

# Informe de Laboratorio 05

## Tema: Python

Nota

Estudiante	Escuela	Asignatura
Hernan Andy Choquehuanca Zapana hchoquehuancaz@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 1702122

Laboratorio	Tema	Duración
05	Python	4 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 20 Mayo 2024	Al 25 Mayo 2024

### 1. Tarea

- La clase Picture contará además con varios métodos que usted deberá implementar:
- 1. verticalMirror: Devuelve el espejo vertical de la imagen
- 2. horizontalMirror: Devuelve el espejo horizontal de la imagen
- 3. negative: Devuelve un negativo de la imagen
- 4. join: Devuelve una nueva figura poniendo la figura del argumento al lado derecho de la figura actual
- 5. up: Devuelve una nueva figura poniendo la figura recibida como argumento, encima de la figura actual
- 6. under: Devuelve una nueva figura poniendo la figura recibida como argumento, sobre la figura actual
- 7. horizontalRepeat, Devuelve una nueva figura repitiendo la figura actual al costado la cantidad de veces que indique el valor de n
- 8. 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 resolver los siguientes ejercicios sólo está permitido usar ciclos, condicionales, definición de listas por comprensión, sublistas, map, join, (+), lambda, zip, append, pop, range.
- 1. Implemente los métodos de la clase Picture. Se recomienda que implemente la clase picture por etapas, probando realizar los dibujos que se muestran en la siguiente preguntas.
- 2. Usando únicamente los métodos de los objetos de la clase Picture dibuje las siguientes figuras (invoque a draw):

## 2. Entregables

- Informe de laboratorio

## 3. Equipos, materiales y temas utilizados

- Sistema Operativo Windows 11 Home Single Language 22H2 64 bits.
- Git 2.41.0.2.
- Python

## 4. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.  
`https://github.com/hz07s/pw2-lab05.git`
- URL del video solicitado.  
El video se encuentra subido en el Flip indicado

## 5. Desarrollo del laboratorio 05

5.1. Implemente los métodos de la clase Picture. Se recomienda que implemente la clase picture por etapas, probando realizar los dibujos que se muestran en la siguiente preguntas.

### 5.1.1. verticalMirror

```
14 def verticalMirror(self):
15     """ Devuelve el espejo vertical de la imagen """
16     vertical = []
17     for value in self.img:
18         vertical.append(value[::-1])
19     return Picture(vertical)
```

- La función verticalMirror es un método de una clase que devuelve el espejo vertical de una imagen.
- Se crea una lista vacía llamada vertical.
- Se itera a través de los valores de la imagen (self.img).
- Cada valor se invierte utilizando value[::-1] y se agrega a la lista vertical.
- Finalmente, se devuelve una nueva instancia de la clase Picture con la imagen reflejada verticalmente.

### 5.1.2. horizontalMirror

```
21 def horizontalMirror(self):
22     """ Devuelve el espejo horizontal de la imagen """
23     new_img = self.img[::-1]
24     return Picture(new_img)
```

- La función 'horizontalMirror' es un método de una clase que devuelve el espejo horizontal de una imagen.
- Se crea una nueva lista 'new\_img' invirtiendo el orden de los elementos de 'self.img' utilizando 'self.img[::-1]'.
- Se devuelve una nueva instancia de la clase 'Picture' con la imagen reflejada horizontalmente.

### 5.1.3. negative

```
26 def negative(self):
27     """ Devuelve un negativo de la imagen """
28     new_img = []
29     for line in self.img:
30         new_img.append("".join(self._invColor(char) for char in line))
31     return Picture(new_img)
```

- La función ‘negative’ es un método de una clase que devuelve un negativo de una imagen.
- Se crea una lista vacía llamada ‘new\_img’.
- Se itera a través de cada línea de la imagen (‘self.img’).
- Para cada línea, se itera a través de cada carácter y se aplica la función ‘\_invColor’ para invertir el color.
- Los caracteres con color invertido se unen en una cadena utilizando ‘.join()’ y se agregan a ‘new\_img’.
- Finalmente, se devuelve una nueva instancia de la clase ‘Picture’ con la imagen en negativo.

### 5.1.4. join

```
33 def join(self, p):
34     """ Devuelve una nueva figura poniendo la figura del argumento
35         al lado derecho de la figura actual """
36     new_img = [self_line + p_line for self_line, p_line in zip(self.img, p.img)]
37     return Picture(new_img)
```

- La función ‘join’ es un método de una clase que devuelve una nueva figura colocando la figura del argumento al lado derecho de la figura actual.
- Se crea una nueva lista ‘new\_img’ utilizando una list comprehension que concatena cada línea de la imagen actual (‘self.img’) con la línea correspondiente de la imagen del argumento (‘p.img’).
- Se utiliza la función ‘zip’ para iterar simultáneamente sobre las líneas de ambas imágenes.
- Finalmente, se devuelve una nueva instancia de la clase ‘Picture’ con la imagen resultante de unir las dos figuras.

### 5.1.5. up

```
39 def up(self, p):
40     """ Devuelve una nueva figura poniendo la figura recibida como argumento,
41         encima de la figura actual """
42     new_img = p.img + self.img
43     return Picture(new_img)
```

- La función up es un método de una clase que devuelve una nueva figura poniendo la figura recibida como argumento encima de la figura actual.
- Se crea una nueva lista new\_img concatenando las líneas de la imagen del argumento (p.img) con las líneas de la imagen actual (self.img).
- Finalmente, se devuelve una nueva instancia de la clase Picture con la imagen resultante de colocar la figura del argumento encima de la figura actual.

### 5.1.6. under

```
45 def under(self, p):
46     """ Devuelve una nueva figura poniendo la figura p sobre la
47         figura actual """
48     new_img = [''.join([self_char if p_char == ' ' else p_char for self_char, p_char in
49                        zip(self_line, p_line)]) for self_line, p_line in zip(self.img, p.img)]
49     return Picture(new_img)
```

- La función under es un método de una clase que devuelve una nueva figura poniendo la figura p sobre la figura actual.
- Se crea una nueva lista new\_img utilizando una list comprehension que itera sobre las líneas de ambas imágenes (self.img y p.img).
- Para cada par de líneas, se crea una nueva línea donde se reemplaza cada carácter de la imagen actual por el carácter correspondiente de la imagen p si este no es un espacio.
- Finalmente, se devuelve una nueva instancia de la clase Picture con la imagen resultante de colocar la figura p sobre la figura actual.

### 5.1.7. horizontalRepeat

```
51 def horizontalRepeat(self, n):
52     """ Devuelve una nueva figura repitiendo la figura actual al costado
53         la cantidad de veces que indique el valor de n """
54     if n < 1:
55         return Picture(self.img)
56     new_img = [line * n for line in self.img]
57     return Picture(new_img)
```

- La función horizontalRepeat es un método de una clase que devuelve una nueva figura repitiendo la figura actual al costado la cantidad de veces que indique el valor de n.
- Si n es menor que 1, se devuelve la imagen actual sin modificaciones.
- De lo contrario, se crea una nueva lista new\_img donde cada línea de la imagen actual (self.img) se repite n veces.
- Finalmente, se devuelve una nueva instancia de la clase Picture con la imagen resultante de repetir horizontalmente la figura actual.

### 5.1.8. verticalRepeat

```
59 def verticalRepeat(self, n):
60     """ Devuelve una nueva figura repitiendo la figura actual debajo,
61         la cantidad de veces que indique el valor de n """
62     if n < 1:
63         return Picture(self.img)
64     new_img = [line for _ in range(n) for line in self.img]
65     return Picture(new_img)
```

- La función verticalRepeat es un método de una clase que devuelve una nueva figura repitiendo la figura actual debajo, la cantidad de veces que indique el valor de n.
- Si n es menor que 1, se devuelve la imagen actual sin modificaciones.
- De lo contrario, se crea una nueva lista new\_img donde se repiten las líneas de la imagen actual (self.img) n veces.
- Finalmente, se devuelve una nueva instancia de la clase Picture con la imagen resultante de repetir verticalmente la figura actual.

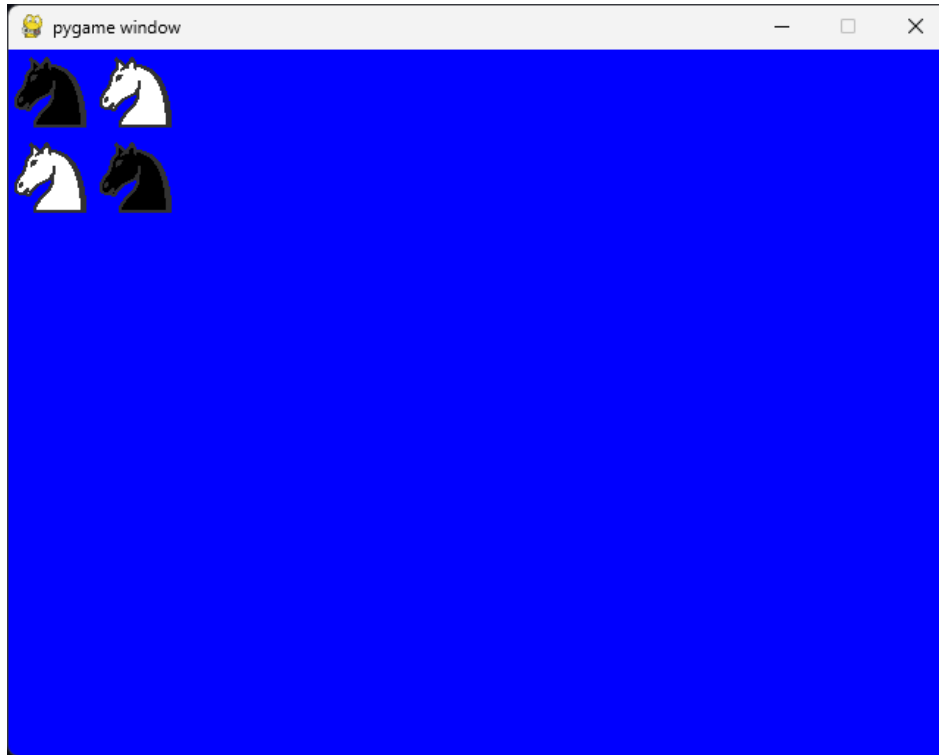
### 5.1.9. rotate

```
68 def rotate(self):
69     """ Devuelve una figura rotada en 90 grados, puede ser en sentido horario
70         o antihorario """
71     ll = len(self.img[0])
72     ls = len(self.img)
73     new_img = ["".join([self.img[j][i] for j in range(ls-1, -1, -1)]) for i in range(ll)]
74     return Picture(new_img)
```

- La función rotate es un método de una clase que devuelve una figura rotada en 90 grados, en sentido horario.
- Se obtienen las dimensiones de la imagen actual (ll es el ancho y ls es el alto).
- Se crea una nueva lista new\_img utilizando una list comprehension que itera sobre las columnas de la imagen original.
- Para cada columna, se crea una nueva línea concatenando los caracteres de la columna en orden inverso (de abajo hacia arriba).
- Finalmente, se devuelve una nueva instancia de la clase Picture con la imagen resultante de rotar la figura actual 90 grados.

## 5.2. Usando únicamente los métodos de los objetos de la clase Picture dibuje las siguientes figuras (invoque a draw):

### 5.2.1. Ejercicio 1:

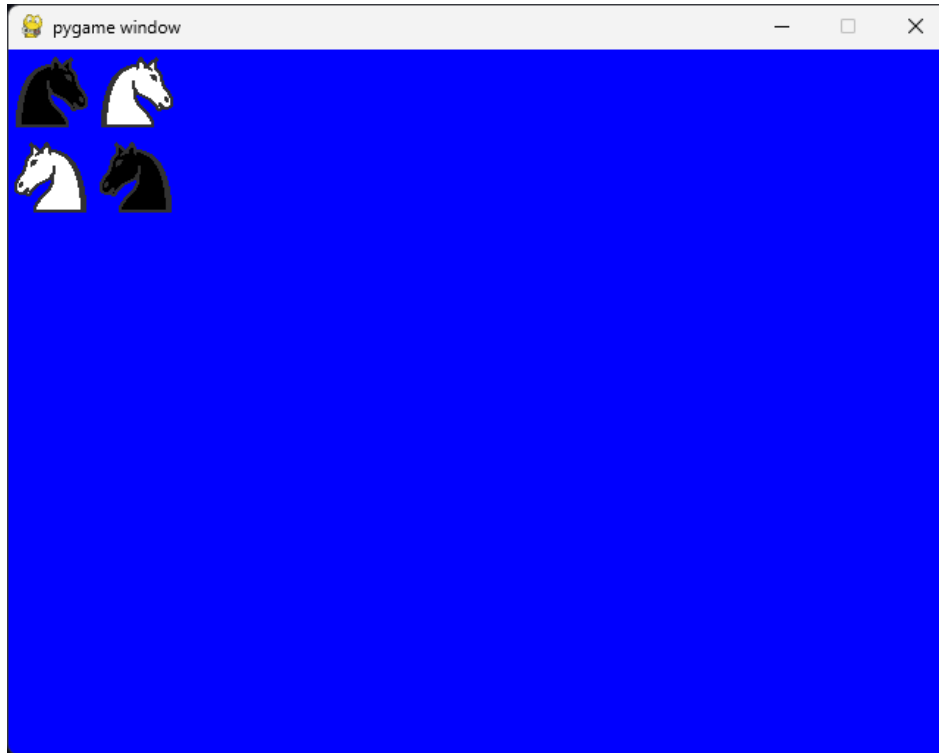


```
1 from interpreter import draw
2 from chessPictures import *
3
4 g1 = knight.join(knight.negative())
5 g2 = knight.negative().join(knight)
6
7 draw(g1.up(g2))
```

- Se crea una figura g1 uniendo horizontalmente un caballo y su negativo.
- Se crea una figura g2 reflejando verticalmente g1.
- Se dibuja g1 encima de g2.



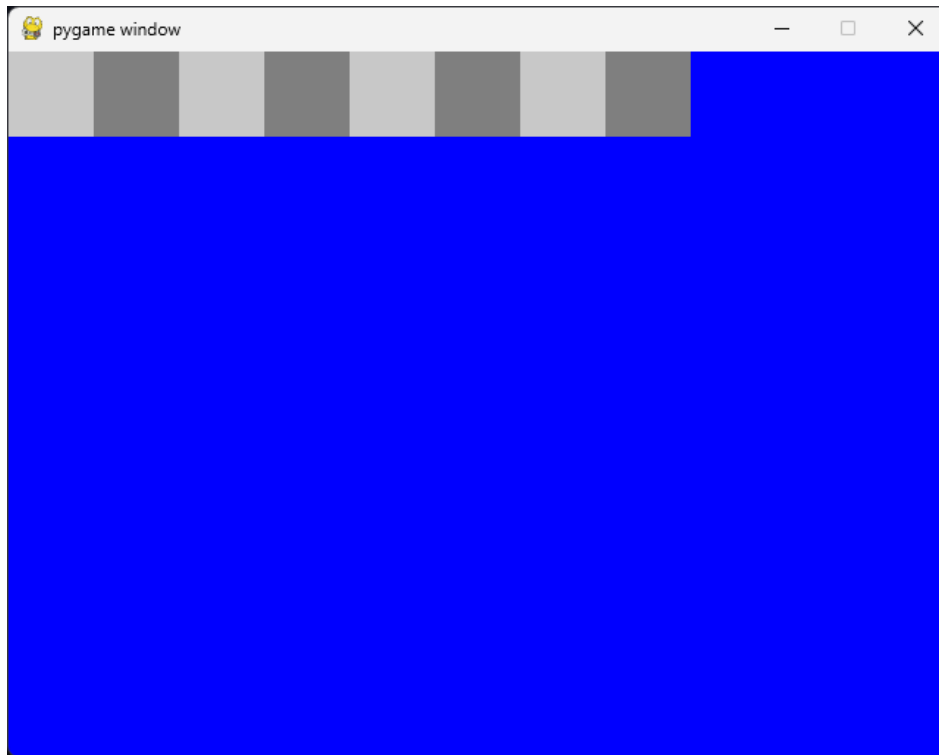
### 5.2.2. Ejercicio 2:



```
1 from interpreter import draw
2 from chessPictures import *
3
4 g1 = knight.join(knight.negative())
5 g2 = g1.verticalMirror()
6
7 draw(g1.up(g2))
```

- Se crea una figura g1 uniendo horizontalmente un caballo y su negativo.
- Se crea una figura g2 reflejando verticalmente g1.
- Se dibuja g1 encima de g2.

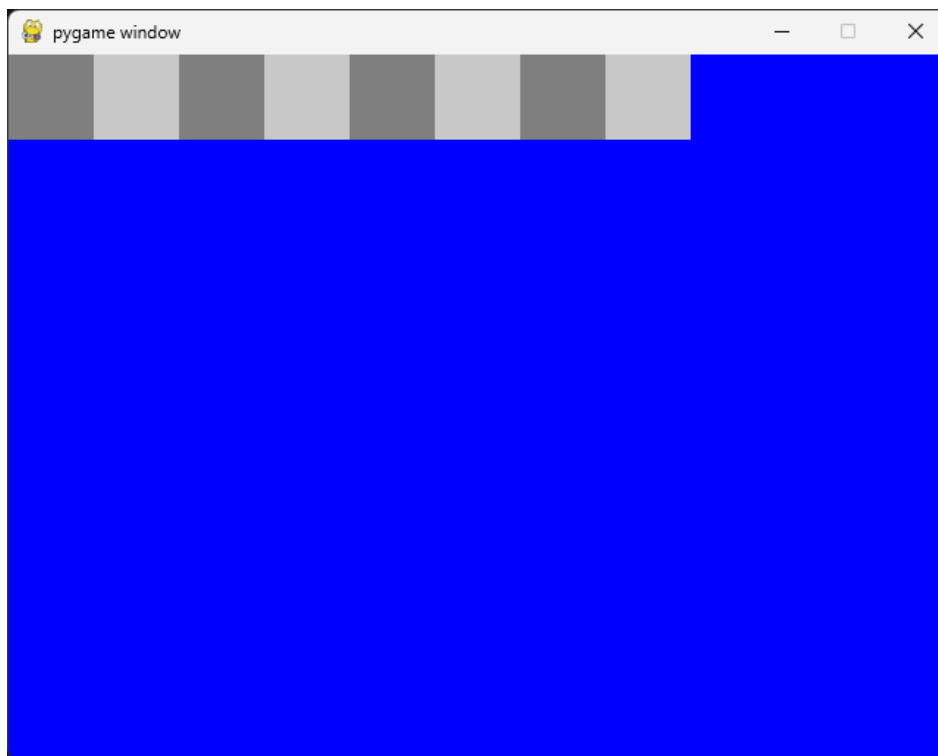
### 5.2.3. Ejercicio 3:



```
1 from interpreter import draw
2 from chessPictures import *
3
4 g1 = square.join(square.negative())
5
6 draw(g1.horizontalRepeat(4))
```

- Se crea una figura g1 uniendo horizontalmente un cuadrado y su negativo.
- Se repite g1 horizontalmente 4 veces.
- Se dibuja el resultado.

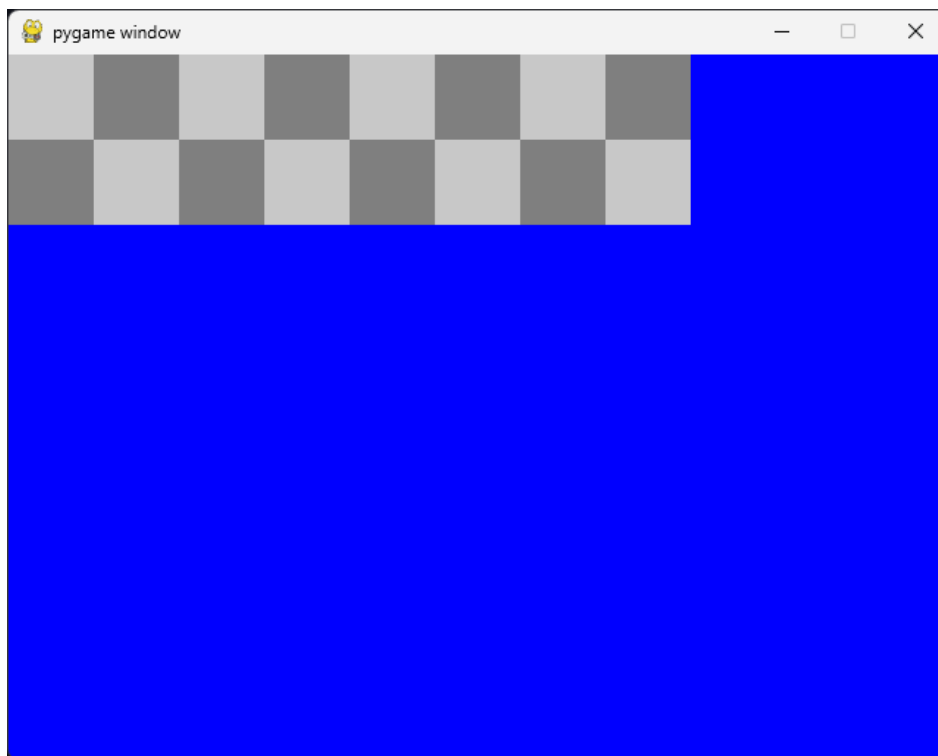
#### 5.2.4. Ejercicio 4:



```
1 from interpreter import draw
2 from chessPictures import *
3
4 g1 = square.negative().join(square)
5
6 draw(g1.horizontalRepeat(4))
```

- Se crea una figura g1 uniendo horizontalmente un cuadrado y su negativo.
- Se repite g1 horizontalmente 4 veces.
- Se dibuja el resultado.

### 5.2.5. Ejercicio 5:



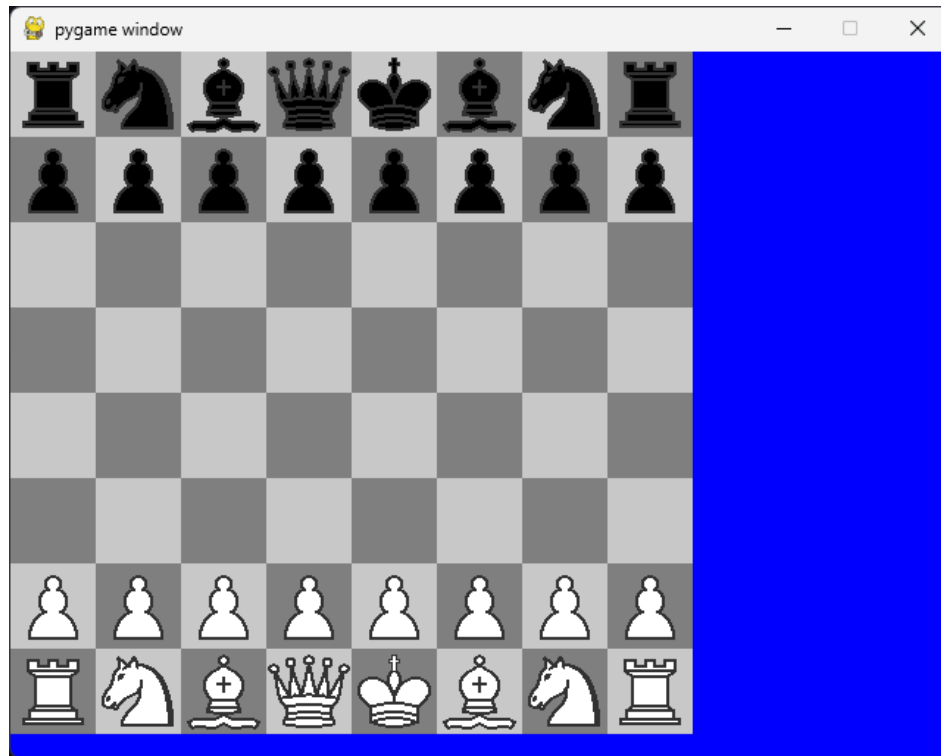
```

1 from interpreter import draw
2 from chessPictures import *
3
4 g1 = square.join(square.negative()).horizontalRepeat(4)
5 g2 = g1.negative()
6 g3 = g1.under(g2)
7
8 draw(g3.up(g3.negative().horizontalMirror()))

```

- Se crea una figura g1 uniendo horizontalmente un cuadrado y su negativo, y repitiendo el resultado 4 veces.
- Se crea una figura g2 que es el negativo de g1.
- Se crea una figura g3 colocando g2 sobre g1.
- Se dibuja g3 encima de su negativo reflejado horizontalmente.

### 5.2.6. Ejercicio 6:



```

1 from interpreter import draw
2 from chessPictures import *
3
4 s = square
5 sn = square.negative()
6 g_left = sn.under(rock).join(s.under(knight)).join(sn.under(bishop))
7 g_right = s.under(bishop).join(sn.under(knight)).join(s.under(rock))
8 g_mid = s.under(queen).join(sn.under(king))
9 g_pawn = s.under(pawn).join(sn.under(pawn)).horizontalRepeat(4)
10 g_square1 = sn.join(s).horizontalRepeat(4)
11 g_square2 = g_square1.negative()
12 g_square = g_square1.up(g_square2).verticalRepeat(2)
13 g_white = g_left.join(g_mid.join(g_right)).up(g_pawn)
14 g_black = g_pawn.up(g_left.join(g_mid.join(g_right))).negative()
15
16 draw(g_white.up(g_square.up(g_black)))

```

- Se crean varias figuras para representar las piezas de ajedrez y los cuadrados del tablero.
- Se crea una figura g.white con las piezas blancas y los peones.
- Se crea una figura g.black con las piezas negras y los peones.
- Se crea una figura g.square con los cuadrados del tablero.
- Se dibuja g.white encima de g.square encima de g.black.

### 5.3. 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.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- 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
<b>1. GitHub</b>	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
<b>2. Commits</b>	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
<b>3. Código fuente</b>	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
<b>4. Ejecución</b>	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
<b>5. Pregunta</b>	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
<b>6. Fechas</b>	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
<b>7. Ortografía</b>	El documento no muestra errores ortográficos.	2	X	2	
<b>8. Madurez</b>	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	4	
<b>Total</b>		20		20	

## 6. Referencias

- <https://docs.python.org/3/>
- <https://www.pygame.org/docs/>