

Informe de Laboratorio 05

Tema: Python

Nota

Estudiante(s)	Escuela	Asignatura
Mariel Alisson Jara Mamani mjarama@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: I Código: 1702122

Laboratorio	Tema	Duración
05	Python	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 B	Del 27 Mayo 2024	Al 31 Mayo 2024












































































Laboratorio 05

1 Tarea	3
2 Commits	4
3 Equipos y materiales utilizados	5
4 Solución	6
4.1 Código	7
4.1.1 Funcionalidad	8
4.2 Ejercicios	9
4.2.1 Ejercicio 2a	9
4.2.2 Ejercicio 2b	9
4.2.3 Ejercicio 2c	10
4.2.4 Ejercicio 2d	10
4.2.5 Ejercicio 2e	10
4.2.6 Ejercicio 2f	11
4.2.7 Ejercicio 2g	11
5 URL del repositorio en GitHub	12
6 Estructura de laboratorio 05	13
7 Rúbrica	14
8 Referencias	14

1 Tarea

- URL GitHub de Tarea del Ajedrez <https://github.com/rescobedoq/pw2/tree/main/labs/lab04/Tarea-del-Ajedrez>
- En esta tarea usted pondrá en práctica sus conocimientos de programación en Python para dibujar un tablero de Ajedrez.
- La parte gráfica ya está programada, usted sólo tendrá que concentrarse en las estructuras de datos subyacentes.
- Con el código proporcionado usted dispondrá de varios objetos de tipo `Picture` para poder realizar su tarea:
- Estos objetos estarán disponibles importando la biblioteca: `chessPictures` y estarán internamente representados con arreglos de strings que podrá revisar en el archivo `pieces.py`
- La clase `Picture` tiene un sólo atributo: el arreglo de strings `img`, el cual contendrá la representación en caracteres de la figura que se desea dibujar.
- La clase `Picture` ya cuenta con una función implementada, no debe modificarla, pero si puede usarla para implementar sus otras funciones: `_invColor`: recibe un color como un caracter de texto y devuelve su color negativo, también como texto, deberá revisar el archivo `colors.py` para conocer los valores negativos de cada caracter.
- La clase `Picture` contará además con varios métodos que usted deberá implementar:
 - `verticalMirror`: Devuelve el espejo vertical de la imagen
 - `horizontalMirror`: Devuelve el espejo horizontal de la imagen
 - `negative`: Devuelve un negativo de la imagen
 - `join`: Devuelve una nueva figura poniendo la figura del argumento al lado derecho de la figura actual
 - `up`: Devuelve una nueva figura poniendo la figura recibida como argumento, encima de la figura actual
 - `under`: Devuelve una nueva figura poniendo la figura recibida como argumento, sobre la figura actual
 - `horizontalRepeat`: Devuelve una nueva figura repitiendo la figura actual al costado la cantidad de veces que indique el valor de `n`
 - `verticalRepeat`: Devuelve una nueva figura repitiendo la figura actual debajo, la cantidad de veces que indique el valor de `n`
- Tenga en cuenta que para implementar todos estos métodos, sólo deberá trabajar sobre la representación interna de un `Picture`, es decir su atributo `img`. Para dibujar una objeto `Picture` bastará importar el método `draw` de la biblioteca `interpreter` y usarlo de la siguiente manera
- Ejercicios
 - Ejercicio 1: Implemente los métodos de la clase `Picture`.
 - Ejercicio 2: Usando únicamente los métodos de los objetos de la clase `Picture` dibuje las siguientes figuras (invoque a `draw`):

2 Commits

Ejercicio 2i completado Alanj20 committed 37 minutes ago	014547c			
Ejercicio 2e completado Alanj20 committed 40 minutes ago	f54a298			
Ejercicio 2d completado Alanj20 committed 41 minutes ago	fb01ac7			
Ejercicio 2c completado Alanj20 committed 45 minutes ago	59cd95a			
Ejercicio 2b completado Alanj20 committed 1 hour ago	b4c1cad			
Ejercicio 2a completado Alanj20 committed 1 hour ago	2b7e71a			
Solucion de pasar una lista de cadenas a caracteres. Alanj20 committed 1 hour ago	3771533			
Método rotate() y pruebas Alanj20 committed 1 hour ago	15505ff			
Método verticalRepeat() y pruebas Alanj20 committed 1 hour ago	ba7899e			
Método horizontalRepeat() y pruebas Alanj20 committed 1 hour ago	2f00f5f			
Método under() y pruebas Alanj20 committed 1 hour ago	05ba953			
Método up() y pruebas Alanj20 committed 2 hours ago	0319552			
Método join() y pruebas Alanj20 committed 2 hours ago	98a765f			
Método negative() y pruebas Alanj20 committed 2 hours ago	1dcddc6			
Método horizontalMirror() y prueba Alanj20 committed 2 hours ago	8b746a6			
Método verticalMirror() y prueba Alanj20 committed 4 hours ago	9c9f13a			
Archivo test para realizar una prueba inicial Alanj20 committed 4 hours ago	8f2dc8a			
Clonado del repositorio base Alanj20 committed 5 hours ago	fb11438			
Inicializando la tarea del lab 04: Tablero Ajedrez Alanj20 committed 5 hours ago	46deb34			
Ejercicio Prueba: Test Matriz Unitaria Alanj20 committed 5 hours ago	3b72cc3			
Ejercicio Prueba: método para la matriz si esUnitaria. Alanj20 committed 5 hours ago	5851e01			
Ejercicio Prueba: Test Matriz Escalar Alanj20 committed 5 hours ago	a141b81			
Ejercicio Prueba: método matriz escalar Alanj20 committed 5 hours ago	27c13e9			
Commits on May 29, 2024				
Lab05: Activando el entorno virtual Alanj20 committed yesterday	7eb65c7			
Lab 05: Entorno virtual creado Alanj20 committed yesterday	956d885			

Lista de commits.

3 Equipos y materiales utilizados

- Cuenta en GitHub con el correo institucional.
- Sistema Operativo Microsoft Windows 10
- Visual Studio Code
- Git
- Windows PowerShell
- Python
- Navegador Mozilla Firefox

4 Solución

- La solución de la tarea se encuentra en el directorio Tarea-del-Ajedrez, instalar primero las dependencias necesarias con el siguiente comando:
- Instalar pygame

```
PS C:\Ordenador\List\PW2\pw2-24a\lab05\Tarea-del-Ajedrez> pip install pygame
Collecting pygame
  Downloading pygame-2.5.2-cp312-cp312-win_amd64.whl.metadata (13 kB)
  Downloading pygame-2.5.2-cp312-cp312-win_amd64.whl (10.8 MB)
    10.8/10.8 MB 6.0 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.5.2
PS C:\Ordenador\List\PW2\pw2-24a\lab05\Tarea-del-Ajedrez> python -m pygame.examples.aliens
>>
pygame 2.5.2 (SDL 2.28.3, Python 3.12.1)
Hello from the pygame community. https://www.pygame.org/contribute.html
PS C:\Ordenador\List\PW2\pw2-24a\lab05\Tarea-del-Ajedrez> 
```

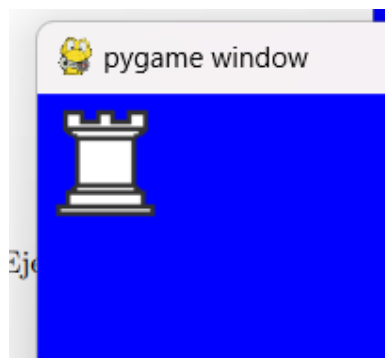
Instalar pygame

- Trabajaremos con la clase Picture y ejecutaremos a través de draw.py

```
pygame 2.5.2 (SDL 2.28.3, Python 3.12.1)
Hello from the pygame community. https://www.pygame.org/contribute.html
PS C:\Ordenador\List\PW2\pw2-24a\lab05\Tarea-del-Ajedrez> python
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from chessPictures import *
>>> from interpreter import draw
pygame 2.5.2 (SDL 2.28.3, Python 3.12.1)
Hello from the pygame community. https://www.pygame.org/contribute.html
>>> draw(rock)
>>> 
```

Ejecutar draw.py

- Prueba de ejecución



Prueba de ejecución

4.1 Código

```
1  from colors import *
2  class Picture:
3      def __init__(self, img):
4          self.img = img
5
6      def __eq__(self, other):
7          return self.img == other.img
8
9      def _invColor(self, color):
10         if color not in inverter:
11             return color
12         return inverter[color]
13
14     def verticalMirror(self):
15         img = self.img[::-1]
16         return Picture(img)
17
18     def horizontalMirror(self):
19         img = [row[::-1] for row in self.img]
20         return Picture(img)
21
22     def negative(self):
23         img = [[''.join([self._invColor(char) for char in row])] for row in self.img]
24         return Picture(img)
25
26     def join(self, other):
27         img = []
28         for i in range(len(self.img)):
29             img.append(self.img[i] + other.img[i])
30         return Picture(img)
31
32     def up(self, other):
33         img = []
34         for rowS in self.img:
35             img.append(rowS)
36         for rowO in other.img:
37             img.append(rowO)
38         return Picture(img)
39
40     def under(self, other):
41         img = []
42         for i in range(len(self.img)):
43             row = []
44             for j in range(len(self.img[i])):
45                 charS = self.img[i][j];
46                 charO = other.img[i][j];
47                 if(charS == ' '):
48                     row.append(charO)
49                 else:
50                     row.append(charS)
51             img.append(row)
52         return Picture(img)
53
54     def horizontalRepeat(self, n):
55         img = [self.img[i] * n for i in range(len(self.img))]
```

```
56     return Picture(img)
57
58 def verticalRepeat(self, n):
59     return Picture(self.img * n)
60
61 #Extra: Sólo para realmente viciosos
62 def rotate(self):
63     img = []
64     for i in range(len(self.img)):
65         row = []
66         for j in range(len(self.img[i])):
67             row.append(self.img[j][i])
68         img.append(row)
69     return Picture(img)
70
```

picture.py

4.1.1 Funcionalidad

- **init(self, img):** Este es el constructor de la clase. Inicializa una nueva instancia de Picture con una imagen img, que parece ser una lista de cadenas.
- **eq(self, other):** Este método comprueba si dos instancias de Picture son iguales comparando sus imágenes.
- **invColor(self, color):** Este método privado invierte un color si el color está en el diccionario inverter importado del módulo colors.
- **verticalMirror(self):** Este método crea una nueva imagen que es un reflejo vertical de la imagen original. Esto se logra invirtiendo el orden de las filas en img(arreglo de strings).
- **horizontalMirror(self):** Este método crea una nueva imagen que es un reflejo horizontal de la imagen original. Esto se logra invirtiendo el orden de los caracteres en cada fila de img.
- **negative(self):** Este método crea una nueva imagen que es el negativo de la imagen original. Esto se logra invirtiendo los colores de cada pixel en img.
- **join(self, other):** Este método crea una nueva imagen que es la imagen original con la imagen other a su derecha. Esto se logra concatenando las filas de img con las filas de other.img.
- **up(self, other):** Este método crea una nueva imagen que es la imagen original con la imagen other encima de ella. Esto se logra concatenando las filas de other.img con las filas de img.
- **under(self, other):** Este método crea una nueva imagen que es la imagen original con la imagen other debajo de ella. Esto se logra concatenando las filas de img con las filas de other.img.
- **horizontalRepeat(self, n):** Este método crea una nueva imagen que es la imagen original repetida n veces a su derecha. Esto se logra concatenando las filas de img con las filas de img n veces.
- **verticalRepeat(self, n):** Este método crea una nueva imagen que es la imagen original repetida n veces debajo de ella. Esto se logra concatenando las filas de img con las filas de img n veces.

4.2 Ejercicios

4.2.1 Ejercicio 2a

```
1 from interpreter import draw
2 from chessPictures import *
3 from picture import Picture
4
5 row1 = Picture(KNIGHT).join((Picture(KNIGHT)).negative())
6 draw(row1.up(row1.negative()))
```



Ejercicio2a.py

4.2.2 Ejercicio 2b

```
1 from interpreter import draw
2 from chessPictures import *
3 from picture import Picture
4
5 row1 = Picture(KNIGHT).join((Picture(KNIGHT)).negative())
6 draw(row1.up(row1.horizontalMirror()))
```



Ejercicio2b.py

4.2.3 Ejercicio 2c

```
1 from interpreter import draw
2 from chessPictures import *
3
4 row = queen.horizontalRepeat(4)
5 draw(row)
```



Ejercicio2c.py

4.2.4 Ejercicio 2d

```
1 from interpreter import draw
2 from chessPictures import *
3
4
5 tableRow = square.join(square.negative()).horizontalRepeat(4)
6 draw(tableRow)
```



Ejercicio2d.py

4.2.5 Ejercicio 2e

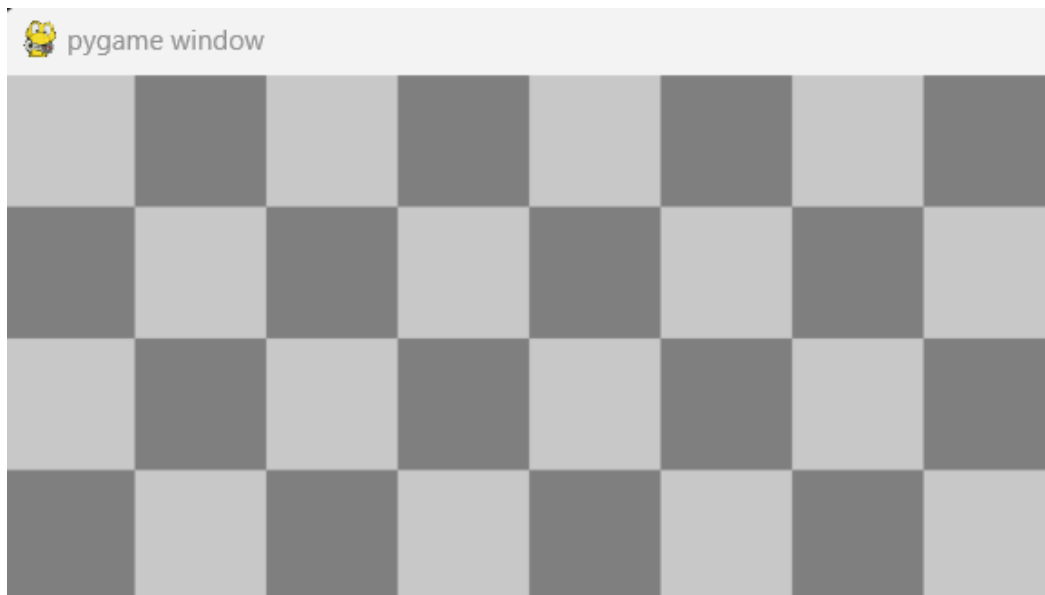
```
1 from interpreter import draw
2 from chessPictures import *
3
4
5 tableRow = square.join(square.negative()).horizontalRepeat(4)
6 draw(tableRow.negative())
```



Ejercicio2e.py

4.2.6 Ejercicio 2f

```
1 from interpreter import draw
2 from chessPictures import *
3
4
5 tableRow1 = square.join(square.negative()).horizontalRepeat(4)
6 tableRow2 = tableRow1.negative()
7 draw(tableRow1.up(tableRow2).verticalRepeat(2))
```



Ejercicio2f.py

4.2.7 Ejercicio 2g

```
1 from interpreter import draw
2 from chessPictures import *
3
4
5 tableRow = square.join(square.negative()).horizontalRepeat(4)
6 darkPieces = rock.join(knight).join(bishop).join(queen).join(king).join(bishop).join(knight).join(rock).negative()
7 darkPawn = pawn.negative().horizontalRepeat(8)
8
9
10 rowsPiecesDark = darkPieces.under(tableRow).up(darkPawn.under(tableRow.negative()))
11
12 tableSquare = tableRow.up(tableRow.negative()).verticalRepeat(2)
13
14 rowPiecesLight = darkPawn.negative().under(tableRow).up(darkPieces.negative().under(tableRow.negative()))
15
16 draw(rowsPiecesDark.up(tableSquare).up(rowPiecesLight))
```



Ejercicio2g.py

5 URL del repositorio en GitHub

- <https://github.com/Alsnj20/pw2-24a/tree/main/lab05>

6 Estructura de laboratorio 05

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab05/
|---/execises
|   |---/_pycache_
|   |---esEscalar.py
|   |---esUnitaria.py
|   |---test_esEscalar.py
|   |---test_esUnitaria.py
|---/latex
|   |--- linopinto_pw2_24a_lab05.tex
|   |--- linopinto_pw2_24a_lab05.pdf
|   |---/img
|       |---2a.png
|       |---2b.png
|       |---2c.png
|       |---2d.png
|       |---2e.png
|       |---2f.png
|       |---2g.png
|       |---commits.png
|       |---draw.png
|       |---drawPrueba.png
|       |---episunsa.png
|       |---logo_abet.png
|       |---pygame.png
|---/my_env
|---/Tarea1-del-Ajedrez
|   |---/_pycache__
|   |---.gitignore
|   |---chessPicture.py
|   |---colors.py
|   |---Ejercicio2a.py
|   |---Ejercicio2b.py
|   |---Ejercicio2c.py
|   |---Ejercicio2d.py
|   |---Ejercicio2e.py
|   |---Ejercicio2f.py
|   |---Ejercicio2g.py
|   |---interpreter.py
|   |---picture.py
|   |---pieces.py
|   |---test.py
|--- README.md
|---.gitignore
```

7 Rúbrica

Tabla: Rúbrica para contenido del Informe y evidencias

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o observación)	4	×	4	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación)	4	×	4	
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4	×	4	
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación)	2	×	2	
7.Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2	×	1	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación)	4	×	4	
Total		20	Completo	19	

8 Referencias

- <https://github.com/>
- <https://git-scm.com/>
- <https://www.w3schools.com/python/>