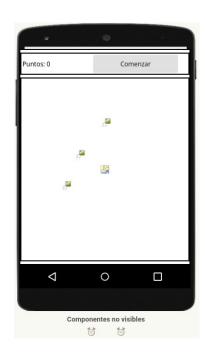
JORNADA CULTURAL

"Cazar al gato"

Diseño:

- 1. Entra a Appinventor: https://appinventor.mit.edu/
 Si no tienes cuenta: https://code.appinventor.mit.edu (anota el código para volver a entrar)
- 2. Crea un proyecto nuevo con el nombre "Cazar_al_gato".
- 3. **Screen1**. Propiedades:
 - o Orientación de pantalla: Sensor
 - o MostrarBarraEstado: Deactivado
 - o TítuloVisible: Desactivado
- 4. Añade los siguientes elementos:
 - Disposición → Disposición Tabular.
 - **Ancho**: Ajustar al contenedor.
 - Columnas: 3 Filas: 1
 - Interfaz de Usuario → Etiqueta (lb_Points).
 - **Ancho**: 80 px
 - **Texto**: Puntos: 0
 - Interfaz de Usuario → Botón (bt_Comenzar).
 - **Ancho**: 50%
 - **Texto**: Comenzar
 - Dibujo y animación → Lienzo
 - **Ancho** y **Alto**: Ajustar al contenedor
 - o Dibujo y animación → SpriteImagen (spri_Ratoli)





- Dibujo y animación → SpriteImagen (spri_Queso)
- Dibujo y animación → SpriteImagen (spri_Gato)
- \circ Sensores → Reloj (rj_Disfraz)

■ IntervaloDelTemporizador: 250

■ **TemporizadorHabilitado**: Desactivado

Sensores → Reloj (rj_Queso)

■ IntervaloDelTemporizador: 8000

■ **TemporizadorHabilitado**: Desactivado

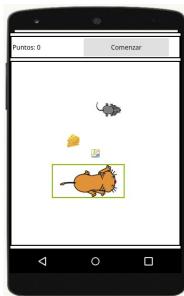
5. Descarga las imágenes que necesitas del siguiente enlace:

https://github.com/fjarabajurado/cazaalgato

Puedes descargarlas una a una o descargar el archivo comprimido "imagenes.zip" y descomprimirlo.

- 6. Sube las imágenes al proyecto que estamos creando.
- 7. Modifica las propiedades de los sprites:

	spri_Ratoli	spri_Gato	spri_Queso
Foto:	ratoli0.png	gato.png	queso.png
Alto:	28 px	60 px	25 px
Ancho:	60 px	133 рх	25 px



Ya tenemos el diseño. A continuación vamos a programar. Durante la programación ve probando el funcionamiento cuando lo consideres conveniente. Pasa a la pantalla Bloques.

Programación:

A) Inicialización del juego

1. Crea dos variables (**velocidad**, **puntos**) e inicialízalas a 0.

```
inicializar global velocidad como 0
inicializar global puntos como 0
```

2. Crea un procedimiento con el nombre **Actualizar_Posicion** que hará que el gato mire todo el tiempo hacia el ratón.

```
como (Actualizar_Posicion)
ejecutar (Ilamar spri_gato v .ApuntarHacia
objetivo (spri_Ratoli v
```

3. Crea un procedimiento con el nombre **Inicia_Queso** que colocará el queso en una posición aleatoria de la pantalla y lo hará visible.

```
ejecutar

poner spri_queso v . X v como centero aleatorio entre 0 y Screen1 v . Ancho v

poner spri_queso v . Y v como centero aleatorio entre 0 y Screen1 v . Alto v

poner spri_queso v . Visible v como verdadero v
```

4. Crea un procedimiento con el nombre **Inicia_Ratoli** que colocará al ratolí en una posición aleatoria de la pantalla y lo hará visible.

```
ejecutar poner spri_Ratoli v . X v como centero aleatorio entre 0 y Screen1 v . Ancho v - spri_Ratoli v . Ancho v poner spri_Ratoli v . Y v como centero aleatorio entre 0 y Screen1 v . Alto v - spri_Ratoli v . Alto v poner spri_Ratoli v . Visible v como centero aleatorio entre verdadero v
```

5. Crea un procedimiento con el nombre **Inicia_Gato** que colocará al gato en una posición aleatoria de la pantalla, lo hará visible y le dará una velocidad inicial de 0.

```
como Inicia_Gato
ejecutar

poner Spri_gato V . XV como entero aleatorio entre (0 y Screen1 V . Ancho V - Spri_gato V . Ancho V

poner Spri_gato V . Visible V como verdadero V

poner Spri_gato V . Visible V como verdadero V

poner Spri_gato V . Velocidad V como verdadero V
```

6. Crea un procedimiento con el nombre **Iniciar** que llamará a los otros procedimientos de inicialización, además de iniciar las variables (puntos y velocidad) y reiniciar el texto que muestra los puntos.

```
ejecutar Llamar Inicia_Ratoli v

Llamar Inicia_Queso v

Llamar Inicia_Gato v

Llamar Actualizar_Posicion v

poner global velocidad v a 0

poner global puntos v a 0

poner lb_Points v . Texto v como ( " Puntos: 0 "
```

7. Al comenzar el juego, es decir, cuando la pantalla Screen1 se inicie, se llamará al procedimiento Iniciar.

```
cuando Screen1 v .Inicializar
ejecutar Llamar Iniciar v
```

8. Para finalizar, la hacer un clic en el botón Comenzar, también debe llamarse al procedimiento Iniciar.

```
cuando bt_Comenzar v .Clic
ejecutar Llamar Iniciar v
```

B) Arrastrar el ratolí

1. Capturar el evento de **arrastrar** el sprite **spri_Ratoli**. Debe actualizar la posición X e Y del sprite.

```
cuando spri_Ratoli . Arrastrado

XInicial YInicial XPrevio YPrevio XActual YActual

ejecutar poner spri_Ratoli . X como tomar XActual . (spri_Ratoli . Ancho . / 2)

poner spri_Ratoli . Y como tomar YActual . (spri_Ratoli . Alto . / 2)

Llamar Actualizar_Posicion .
```

C) Colisión del objeto spri_Queso con el objeto spri_Ratoli

1. Al colisionar la imagen del queso con la imagen del ratolí hay que **aumentar** los **puntos**, la **velocidad** del gato y **actualizar el texto** que aparece arriba con los puntos obtenidos hasta el momento.

```
cuando spri_queso v .EnColisiónCon
 otro
                        llamar spri_queso ▼ .ChocandoCon
         🗯 si
                                                     otro
                                                            spri_Ratoli v
                    poner global velocidad v a
                                                omar global velocidad
                                                                                   1
                    poner global puntos v a
                                              tomar global puntos
                    poner | Ib_Points ▼ . Texto ▼ como
                                                                    Puntos:
                                                                   tomar global puntos
                    Llamar Inicia_Queso v
                    poner spri_gato v . Velocidad v
                                                            tomar global velocidad
                                                   como
```

D) Colisión del objeto spri_Gato con el objeto spri_Ratoli

1. Al colisionar los objetos **spri_Gato** y **srpi_Ratolí** se acabará la partida. Al producirse la colisión se ocultarán los tres sprites: **spri_Ratoli**, **spri_Gato** y **spri_Queso**, de forma que no se pueda seguir jugando hasta que se pulse el botón **Comenzar**.

```
cuando spri_gato v .EnColisiónCon

otro
ejecutar si llamar spri_gato v .ChocandoCon

otro spri_Ratoli v

entonces poner spri_Ratoli v . Visible v como falso v

poner spri_gato v . Visible v como falso v

poner spri_queso v . Visible v como falso v
```

E) Animar el sprite del Gato y del Ratolí

1. Vamos a utilizar el reloj rj_Disfraz para animar spri_Gato. Crea e inicia una variable con el nombre **img_gato** con un valor inicial de cero.

```
inicializar global (img_gato) como 🚺 🔾
```

2. Modifica el procedimiento **Iniciar** para que la variable **img_gato** tome el valor **0**. y se habilite el reloj **rj_Disfraz**.

```
ejecutar

Llamar Inicia_Ratoli v

Llamar Inicia_Queso v

poner global img_gato v a 0

poner rj_disfraz v . TemporizadorHabilitado v como verdadero v
```

3. Modifica el evento de **colisionar** el **gato** con el **ratolí**, de forma que se **dehabilite** el temporizador **rj_Disfraz**.

```
cuando spri_gato v .EnColisiónCon

otro
ejecutar

otro
entonces

poner spri_Ratoli v . Visible v como falso v

poner spri_gato v . Visible v como falso v

poner spri_queso v . Visible v como falso v

poner rj_disfraz v . TemporizadorHabilitado v como falso v
```

4. Cada vez que ocurra un **evento del reloj** rj_Disfraz se debe cambiar la imagen del gato. Los nombres de las imágenes son: gato0.png, gato1.png, gato2.png y gato3.png. Al llegar a la última debe volver a comenzar por la primera. El nombre e la imagen lo obtenemos mediante la unión de "gato" + img_gato + ".png". Cuando la variable **img_gato** tome el valor 4 debemos asignarle el valor 0 para volver a comenzar. La imagen del ratolí solo tiene dos imágenes, por lo que debe reiniciarse a cero cada 2.

```
cuando rj_Disfraz v .Temporizador
ejecutar poner global img_gato v a tomar global img_gato v + 1

si tomar global img_gato v = v 4
entonces poner global img_gato v a 0

poner spri_Gato v . Foto v como ( unir ( " gato " tomar global img_gato v " .png "
poner spri_Ratoli v . Foto v como ( unir ( " ratoli " resto de v tomar global img_gato v † 2

" .png "
```

F) Hacer que el queso cambie de posición cada 8 segundos

1. Capturar el evento del temporizador del queso (**rj_Queso**) de forma que se llame al procedimiento **Inicia_Queso**.

```
cuando rj_Queso v .Temporizador
ejecutar Llamar Inicia_Queso v
```

2. Es más interesante si el queso está cada vez menos tiempo en una posición. Así, cada vez que se captura un queso (**colisión** entre **queso** y **ratolí**) debe disminuir el tiempo del temporizador **rj_Queso**.

```
cuando spri_Queso v .EnColisiónCon
otro
ejecutar

si llamar spri_Queso v .ChocandoCon
otro spri_Ratoli v
entonces

poner global velocidad v a tomar global velocidad v + 1

poner global puntos v a tomar global puntos v + 1

poner lb_Points v . Texto v como tomar global puntos v

Llamar Inicia_Queso v
poner spri_Gato v . Velocidad v como tomar global velocidad v

poner rj_Queso v . IntervaloDelTemporizador v como

8000 - tomar global puntos v × 100
```

3. Modificar el procedimiento **Iniciar** para poner el **InervaloDelTemporizador** a 8.000 (*8 segundos*) y habilitarlo.

```
ejecutar

Llamar Inicia_Ratoli 
Llamar Inicia_Queso 
Llamar Inicia_Gato 
Llamar Actualizar_Posicion 

poner global velocidad a 0

poner global puntos a 0

poner global img_gato a 0

poner rj_Disfraz  TemporizadorHabilitado como verdadero 

poner rj_Queso  IntervaloDelTemporizador como 8000
```

4. Al **colisionar** el **gato** con el **ratolí**, el reloj *rj_Queso* debe deshabilitase de forma que pare la animación.

G) Ratolí mirando en la dirección de desplazamiento

1. Para que el **ratolí** mire en la dirección de desplazamiento crea una **variable** con el nombre **angulo** y dale un valor inicial de **0**.

```
inicializar global angulo como 🚺 🔾 🔾
```

2. Modifica el procedimiento **Iniciar** de forma que la variable **angulo** tome el valor 0 y el **ratolí apunte** en la **Dirección** del valor de la **variable angulo**.

3. Para **calcular la dirección en la que debe mirar el ratolí** al ser arrastrado utilizaremos una fórmula matemática (*angulo* = *arcotangente* (*inc Y* / *inc X*)) que adaptaremos al sistema de coordenadas de la pantalla del juego. Observa que debes poner el código antes de llamar a **Actualizar_Posición**.

```
cuando spri_Ratoli v .Arrastrado
XInicial YInicial XPrevio YPrevio XActual YActual
        🔯 si
                                                              #7 0
                           mar XActual 🔻
                        global angulo 🔻
                                                            YActual 🔻
                                                                          tomar YPrevio 🔻
                                                                                                       XActual v
                                                                                                                          XPrevio ▼
                 poner spri_Ratoli 🗸 . Dirección 🗸 como 📗
                                                             tomar (global angulo v + 90)
                                                                     yv tomar (XActual v >v)
                                       tomar global angulo v >v 0
                           poner spri_Ratoli 🔻 . Dirección 🔻 como 📗
                                                                       tomar (global angulo 🔻 🕴 -90
                           poner spri_Ratoli v . Dirección v como
                                                                  tomar global angulo v
```

Instalar en el móvil:

- 1. Ir al menú **Generar**.
- 2. Hacer clic en App de Android (.apk).
- 3. Aparecerá un **código QR**. Leerlo con el móvil y aceptar la instalación.