



LISTAS

Las listas son una sucesión de cero o más elementos. Esta es una definición muy simple y que no aclara demasiado en términos informáticos.

Hay varios tipos de listas, las hay enlazadas, no enlazadas, ordenadas y no ordenadas. Nosotros vamos a estudiar las listas enlazadas, tanto ordenadas como no ordenadas.

Una lista es una sucesión de nodos en la que a partir de un nodo se puede acceder al que ocupa la siguiente posición en la lista. Esta característica nos indica que el acceso a las listas es secuencial y no indexado, por lo que para acceder al último elemento de la lista hay que recorrer los $n-1$ elementos previos (n es el tamaño de la lista).

Utilización de las listas

Las listas son unas estructuras de datos muy útiles para los casos en los que se necesita almacenar información de la que no se conoce su tamaño con antelación.

También son valiosas para las situaciones en las que el volumen de datos se puede incrementar o decrementar dinámicamente durante la ejecución del programa.

Aplicación de Listas

Las operaciones que se presentan en esta aplicación son las siguientes:

Insertar: Para insertar un elemento hay que introducir, en el campo de texto del diálogo lanzado al pulsar el botón Insertar, el valor deseado. Este valor puede ser un número entero o una cadena de caracteres.



Borrar: Si se desea borrar un nodo se debe seleccionar dicho nodo y pulsar el botón Borrar.

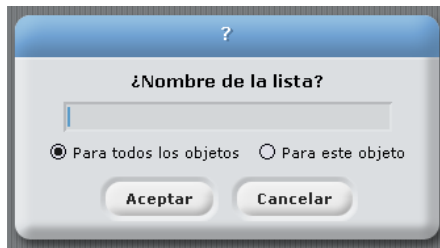
Vaciar lista: Esta acción elimina todos los elementos presentes en la lista.



Listas en Scratch

Crear: Para crear una nueva lista presionamos el siguiente botón, esta ubicado en el bloque de Variables.

Colocar un nombre representativo de los valores que la lista almacenara.



Añadir un elemento a la lista:

Al crear una lista se puede observar en el escenario el visor en el que haciendo clic sobre el + se crean espacios para ingresar valores en la lista, luego de escribir el valor debemos hacer doble clic o presionar enter.



Se puede utilizar el bloque añade, el cual nos permite agregar cualquier valor a la lista.



Borrar un elemento de la lista:

Así mismo como en una lista podemos agregar cualquier elemento, se puede eliminar cualquiera de esta.

Para ello seleccionamos la opción borrar del menú listas.



Ejercicio 1

Crear una Lista: Animales

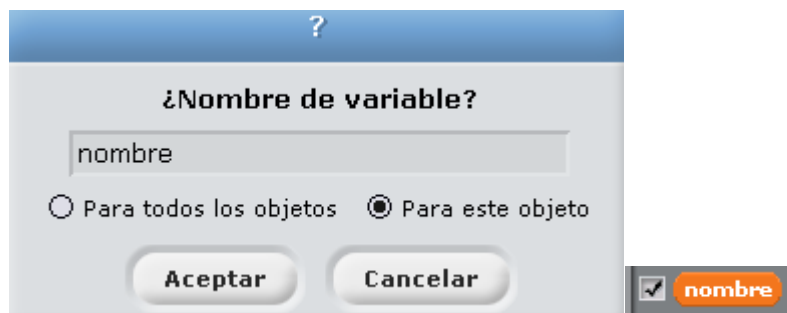
Crear un objeto caja que representa la Lista por medio de un editor y Ponerle el nombre: Caja1



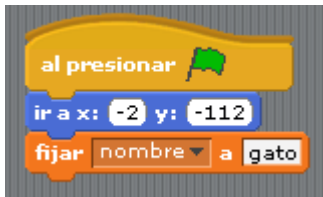
Crear elementos: Crear un objeto que represente un animal y poner en su nombre, un número.



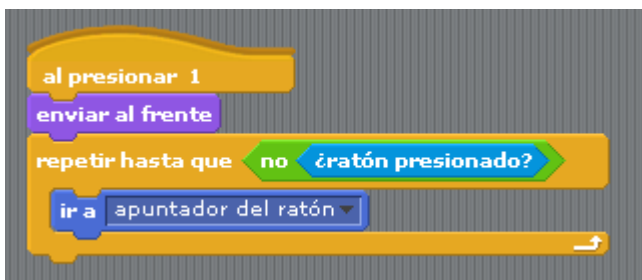
Crear variable "nombre" solo para este objeto.



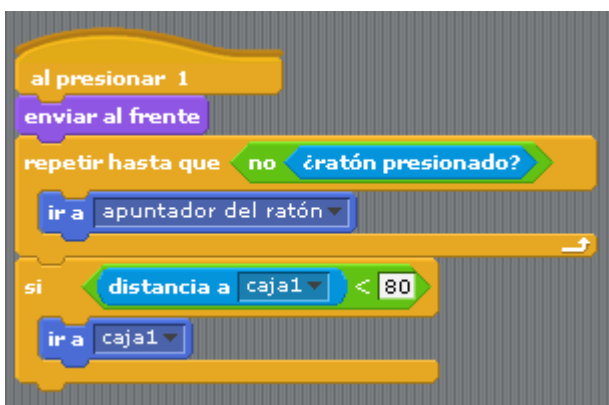
Ubicar el objeto en el centro y ponerle en la variable nombre al objeto para poderlo identificar después.



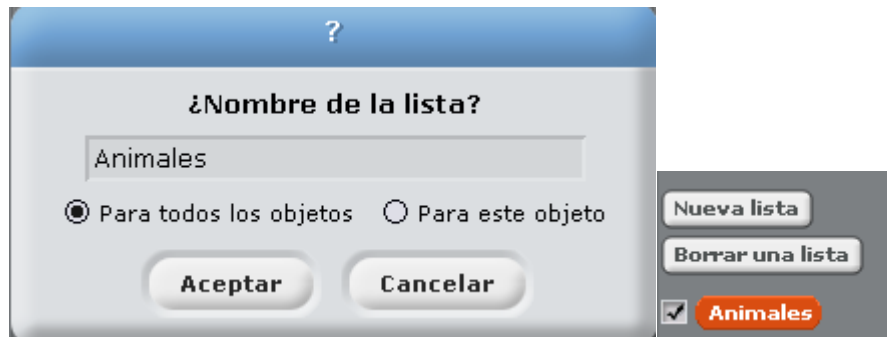
Bloque para poder arrastrar el objeto con el ratón.



Interacción de la colección con el elemento: Magnetizar: Queremos que cuando se arrastre y se suelte el objeto, si la distancia al centro de la caja (centro del disfraz) es menor a 80, vaya hacia la caja.



Validar si el elemento pertenece a la Lista: Para validar necesitamos una lista con los nombres de todos los objetos que pertenecen a la lista animales.



Inicializar la lista y mostrarla

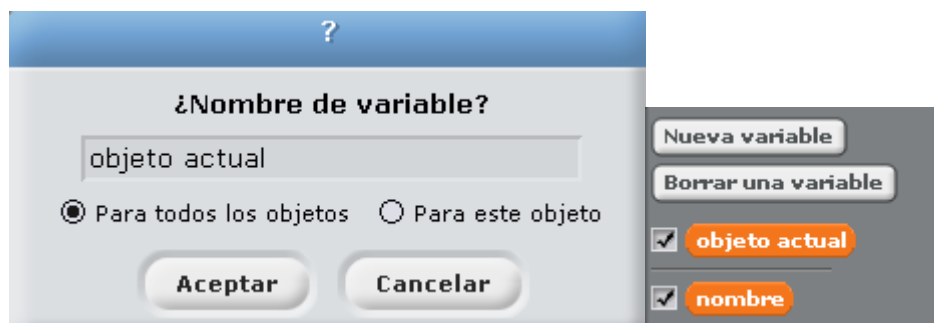
<EN CAJA1>



Validar si existe en la lista o no

¿Cómo sabemos cuál es el animal que se arrastró a la caja?

Crear una variable para todos los objetos llamada: objeto actual



<EN 1> (objeto gato)



```
al presionar 1
  enviar al frente
  repetir hasta que no ¿ratón presionado?
    ir a apuntador del ratón
  si distancia a CAJA1 < 80
    ir a CAJA1
    fijar objeto actual a NOMBRE
    enviar a todos validar caja1
```

<EN CAJA1>

```
al recibir validar caja1
  si ANIMALES contiene objeto actual
    decir Esta en la coleccion por 2 segundos
  si no
    decir No está en la coleccion por 2 segundos
```

Probamos la validación

1 nombre gato

objeto actual gato

está en la colección

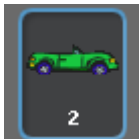




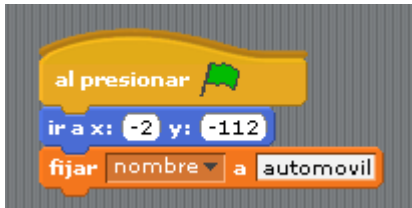
Ejercicio 2

Creamos otro objeto, que no pertenezca a la lista y validamos.

Duplicamos el objeto <1> y le cambiamos el nombre y el disfraz



Como está duplicado, queda con todos los bloques y le cambiamos la variable nombre.



Probamos y el programa debe buscar la palabra automóvil en la lista.

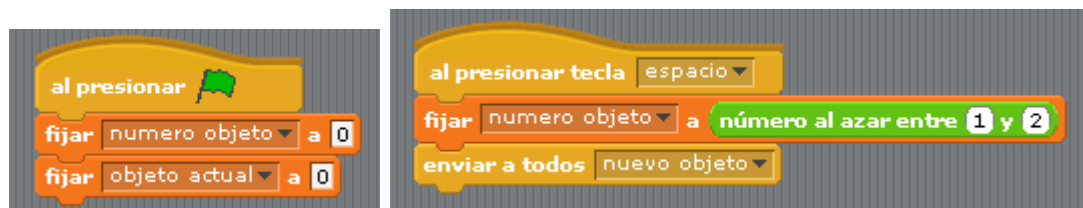




Hacer que los objetos aparezcan de a uno y de forma aleatoria. Creamos la generación de números aleatorios (en este caso de 1 a 2)

Crear variable para todos numero objeto

<EN ESCENARIO>



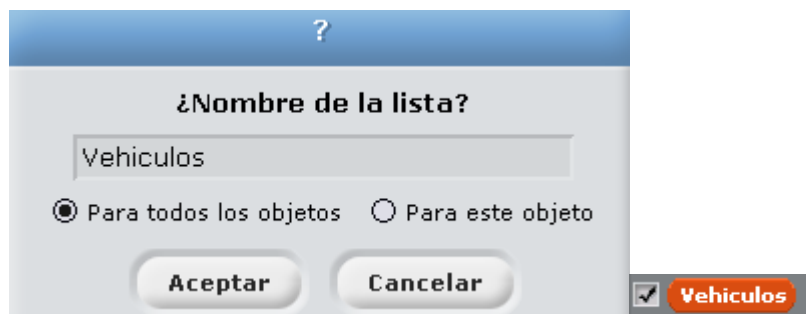
Hacemos que cada objeto identifique si debe aparecer o no y que lo ubique siempre en un punto.

<EN 1> y <EN 2>



Probamos con la tecla espacio

Creamos una nueva Lista: Vehículos.





Duplicamos el objeto caja1, cambiamos el nombre y disfraz



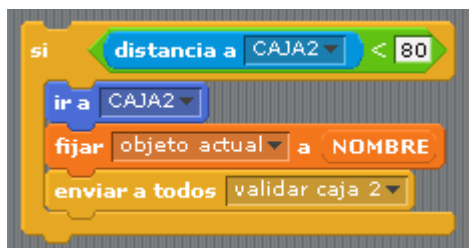
Modificamos bloques del objeto para asociarlo a la lista automóvil.



Inicializamos con bandera verde, y probamos con la tecla espacio.

Si arrastramos un objeto a la caja 2, no magnetiza. ¿Por qué?

Corrección: hay que incluir esta parte del código en cada uno de los objetos. <1, 2 y 3>, pero con su correspondiente mensaje “Validar caja2”.



Crear colecciones adicionales (OPCIONAL)

- Crear una nueva lista
- Duplicar caja y modificarla
- Modificar objetos para cuando tocan caja3, caja4, etc

MARTIN JARAMILLO PINO

Proyecto Final

Programación, creatividad y solución de problemas con Scratch



Creamos nuevos objetos: perro, caballo, helicóptero

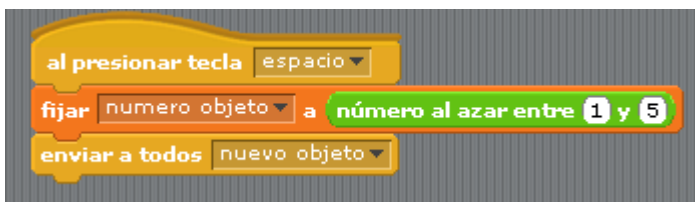
NOTA: antes de crear nuevos objetos, es mejor definir todas las listas que se van a tener.

Se duplican y se les modifica el disfraz



Se modifica rango de aleatorios

<EN ESCENARIO>



Se modifican los bloques

<En 3>





<En 4>

The code for 'En 4' consists of two event-driven blocks. The first block, 'al presionar' (when green flag clicked), contains: 'esconder' (hide), 'ir a x: -2 y: -112' (go to x: -2, y: -112), and 'fijar nombre a helicoptero' (set name to helicopter). The second block, 'al recibir nuevo objeto' (when new object received), contains: 'ir a x: -2 y: -112' (go to x: -2, y: -112), 'esconder' (hide), and a conditional 'si numero objeto = 4' (if object number = 4) followed by 'mostrar' (show).

<En 5>

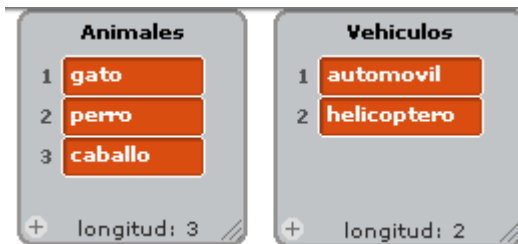
The code for 'En 5' consists of two event-driven blocks. The first block, 'al presionar' (when green flag clicked), contains: 'esconder' (hide), 'ir a x: -2 y: -112' (go to x: -2, y: -112), and 'fijar nombre a caballo' (set name to horse). The second block, 'al recibir nuevo objeto' (when new object received), contains: 'ir a x: -2 y: -112' (go to x: -2, y: -112), 'esconder' (hide), and a conditional 'si numero objeto = 5' (if object number = 5) followed by 'mostrar' (show).

Se modifican las listas

<EN CAJA1>

The code for 'EN CAJA1' is a single event-driven block 'al presionar' (when green flag clicked) containing four list operations: 'borrar todos de Animales' (delete all from Animales), 'insertar gato en último de Animales' (insert gato at the end of Animales), 'insertar perro en último de Animales' (insert perro at the end of Animales), and 'insertar caballo en último de Animales' (insert caballo at the end of Animales).

<EN CAJA2>



ACTIVIDAD

1. Analizar el ejercicio anterior y describir que fue lo que se realizó y para que crees que lo utilizarías.
2. A partir del ejemplo realizado, crear un juego o animación donde utilices el concepto de LISTAS, recuerda la importancia de manejar cada una de las temáticas vistas en clases anteriores.

**LA CREATIVIDAD JUEGA UN PAPEL IMPORTANTE EN TU VALORACIÓN
COMO ESTUDIANTE, HAS USO DE ELLA.**