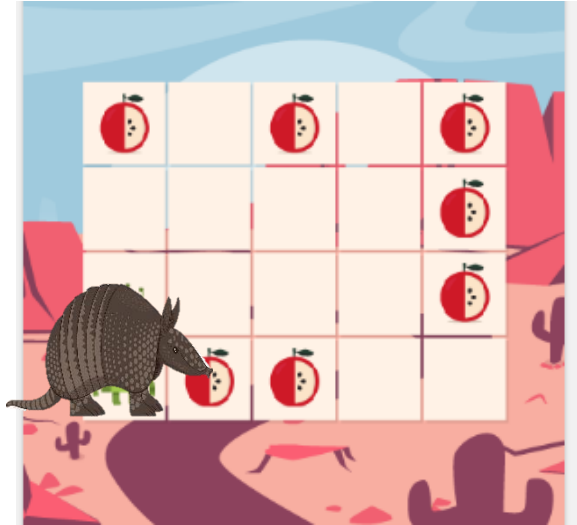
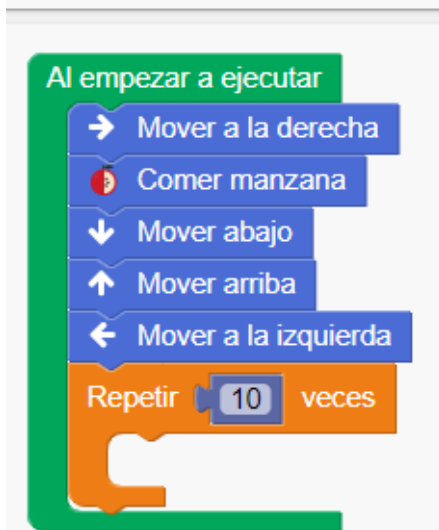
EN LA FILA 3 SE REPITE VECES

TAREA 2:

Como ya sabemos utilizar la interfaz de programación, los bloques y el bucle de repetición, vamos a analizar y reparar algunos algoritmos que tienen uno que otro error, esta actividad se debe desarrollar en grupos y en el cuaderno de tecnología.

1. Analizar la situación problema representada en la cuadrícula, luego dibuja el algoritmo que solucione la situación con los bloques vistos y disponibles, recordar siempre usar el bucle repetir si es viable para la construcción del algoritmo.

2. Revisa el algoritmo, encuentra los errores y dibújalo con las correcciones.



2. Construye el algoritmo con los bloques vistos para que la luciérnaga llegue a todos los bombillos, recuerda usar repetición.



Evaluación Formativa



	Objetivo de aprendizaje: Construye y corrige patrones y algoritmos usando repeticiones de forma efectiva y pertinente para optimizar recursos de programación.			
	CRITERIO	Valoración		
	DESEMPEÑO	ALTO	MEDIO	BAJO
	Interactúa adecuadamente con la interfaz del software de programación.	Usa adecuadamente las funciones y herramientas en la interfaz de programación. 1PT	Se le dificulta interactuar con la interfaz y las herramientas de software de programación. 1PT	No presenta trabajo. 0.5 PT
	Organizar secuencias de algoritmos ajustadas a un fin específico.	Realiza correcciones y adecuaciones a algoritmos usando la función repetición. 3PT	Codifica algoritmos con poco nivel de optimización sin usar correctamente repetir. 1.PT	No presenta trabajo. 0.5 PT
Trabajo en equipo y relaciones interpersonales.	Se comunica asertivamente con sus compañeros y trabaja en equipo con facilidad. 1PT	Se le dificulta comunicarse con asertividad al trabajar en equipo. 1PT	No presenta trabajo. 0.5 PT	