

Contribuyendo SugarLabs con Python

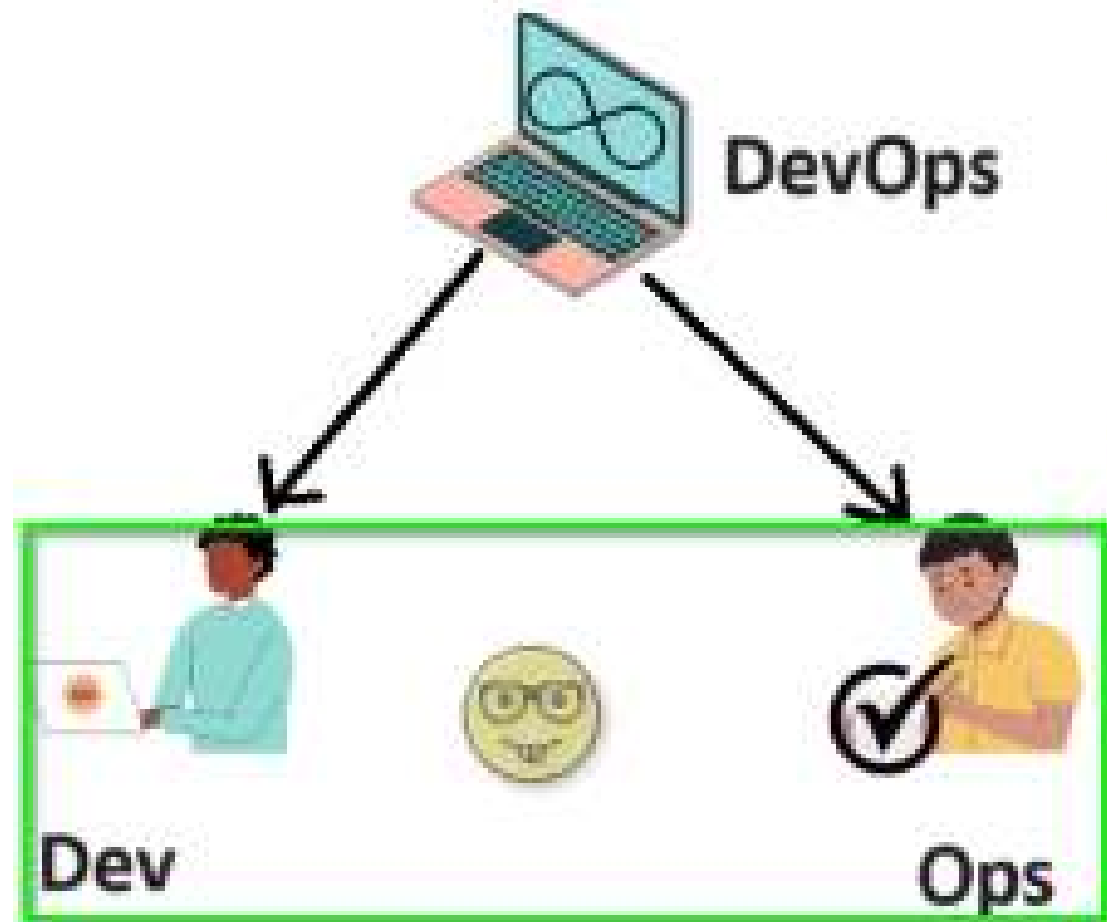
¿Quién soy?



¿Qué te motiva a levantarte
cada mañana?

G HAHA TGCCA	TTCTCCTGCC	TCAGCCTCCC	GAGTAGCTGG	GACTACAGGC
GCCCGCCACC	ACGCCCAGCT	AATTTTTTGT	ACTTTTAGTA	GAGACGGGGT
TTCACCGTGT	TAGCCAGGAT	GGTCTCGATC	TCCTC HAHA C	GTGATCCGCC
CGCCTCAGCC	TCCCAAAGTG	CTGGGGGATAC	AGGCGTGAGC	CACTGGCTCG
TTCTTTTTTTT	TTT HAHA TTT	TTTTGAGACA	GAGTCTGGCT	CTGTCTCCCA
GGCTGGAGTA	CAGTGGTATG	ATCTCGGCTC	ACTGCAGGCT	CTGCCTCCGG
GTTCATGCCA	TTCTCCTGCC	TCAGCCTCCC	GAGTAGCTGG	GACTAC HAHA
GCCCGCCACC	ACGCCCAGCT	AATTTTTTGT	ACTTTTAGTA	GAGACGGGGT
TTCACCGTGT	TAGCCAGGAT	HAHA TCGATC	TCCTCACCCAC	GTGATCCGCC
CGCCTCAGCC	TCCCAAAGTG	CTGGGGGATAC	AGGCGTGAGC	CACTGGCTCG
TTCTTT HAHA	TTTTTTTTTTT	TTTTGAGACA	GAGTCTGGCT	CTGTCTCCCA

¿En que trabajo?



¿Qué aprendió en sus
empleos anteriores?

Comunidades Tecnológicas



Roothouse
miembro

#TechnoLatinas
miembro



blogger
miembro



¿Pueden ayudarte a
resolver tus problemas?

¿Quienes somos?



¿Esta la educacion
en Bolivia mejorando?

Estado de la educación

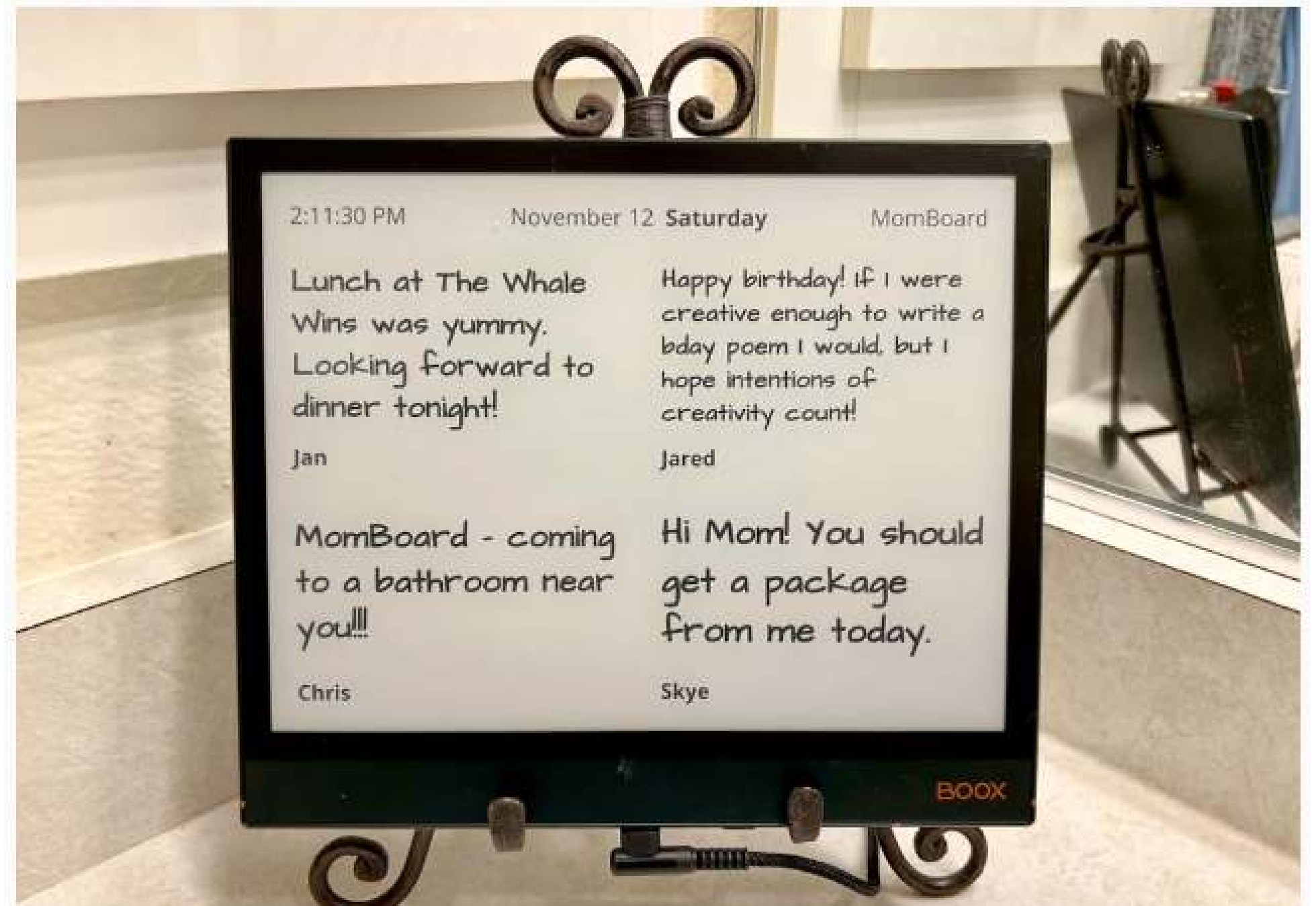
- La tecnología podría estar empeorando la educación
- Dejemos de culpar a la tecnología educativa por nuestra desigualdad educativa actual.



¿Esta la educación en Bolivia mejorando?

MomBoard: pantalla de tinta electrónica para un padre con amnesia

12 de noviembre de 2024



Software para la educación



¿Qué nos ofrece el software educativo?

FOSS educativo

- Free Licensed
- Open
- Source
- Software



**Nextcloud Introduces Easy
Website Builder for Education
Edition with PicoCMS**



¿Qué tipo de
software educativo usa
regularmente?

Porque es importante software en la educación



¿Cómo ayudan los videojuegos en la educación?

Porque es importante software en la educación

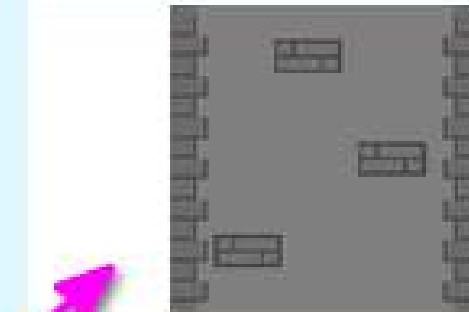
Create a Wall

Docs: [show_graphic](#)

```
1 from pytamaro import show_graphic
2 from castle_components import wall
3 from colors import red, yellow, green, cyan, blue, purple, pink, grey
4 from wall_borders import borders, no_borders
5 from wall_properties import door, double_door, window, arched_window
6 from wall_properties import bricks, ivy, crack, nothing
7
8
9 my_wall = wall(grey, borders, bricks)
10
11 show_graphic(my_wall)
```

RUN 

CREATE A WALL



MY CURRICULUM

 [Program Your Own Castle](#)

Producing Castle Components

COMPETENCIES

PyTamaro

Learn **Programming** in Python with **Graphics**

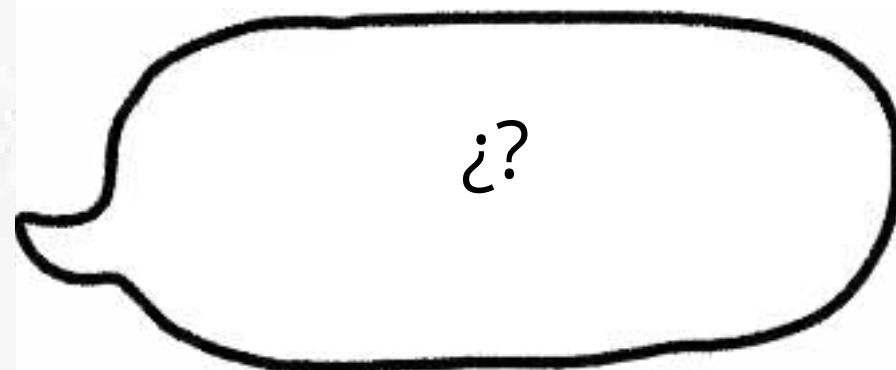
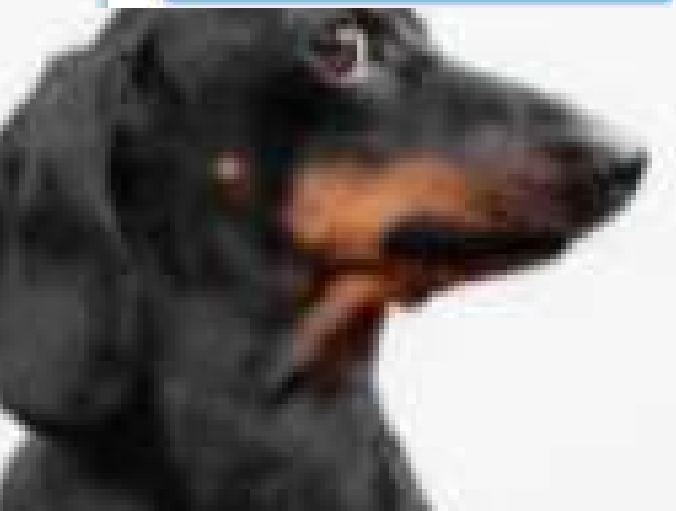
Start Simple By Calling Functions

```
root = sqrt(2)
print(root)
```

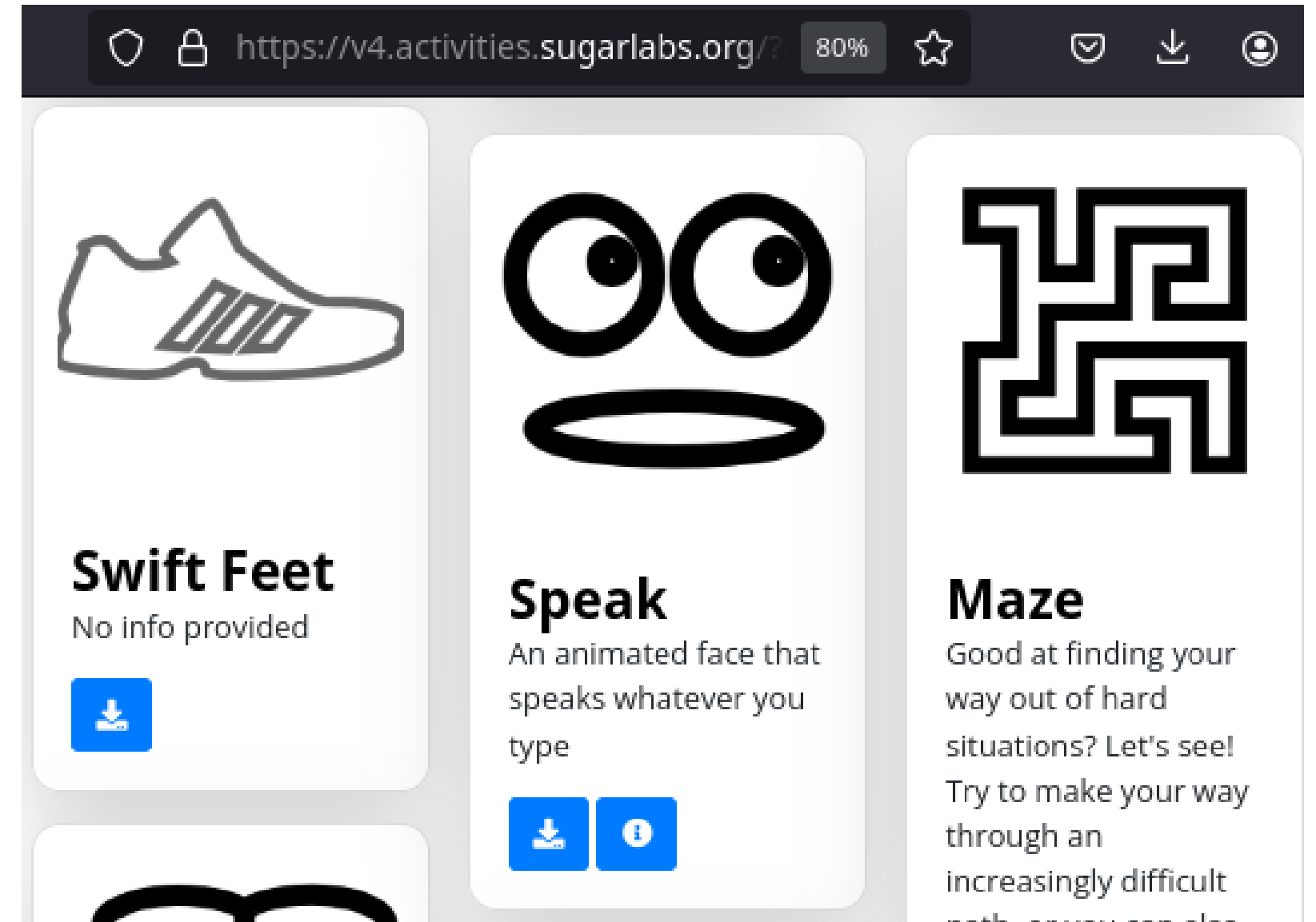
1.4142135623730951

```
field = rectangle(100, 60, green)
show_graphic(field)
```

TRY IT!



SugarLabs



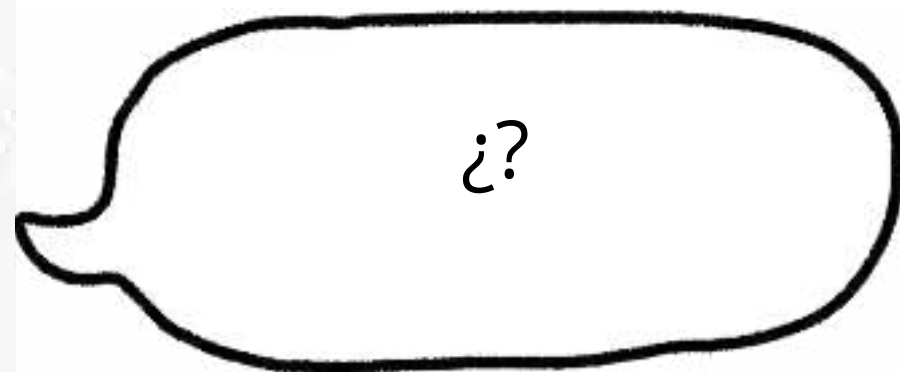
proyecto de software
administrado por la
comunidad

Características de Sugar



Top languages

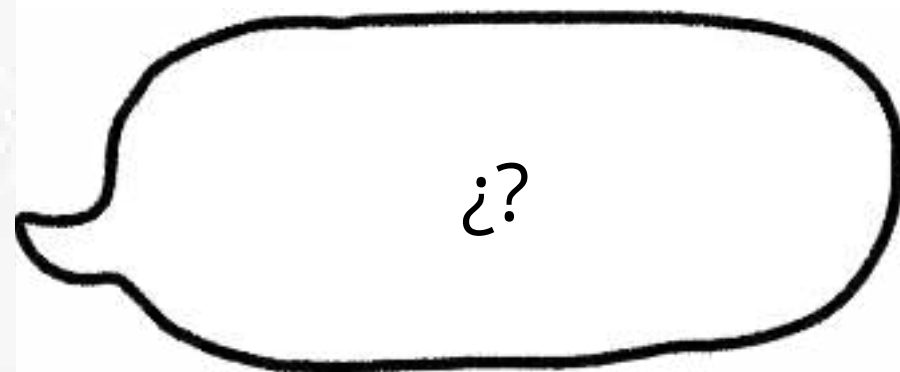
- Python ● JavaScript
- Shell ● HTML
- TypeScript



Mas proyectos educativos



Languages



Parte II: Preparación

Requisitos:

1. Paquete de SugarLab actual
2. Crear una actividad de SugarLab .
3. La actividad debe ser portable
4. Documentar la actividad



¿Qué tipo de software educativo usa regularmente?

Requisito 1

1. Paquete de SugarLab actual
 - a. Encontrar el repositorio
 - b. Instalar el programa.
 - c. Leer la documentacion de los primeros pasos:

link: <https://github.com/sugarlabs/sugar-toolkit-gtk3>

link: <https://github.com/sugarlabs/sugargame>



¿Qué tipo de software educativo usa regularmente?

Requisito 2

1...

2. Crear una actividad de SugarLab con:

a. Contenido Educativo

b. Contenido de Texto



¿Qué tipo de software educativo usa regularmente?

walterbender commented on Apr 8 • edited by chimosky ▾

Ticket Contents

Description

While Sugar has lots of activities, you can never have enough math game of ideas (See <https://www.cut-the-knot.org/>) for Sugar activities. This proj based on some of Bogomolny's ideas (or some of your own ideas).

We are interested in these games:

- [Lewis Carroll's Game of Logic](#)
- [Pascal's Triangle](#)
- [Nim](#)
- [The Candy Game](#)
- [Number Guessing Game](#)
- [Latin Squares](#)
- [3 Utilities Puzzle](#)
- [Goat Cabbage Wolf](#)

In the proposal, please mention some specific ideas for games.

Requisito 3

1...

2....

3. La actividad debe ser portable

a. convertir un paquete 📦

link: <https://github.com/flathub/org.sugarlabs.TurtlePondActivity>



¿Qué tipo de software educativo usa regularmente?

34 results (248 ms) in [flathub](#) X

 [flathub/org.sugarlabs.AbacusActivity](#)

☆ 0 · Updated on May 31

 [flathub/org.sugarlabs.Pippy](#)

☆ 0 · Updated on Jul 30

 [flathub/org.sugarlabs.ReadETexts](#)

☆ 0 · Updated on Jun 19

Requisito 4

1...

2....

3.....

4. Documentar el proceso.

link:<https://github.com/sugarlabs/DM>
P/issues/6



¿Qué tipo de
software educativo usa
regularmente?

Parte II: Desarrollo

VM



Configure the package manager

Scanning your installation media finds the label:

Debian GNU/Linux 12.8.0 _Bookworm_ - Official amd64 NETINST with firmware 20241109-11:04

You now have the option of scanning additional media for use by the package manager (apt). No have any additional media, this step can just be skipped.

If you wish to scan more media, please insert another one now.

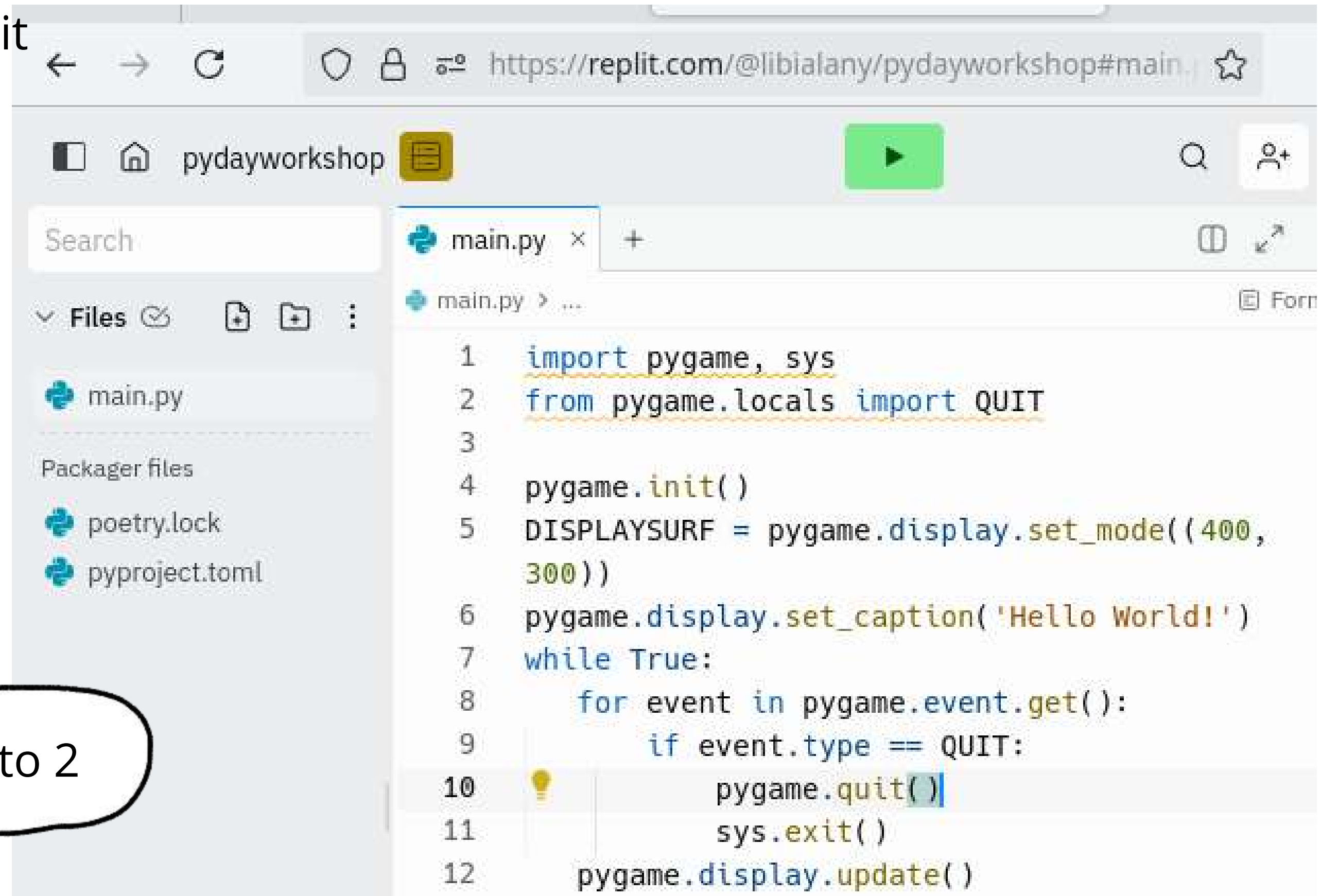
Scan extra installation media?

☒ No

☐ Yes

Crear una actividad de SugarLab

crear su proyecto en repl.it



The screenshot shows a web browser window with the URL `https://replit.com/@libialany/pydayworkshop#main`. The workspace is titled 'pydayworkshop' and contains a file named 'main.py'. The code in 'main.py' is as follows:

```
1 import pygame, sys
2 from pygame.locals import QUIT
3
4 pygame.init()
5 DISPLAYSURF = pygame.display.set_mode((400,
6 300))
7 pygame.display.set_caption('Hello World!')
8 while True:
9     for event in pygame.event.get():
10         if event.type == QUIT:
11             pygame.quit()
12             sys.exit()
13         pygame.display.update()
```

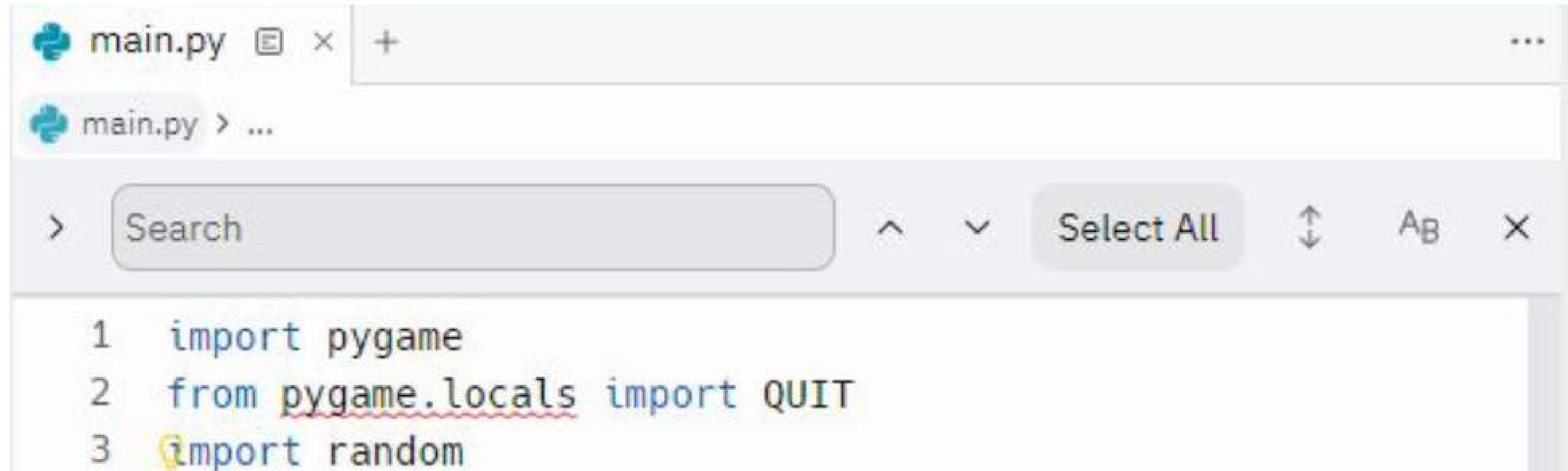
Requisito 2



Instalación

Tipo de importación **pygame** y ejecutarlo.

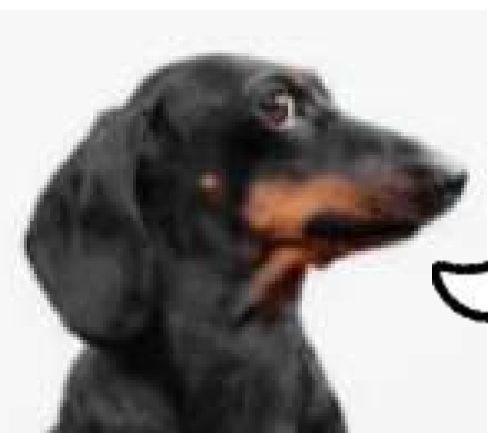
El juego debe instalarse automáticamente para usted. (En otros entornos, puedes necesidad de instalarlo usando pip)



The screenshot shows a code editor window with a tab labeled 'main.py'. The editor contains the following Python code:

```
1 import pygame
2 from pygame.locals import QUIT
3 import random
```

The code is syntax-highlighted, with 'import' in blue, 'pygame' in green, 'from' in blue, 'pygame.locals' in green, 'import' in blue, 'QUIT' in blue, and 'random' in green. The word 'pygame' is underlined in red. The editor has a search bar at the top with the text 'Search' and a 'Select All' button. The window title bar shows 'main.py' and a close button.



Requisito 2

Inicialización del pygame

link codigo:

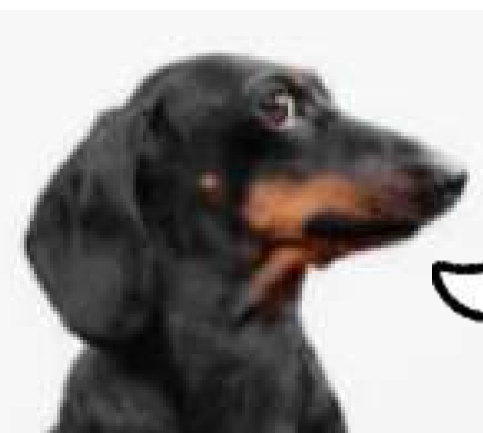
<https://shorturl.at/Wuyse>

```
import pygame

pygame.init()
pygame.display.set_caption("Some Title Goes Here")
canvas = pygame.display.set_mode((1200, 800))

def main():
    pass # We'll put future code here...

main()
```



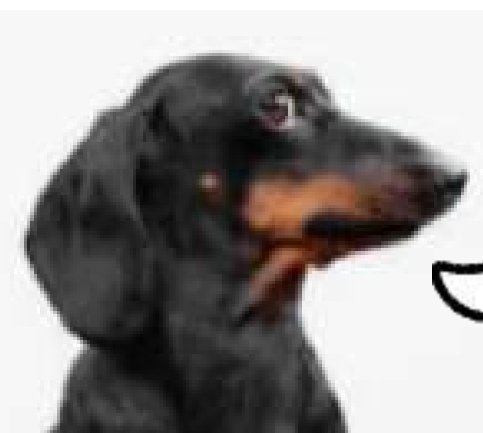
Requisito 2

Inicialización del Pygame

Añadir un bucle principal a la función main()

link codigo: <https://shorturl.at/Wuyse>

```
def main():  
    while True:  
        for event in pygame.event.get():  
            if event.type == pygame.QUIT:  
                pygame.quit()  
                exit()  
        pygame.display.update()
```



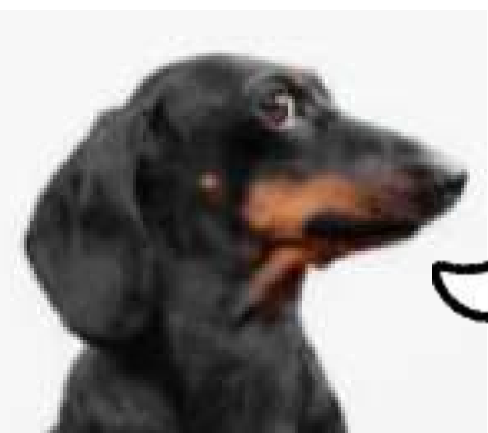
Requisito 2

Printing text on screen

Añadir dentro de la función principal pero antes de su bucle

link codigo: <https://shorturl.at/Wuyse>

```
def main():  
    configuraciones fijas  
    my_font = pygame.font.SysFont('Calibri', 20)  
    rendered_text = my_font.render('This is some text', True, (0, 255))  
    canvas.blit(rendered_text, (200, 200))  
  
while True:
```



Requisito 2

Dibujo rectángulos

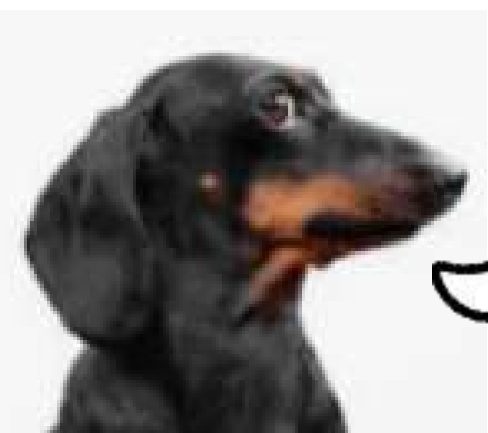
Añáde esto después del código anterior pero antes del bucle:

link codigo: <https://shorturl.at/Wuyse>

```
canvas.blit(rendered_text, (200, 200))
```

```
pygame.draw.rect(canvas, (255, 255, 255), (300, 300, 50  
50))
```

```
while True:
```



Requisito 2

Detectar clicks del ratón

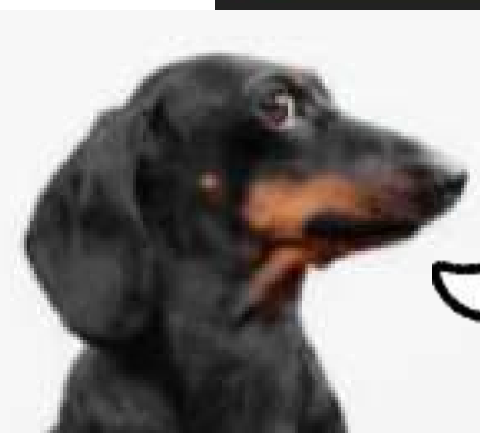
Añáde este if :

link codigo: <https://shorturl.at/Wuyse>

```
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            exit()

        if event.type == pygame.MOUSEBUTTONDOWN:
            canvas.fill((0, 0, 0))
            rendered_text = my_font.render('You clicked at '
+str(event.pos), True, (0, 255, 0))
            canvas.blit(rendered_text, (300, 300))

    pygame.display.update()
```



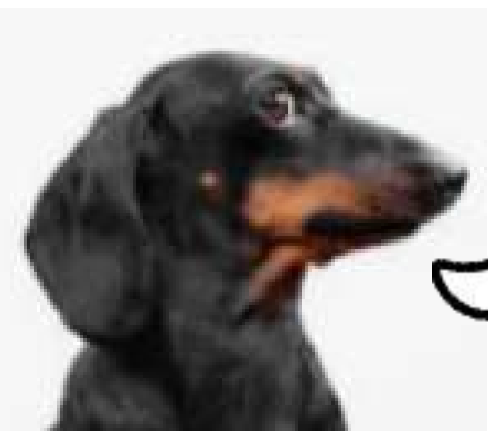
Requisito 2

Detectar clicks del ratón

Añáde este if :

link codigo: <https://shorturl.at/Wuyse>

- Buscar nuevos eventos (usando `pygame.event.get`)
- La pulsación del botón del ratón es un evento (**pygame**. BOTÓN DEL RATÓN HACIA ABAJO)
- Para verificar el click izquierdo / derecho, puede usar **event.button** (la izquierda es 1, el medio es 2, la derecha es 3)
- Para obtener la posición del click, puede usar **event.pos**
- Luego limpiamos el lienzo: **canvas.fill((0, 0, 0))** lo llena todo de negro
- Y finalmente redibujar nuestro texto



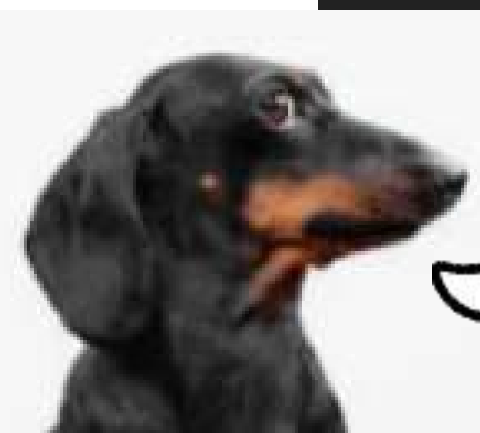
Requisito 2

Tiempo

Primero, vamos a mover nuestro viejo código en el bucle para no borrar cada click

link codigo: <https://shorturl.at/Wuyse>

```
def main():  
    my_font = pygame.font.SysFont('Calibri', 20)  
    clock = pygame.time.Clock()  
  
    while True:  
        clock.tick(30)  
        for event in pygame.event.get():
```



Requisito 2

Preguntas....



y donde esta
sugar???

Presentando SugarGame

1. SugarGame dentro de tu paquete de actividades.

```
mkdir sugargame  
cd sugargame  
git clone https://github.com/libialany/pydaytalk2024.git
```

2. Programa PyGame independiente a una actividad. Archivos [TestGame.py](#) y [TestActivity.py](#).

```
cd code/parte3/test && python TestActivity.py
```

3. Compilar

```
./setup.py dev
```

4. Ejecutar

```
cd ~/Activities/ && sugar-activity3 SugargameTest.activity
```



Requisitos 2. (a)

Preguntas....



y como hacemos
eso???

Sugar Actividad a partir de un PyGame

```
#!/usr/bin/env python
import pygame
import gtk
```

```
# Pump GTK messages.
while gtk.events_pending():
    gtk.main_iteration()
```



Requisitos 2. (a)

Portabilidad

Un paquete xo contiene:

- ~~Un archivo **MANIFEST** que enumera todo el contenido del paquete~~
- **Un archivo Activity.info** que tiene atributos que describen la actividad.
- **Un archivo de icono** (en formato SVG)
- ~~Archivos que contienen traducciones de las cadenas de texto.~~
- **El código del programa** para ejecutar la Actividad.



Requisito 3

Portabilidad

Creación de la primera actividad de Sugar:

- Hacer una versión independiente fuera del entorno de Sugar.
- Herencia.

Hay una pieza de código Python llamada actividad de Sugar. Actividad que es el mejor padre que una actividad.

- Empaquetar la actividad
- Convertirlo en algo que pueda ser ejecutado bajo Sugar y distribuido como un archivo .xo. Esto implica crear un MANIFEST, activity.info, setup.py, y crear un icono.



Requisito 3

Portabilidad

- Unidades de objetos.
- Un objeto como una colección de campos o atributos.
- Todas las **actividades de Sugar** amplían una clase de Python llamada **sugar.activity**. esta clase proporciona métodos que todas las actividades necesitan.
- **`__init__`**: Esto se llama cuando se inicia su actividad. Aquí es donde configurarás el usuario.
- **`read_file(self, file_path)`**
- **`write_file(self, file_path)`**



Requisito 3

Parte III: DISTRIBUCION

Package la actividad

- Adiciona un **setup.py** , cada setup.py es el inicio de la compilacion.

```
from sugar3.activity import bundlebuilder  
bundlebuilder.start()
```

Sugar utiliza el programa setup.py para diversos fines. Si ejecuta setup.py desde la línea de comandos, verá las opciones que se utilizan con él y lo que hacen.



Requisito 3

Crear actividad.info

```
code > parte3 > test > activity > ≡ activity.info
1  [Activity]
2  name = SugargameTest
3  bundle_id = org.sugarlabs.SugargameTest
4  exec = sugar-activity3 TestActivity.TestActivity
5  icon = activity-generic
6  activity_version = 1
```

name: Un nombre único que Sugar utilizará para referirse a su Actividad.

icon: Iconos son siempre .svg

exec: Esto le indica a Sugar cómo iniciar su actividad. Lo que dice es crear una instancia de la clase **TestActivity** que encontrará en el archivo.



Requisito 3

Instalar y ejecutar la actividad

1. SugarGame dentro de tu paquete de actividades.

```
mkdir sugargame  
cd sugargame  
git clone https://github.com/libialany/pydaytalk2024.git
```

2. Programa PyGame independiente a una actividad. Archivos [TestGame.py](#) y [TestActivity.py](#).

PARTE4

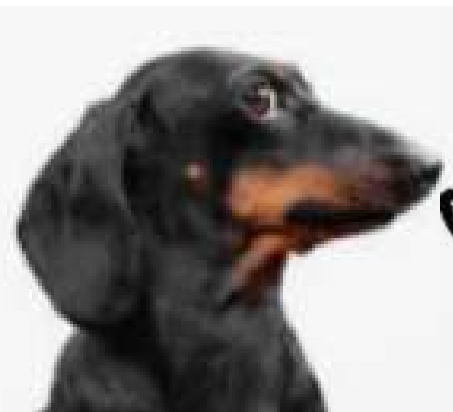
```
cd code/parte3/test && python TestActivity.py
```

3. Compilar

```
./setup.py dev
```

4. Ejecutar

```
cd ~/Activities/ && sugar-activity3 SugargameTest.activity
```



Añada su código
de actividad al
GU*

Parte IV: Documentacion

Documentar

¿Qué es esto?

La actividad de lectura permite que la <>. Tiene una interfaz sencilla y permite <>

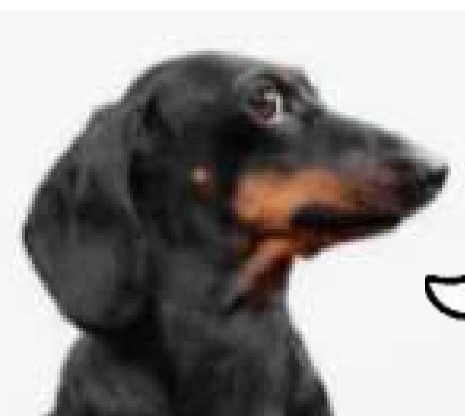
¿Cómo utilizar?

Leer es parte del escritorio de Sugar. Consulte;

- [Cómo obtener actividad en sugarlabs.org,](#)
- [Cómo utilizar Leer](#)

Ramas

- use la rama `main` para los sistemas Fedora y Debian, con Python 3 y Web
- use la rama `python2` y la serie de versiones v122.x para sistemas Fedora Ubuntu 18.04,



Requisito 4

GRACIAS