

Informe de Laboratorio 05

Tema: Laboratorio 05

Nota

Estudiante	Escuela	Asignatura
Jhonatan David Arias Quispe jariasq@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: II Código: 1702122

Laboratorio	Tema	Duración
05	Laboratorio 05	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 29 Mayo 2024	Al 1 Junio 2024

1. Equipos, materiales y temas utilizados

- Sistema Operativo ArchCraft GNU Linux 64 bits Kernell
- NeoVim
- Git 2.42.0
- Cuenta en GitHub con el correo institucional.
- Python
- Latex

2. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/JhonatanDczel/pweb2.git>
- URL para el laboratorio 05 en el Repositorio GitHub.
- <https://github.com/JhonatanDczel/pweb2/tree/main/lab05>

3. Informe sobre la clase Picture

La clase `Picture` en el código presentado está diseñada para manejar y manipular imágenes representadas como listas de cadenas de caracteres. Esta clase ofrece diversas funcionalidades para transformar y combinar imágenes de diferentes maneras. A continuación, se presenta un análisis detallado de sus métodos y atributos.

3.1. Atributos

- `img`: Este atributo almacena la imagen, que es una lista de cadenas de caracteres donde cada cadena representa una fila de la imagen.

3.2. Métodos

1. Constructor (`__init__`)

```
1 def __init__(self, img):
2     self.img = img
```

2. Igualdad (`__eq__`)

```
1 def __eq__(self, other):
2     return self.img == other.img
```

3. Invertir Color (`_invColor`)

```
1 def _invColor(self, color):
2     if color not in inverter:
3         return color
4     return inverter[color]
```

4. Espejo Vertical (`verticalMirror`)

```
1 def verticalMirror(self):
2     vertical = []
3     for fila in self.img:
4         espejo = ""
5         for letra in fila:
6             espejo = letra + espejo
7         vertical.append(espejo)
8     return Picture(vertical)
```

5. Espejo Horizontal (`horizontalMirror`)

```
1 def horizontalMirror(self):
2     horizontal = []
3     for i in range(len(self.img)):
4         l = len(self.img) - i - 1
5         horizontal.append(self.img[l])
6     return Picture(horizontal)
```

6. Negativo (`negative`)

```
1 def negative(self):
2     new_img = []
```

```
3     for string in self.img:
4         new_string = [self._invColor(letter) for letter in string]
5         new_img.append(new_string)
6     return Picture(new_img)
```

7. Unir (join)

```
1     def join(self, p):
2         new_img = []
3         l = max(len(self.img), len(p.img))
4         for i in range(l):
5             izquierda = self.img[i] if i < len(self.img) else [" " for _ in
6                 range(len(self.img[0]))]
7             derecha = p.img[i] if i < len(p.img) else [" " for _ in range(len(p.img[0]))]
8             row = "".join(izquierda) + "".join(derecha)
9             new_img.append(row)
10        return Picture(new_img)
```

8. Colocar Arriba (up)

```
1     def up(self, p):
2         new_img = self.img + p.img
3         return Picture(new_img)
```

9. Colocar Debajo (under)

```
1     def under(self, p):
2         new_img = p.img + self.img
3         return Picture(new_img)
```

10. Repetición Horizontal (horizontalRepeat)

```
1     def horizontalRepeat(self, n):
2         new_img = []
3         for row in self.img:
4             new_row = row * n
5             new_img.append(new_row)
6         return Picture(new_img)
```

11. Repetición Vertical (verticalRepeat)

```
1     def verticalRepeat(self, n):
2         new_img = self.img * n
3         return Picture(new_img)
```

12. Rotar en Sentido Horario (rotate_horario)

```
1     def rotate_horario(self):
2         transpuesta = list(zip(*self.img))
3         rotada = [list(fila)[-1] for fila in transpuesta]
4         return Picture(rotada)
```

13. Rotar en Sentido Antihorario (rotate_antihorario)

```
1     def rotate_antihorario(self):
2         return self.rotate_horario().rotate_horario().rotate_horario()
```

14. Superposición (on)

```
1 def on(self, p):
2     new_img = []
3     for i in range(len(self.img)):
4         row = self.img[i]
5         p_row = p.img[i]
6         new_row = []
7         for j in range(len(row)):
8             if row[j] == " ":
9                 new_row.append(p_row[j])
10            else:
11                new_row.append(row[j])
12        new_img.append(new_row)
13    return Picture(new_img)
```

15. Métodos Estáticos para Crear Piezas

```
1 @staticmethod
2 def rock():
3     return Picture(ROCK)
4
5 @staticmethod
6 def king():
7     return Picture(KING)
8
9 @staticmethod
10 def bishop():
11     return Picture(BISHOP)
12
13 @staticmethod
14 def square():
15     return Picture(SQUARE)
16
17 @staticmethod
18 def knight():
19     return Picture(KNIGHT)
20
21 @staticmethod
22 def pawn():
23     return Picture(PAWN)
24
25 @staticmethod
26 def queen():
27     return Picture(QUEEN)
```

4. Ejercicios con la biblioteca

Las figuras que se nos pidió hacer son:

5. Pregunta

El directorio pycache es para guardar las instancias de las clases para c posterior consulta

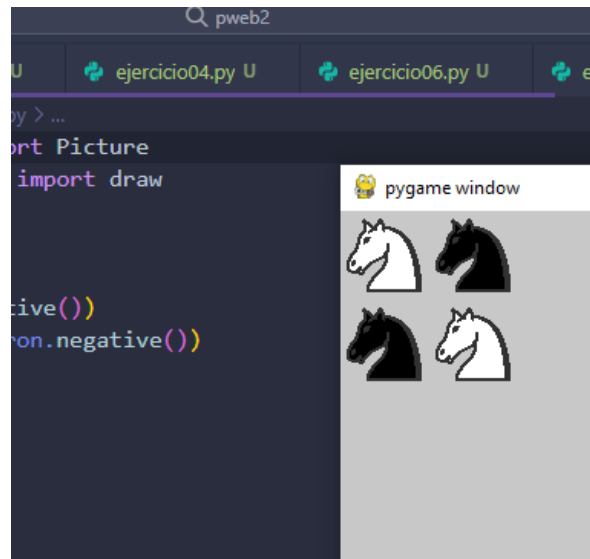


Figura 1: 1

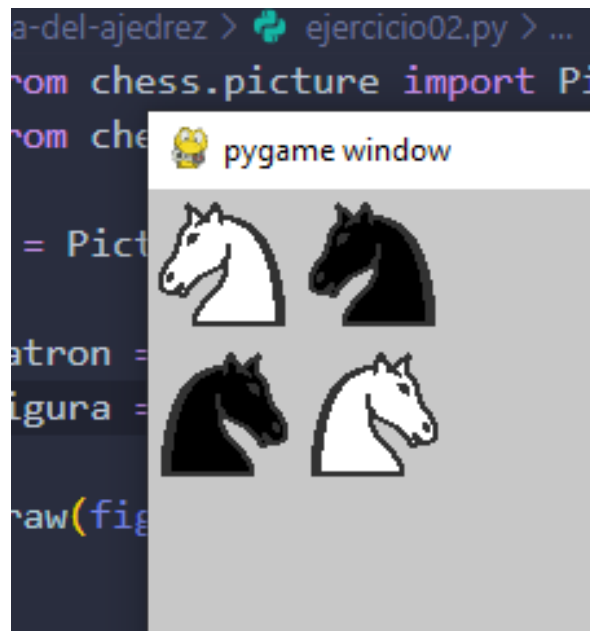


Figura 2: 2

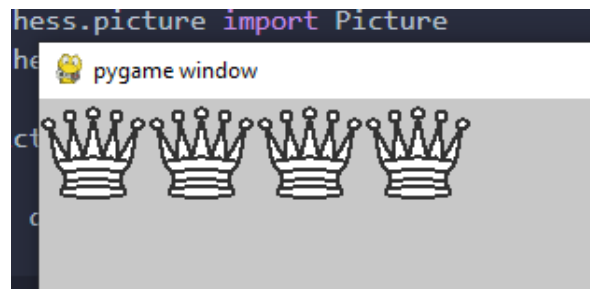


Figura 3: 3

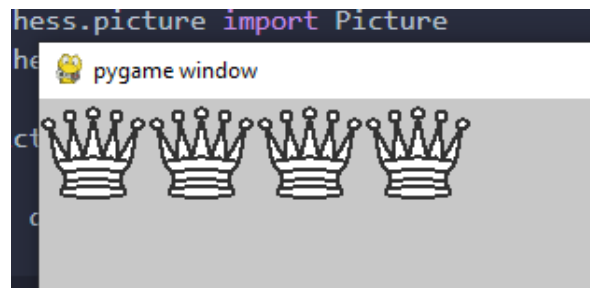


Figura 4: 4

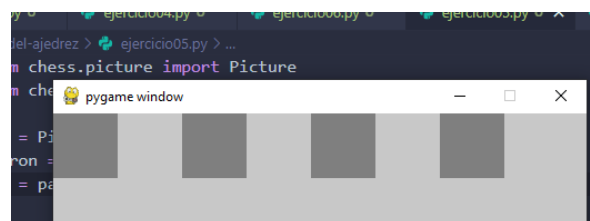


Figura 5: 5

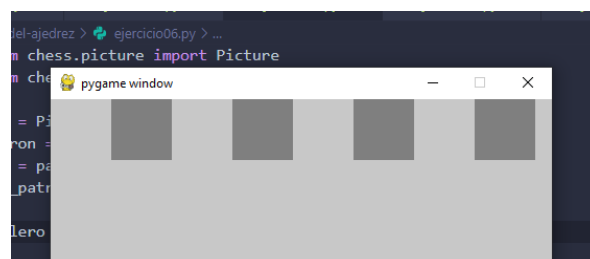


Figura 6: 6



Figura 7: 7

6. Rúbricas

6.1. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 2: Rúbrica para contenido del Informe y demostración

	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	1.5	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
Total		20		18.5	