

## GUIA DE PRÁCTICAS 05

## ESTRUCTURA ITERATIVA DESDE (FOR) - MIT APPINVENTOR

**1. Objetivos: -**

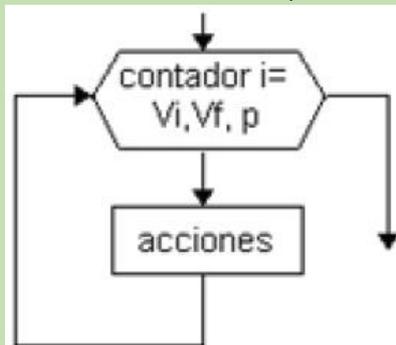
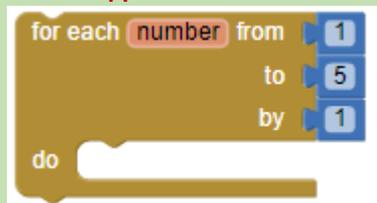
- Aplicar las estructuras iterativas DESDE (FOR) en aplicaciones.
- Implementar aplicaciones con la estructura iterativa DESDE (FOR).

**2. Recursos y materiales:**

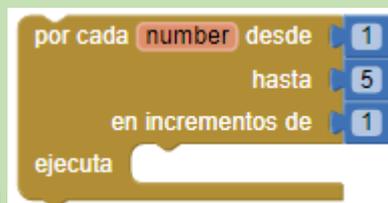
- Computador con conexión a internet.
- Acceso al aplicativo MIT AppInventor.

**INFORMACIÓN TEÓRICA****Estructura iterativa DESDE o PARA (FOR)**

A diferencia del ciclo MIENTRAS que utiliza centinelas, el ciclo PARA tiene control automático y cuando se rompe el ciclo, el programa continúa su flujo normal. La estructura para (desde) comienza con un valor inicial de la variable índice. Esta se incrementa de uno en uno y si este nuevo valor no excede al final, se ejecutan de nuevo las acciones. El incremento de la variable índice siempre es 1 a menos que se indique expresamente lo contrario. La variable índice de control normalmente será de tipo entero y es normal emplear como nombres I, J, K.

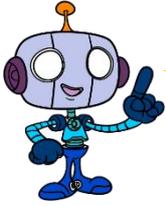
**Control en AppInventor:**

o en español



Repite la acción o conjunto de acciones situadas dentro del bloque mientras la evaluación del test (expresión lógica) de como resultado verdadero. Comienza evaluando la condición, si es verdadero realiza el proceso incluido dentro del bloque. Cuando el resultado es falso el bloque termina.

## 3. PROCEDIMIENTOS:



Implementa aplicaciones con MIT AppInventor.

## Programa que halla la sumatoria de series:

Tenemos que crear una interface para para que la aplicación se relacione con el usuario, le solicite el ingreso de un número y según la imagen que se muestra, se calcule la suma según el botón que se pulse.

**Paso 1:** Implemente la siguiente interface en MIT AppInventor. Considere los siguientes objetos y sus propiedades

**Paso 2:** En el editor de bloques escribe el siguiente código.

- a) Ingresar el siguiente código, que crea las variables **numero**, **suma** y **contador** y le asigna el valor inicial de cero.

```
initialize global numero to 0
initialize global suma to 0
```

b) Implemente el código para el Boton1 que realiza la sumatoria S1.

```

when Button1 .Click
do
  set global suma to 0
  set global numero to TextBox1 . Text
  for each X from 1
    to get global numero
    by 1
  do
    set global suma to get global suma + get X
  set Label2 . Text to get global suma

```

c) Implemente el código para el Boton2 que realiza la sumatoria S2.

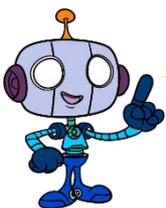
```

when Button2 .Click
do
  set global suma to 0
  set global numero to TextBox1 . Text
  for each X from 1
    to get global numero
    by 1
  do
    set global suma to get global suma + (2 * get X) - 1
  set Label2 . Text to get global suma

```

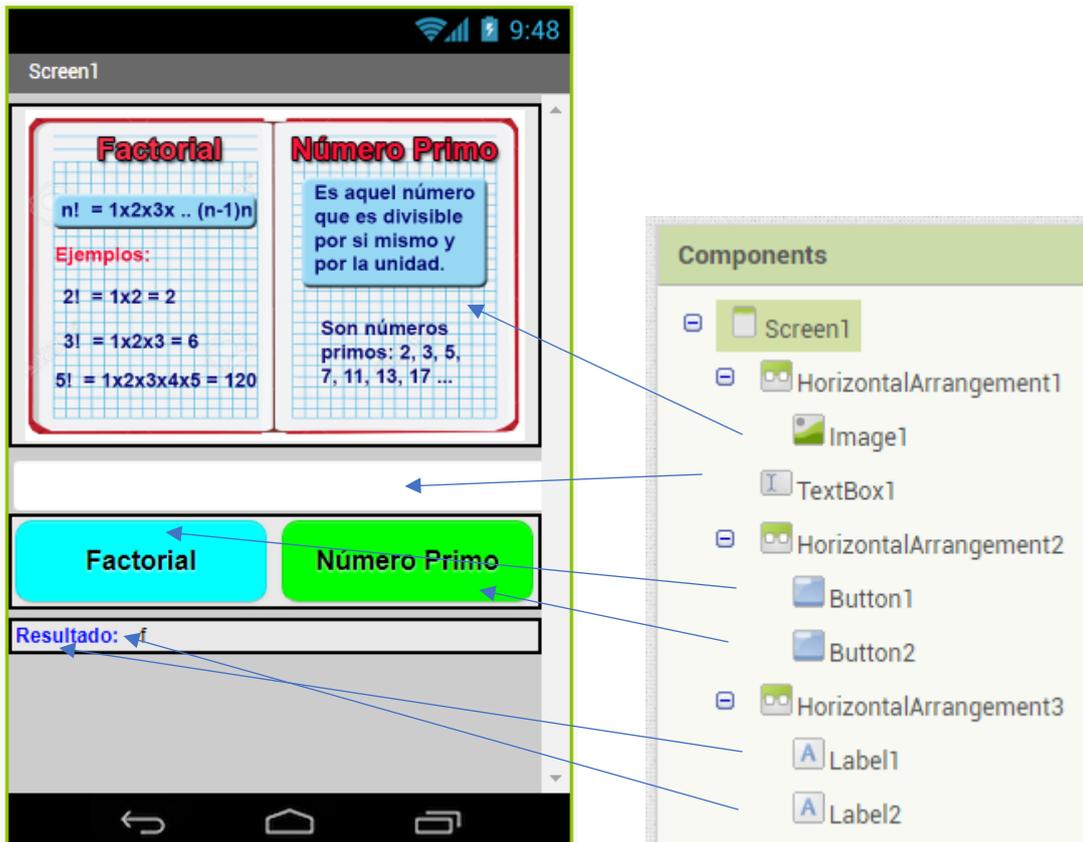
d) Implemente el código para el Boton3 que realiza la sumatoria S3.

e) Implemente el código para el Boton4 que realiza la sumatoria S4.



**Ahora implementaremos una nueva aplicación**

**Paso 3:** Crea un nuevo proyecto e implementa la siguiente interface. Descarga las imágenes del aula virtual.



Modifica los atributos de los objetos, para que se vea igual que la imagen.

**Paso 6:** Implementa el siguiente código.

- a) En el boton1 implementa el siguiente código, para hallar el factorial de un número.

```

initialize global num to 0
initialize global fac to 1

when Button1 .Click
do
  set global fac to 1
  set global num to TextBox1 . Text
  for each X from 1
    to get global num
    by 1
  do
    set global fac to get global fac × get X
  set Label2 . Text to get global fac

```

- b) En el boton2 implementa el siguiente código, para hallar si el número ingresado es un número primo o no.

```
initialize global cont2 to 0

when Button2 .Click
do
  set global cont2 to 0
  set global num to TextBox1 . Text
  for each X from 1
    to get global num
    by 1
  do
    if modulo of get global num ÷ get X = 0
    then set global cont2 to get global cont2 + 1

  if get global cont2 > 2
  then set Label2 . Text to join get global num " NO es primo "
  else set Label2 . Text to join get global num " SI es primo "
```

The image shows a Scratch script for checking if a number is prime. It starts with an 'initialize global' block for 'cont2' set to 0. A 'when Button2 .Click' event triggers a 'do' block. Inside, 'global cont2' is reset to 0, and 'global num' is set to the text of 'TextBox1'. A 'for each' loop iterates 'X' from 1 to 'global num' with a step of 1. Inside the loop, an 'if' block checks if the remainder of 'global num' divided by 'X' is 0. If true, 'global cont2' is incremented by 1. After the loop, another 'if' block checks if 'global cont2' is greater than 2. If true, 'Label2 . Text' is set to 'global num' followed by ' NO es primo '. Otherwise, it is set to 'global num' followed by ' SI es primo '.