

Proyecto final

Desarrollo con Raspberry Pi y sensores.
Desarrollo educativo destinado a escuelas primarias:
Sopa de letras

J.P. Sánchez Magariños¹ B. Sbrancia¹ M. A. Cabral¹

¹Universidad Nacional de La Plata

Seminario Python 2019



Introducción

Descripción de la consigna

- Enfocado a niños y niñas que cursan el primer ciclo educativo de nivel primario.
- Educativo y divertido que los ayuda a desarrollar la atención y concentración, pero de una forma amena y lúdica.
- Fomentar la diversión para mejorar el aprendizaje.



Módulos utilizados

- PySimpleGUI
 - Interfaz Gráfica de Usuario Simple
 - Basado en tkinter
- Pattern
 - pattern.es
 - Procesamiento de lenguaje natural
 - Etiquetado gramatical
 - pattern.web
 - minería de datos web
 - Wiktionary
 - analizador HTML



Módulos implementados

grilla.py

Lógica en la que se basa el juego.

config.py

Método de entrada de datos.

sopa.py

Interfaz de usuario basada en la config y grilla



grilla.py

```
class Grilla:
    def __init__(self):
        self.n_intentos = 0
        self.celdas =
            ↪ [[{'key':str(j)+'_'+str(i), 'tipo':
            ↪ None,
            ↪ 'marcada':False, 'color':None,
            ↪ 'letra':char_vacio} for i in
            ↪ range(n_columnas)] for j in
            ↪ range(n_filas)]
        self.soluciones = []
```



grilla.py

```
Intentos: 1
Numero de palabras: 3

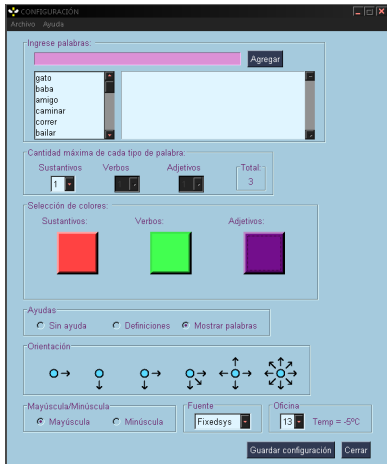
      0  1  2  3  4  5  6  7  8  9

0    L  E  T  R  I  N  A  M
1    G  F  A  V  S  T  N  E
2    B  A  D  L  E  H  J  A
3    P  A  T  A  L  E  A  R
4    W  E  Y  Q  S  C  Z  T
5    K  C  V  R  N  O  E  F
6    U  N  Z  Z  W  T  T  L
7    T  D  Q  C  Q  P  T  C

patalear  (0,3)(7,3)    feo      (7,5)(5,5)
letrina   (0,0)(6,0)
```



config.py

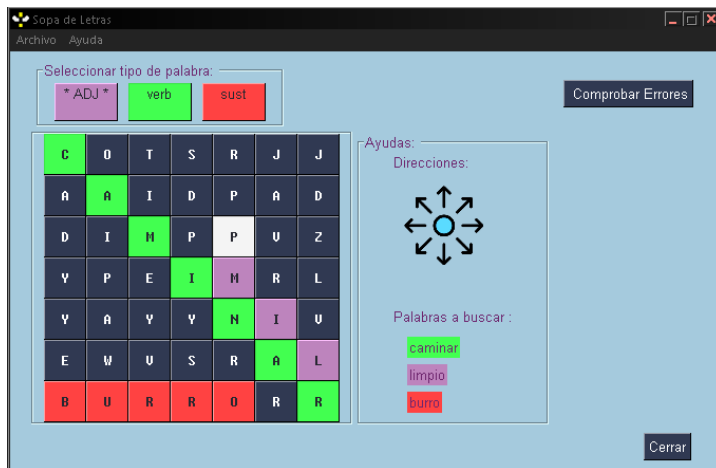


Lo que no se ve:

- Analizar palabras
- Reporte de errores
- Cargar archivo de configuración previa
- Cargar archivo de datos meteorológicos
- Definir colores de interfaz
- Evaluar colores elegidos para tipos
- Obtener lista de palabras definitiva



sopa.py



Problema Pattern

```
x= "danzar amigable barrilete asdfg"
print(pattern.es.tag(x, tokenize=True,
↳ encoding='utf-8'))
x = ''.join([random.choice(string.ascii_letters +
↳ string.digits) for n in range(32)])
print(pattern.es.tag(x, tokenize=True,
↳ encoding='utf-8'))
```

```
[('danzar', 'VB'), ('amigable', 'JJ'), ('barrilete', 'NN')
[('KD1mrZi6SwpZN25arEGv4TugHLAy2BYC', 'NN')]
```

Problema Pattern

```
from pattern.es import verbs, tag, spelling, lexicon
import string
palabra = 'Camino'

if not palabra.lower() in verbs:
    if not palabra.lower() in spelling:
        if (not(palabra.lower() in lexicon) and
            ↪ not(palabra.upper() in lexicon) and
            ↪ not(palabra.capitalize() in lexicon)):
            print('No se encuentra en pattern.es')
        else:
            print('La encontró en lexicon')
    else:
        print('La encontró en spelling')
else:
    print('La encontró en verbs')
```



Wiktionary

artilugio

Sumario [[ocultar](#)]

▼ Español

- ▶ Etimología
- ▶ Sustantivo masculino
- ▶ Traducciones
- ▶ Referencias y notas

Español [[editar](#)]

Etimología [[editar](#)]

Si puedes, incorpórala: ver cómo

Sustantivo masculino [\[editar \]](#)

Singular	Plural
artilugio	artilugios

1

Mecanismo, aparato, dispositivo. Se utiliza n

Trampa, ardid,

Traducciones [[editar](#)]

fideo

Sumario [[ocultar](#)]

▼ Español

- ▶ [Etimología](#)
- ▶ [Sustantivo masculino](#)
- ▶ [Locuciones](#)
- ▶ [Véase también](#)
- ▶ [Traducciones](#)
- ▶ [Referencias y notas](#)

Español [editar]

Etimología [[editar](#)]

Del mozárabe y árabe hispánico *fidáwš*

Sustantivo masculino [\[editar \]](#)

Singular	Plural
fideo	fideos

1 Gastronomía.

Pasta alimenticia delgada y alargada.

- **Hiperónimos:** masa, pasta.
- **Hipónimo:** tallarín.



Wiktionary

```
▶ <p>...</p>
▶ <h3>...</h3>
▶ <table class="inflection-table">...</table>
▼ <dl>
  <dt>1 Gastronomía.</dt> == $0
  ▶ <dd>...</dd>
</dl>
▼ <dl>
  <dt>2 Gastronomía.</dt>
  ▶ <dd>...</dd>
</dl>
▶ <dl>...</dl>
```

Wiktionary

```
from pattern.web import Wiktionary, plaintext
engine = Wiktionary(language="es")
sch=engine.search(palabra)
if sch != None:
    pos_1 = sch.source.find('<dt>1</dt>')
    if pos_1 == -1:
        pos_1 = sch.source.find('<dt>')
        pos_cierre_1 = sch.source.find('</dt>',pos_1+1)
        ↪ #busca a partir de pos 1
    else:
        pos_cierre_1 = pos_1
        .
        .
        .
definicion = plaintext(sch.source[pos_cierre_1:
    ↪ pos_2])[:pos_punto+1]
```



Combos

```
Traceback (most recent call last):
  File "run.py", line 56, in <module>
    config.main()
    .
    .
    .
  File "C:\~\Python36\lib\site-packages\PPySimpleGUI\PySimpleGUI.py",
line 5327, in PackFormIntoFrame
    combostyle.element_create(unique_field, "from", "alt")
  File "C:\~\Python36\lib\tkinter\ttk.py", line 468, in element_create
    spec, *opts)
_tkinter.TclError: Duplicate element _CANT_S_.TCombobox.field
```



Combos

```
# Creates a unique name for each field element(Sure there is a  
↪ better way to do this)  
unique_field = str(element.Key) + '.TCombobox.field'
```



Combos

```
# Creates a unique name for each field element(Sure there is a  
↪ better way to do this)  
unique_field = str(element.Key) + '.TCombobox.field'
```

```
unique_field = str(time.time()).replace('.', '') +  
↪ str(element.Key) + '.TCombobox.field'
```



A futuro

- Embellecimiento de la interfaz
 - Gráficas más amigables (*splash-art*).
 - Tachado de las palabras encontradas.
 - Incorporación de efectos sonoros.
- Salvar y cargar la partida.
- Portabilidad.



Para saber más I



Github del proyecto

[https://github.com/lossh/tpfinalpython.](https://github.com/lossh/tpfinalpython)



Informe completo

<https://github.com/lossh/Informe-TPFinal-Python>

