

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	PROGRAMACIÓN WEB 2				
TÍTULO DE LA PRÁCTICA:	TABLERO				
NÚMERO DE PRÁCTICA:	4	AÑO LECTIVO:	2024	NRO. SEMESTRE:	II
FECHA DE PRESENTACIÓN	19/10/2024	HORA DE PRESENTACIÓN	23/21/00		
INTEGRANTE (s)  Marron Puma, Maritza Claudia				NOTA (0-20)	
DOCENTE(s): Lino José Pinto Oppe					

LabOficial\_04

LabOficial\_04/LaboratorioIVOficial\_Marron\_Puma\_Maritza  
[https://github.com/LINOPINTO2023/PWII-2024B/tree/89f05e7b76c61443ddfc37f09857e4535542591f/LabOficial\\_04](https://github.com/LINOPINTO2023/PWII-2024B/tree/89f05e7b76c61443ddfc37f09857e4535542591f/LabOficial_04)

Commits

History for PWII-2024B / LabOficial\_04 / LaboratorioIVOficial\_Marron\_Puma\_Maritza on Grupo-H1au

🔍 All users All time

⏪ Commits on Oct 18, 2024

AGREGANDO LOS EJERCICIOS PRPUESTOS

maritza1818 committed 27 minutes ago 89f85e7

⏪ Commits on Oct 17, 2024

Inserseccion de los archivos del tablero para poder trabajr en el ejercio propuesto

maritza1818 committed yesterday 0184966

Verificacion de si una matriz es escalar o no ademas se implemento comentarios que indiquen los casos como matrices escalares y no escalares como casos de prueba

maritza1818 committed yesterday b1073b0

Actualizacion de pip 24.2 ademas de la instalación de la libreria numpy

maritza1818 committed yesterday 539f90a

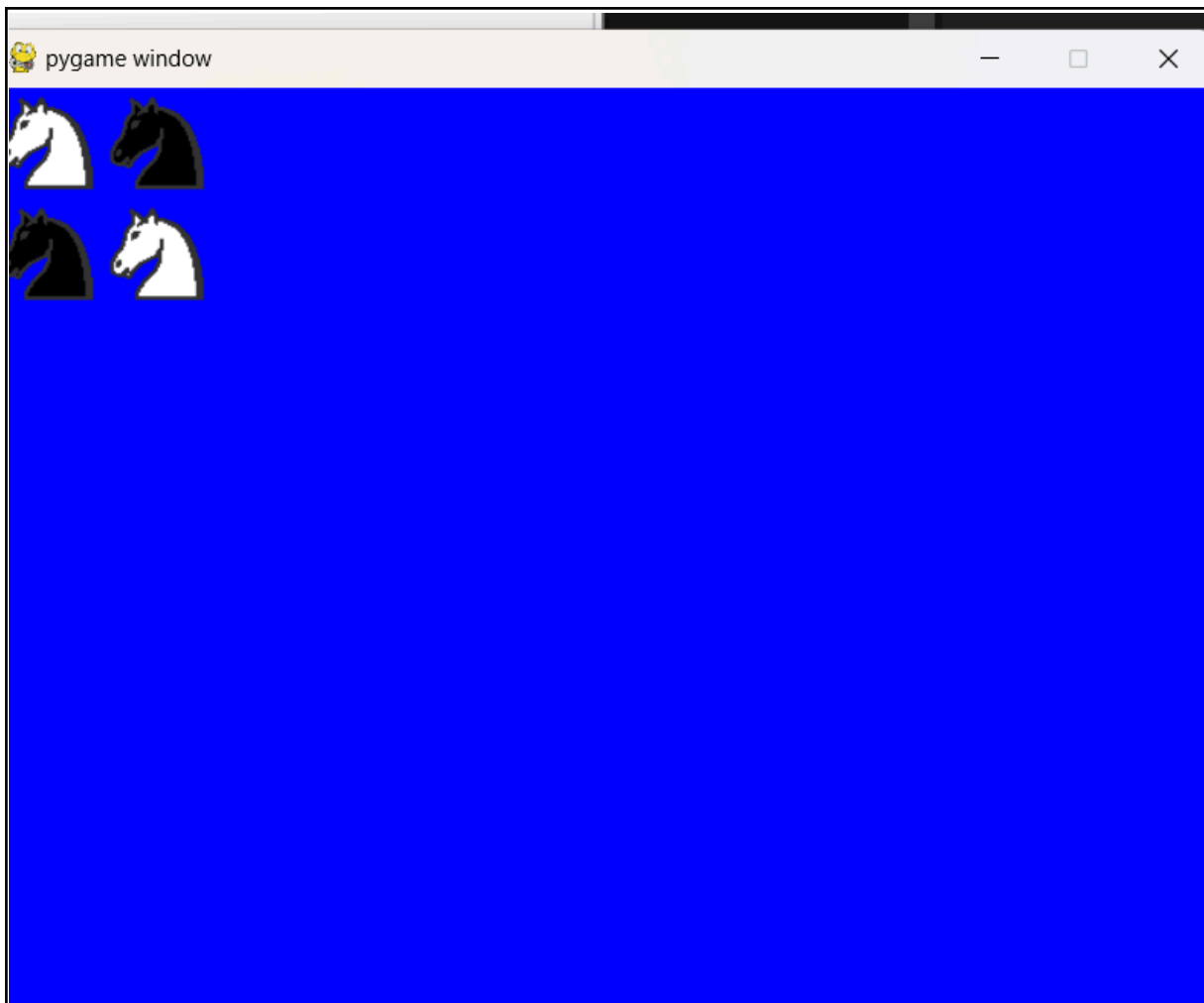
⏪ End of commit history for this file

ejecuciones/pruebas del código fuente explicadas gradualmente.

## RESULTADOS Y PRUEBAS

El informe es la corrección de un sistema basado en Python para generar tableros de ajedrez y manipular imágenes de figuras como caballos, reinas, y peones. El sistema hace uso de las funciones de las clases proporcionadas para manipular, unir, y reflejar imágenes, permitiendo la creación de diversas configuraciones de tableros de ajedrez, tanto de figuras como de casillas. Además, se describe cómo las figuras se posicionan en el tablero utilizando las herramientas disponibles para crear diferentes patrones y diseños.

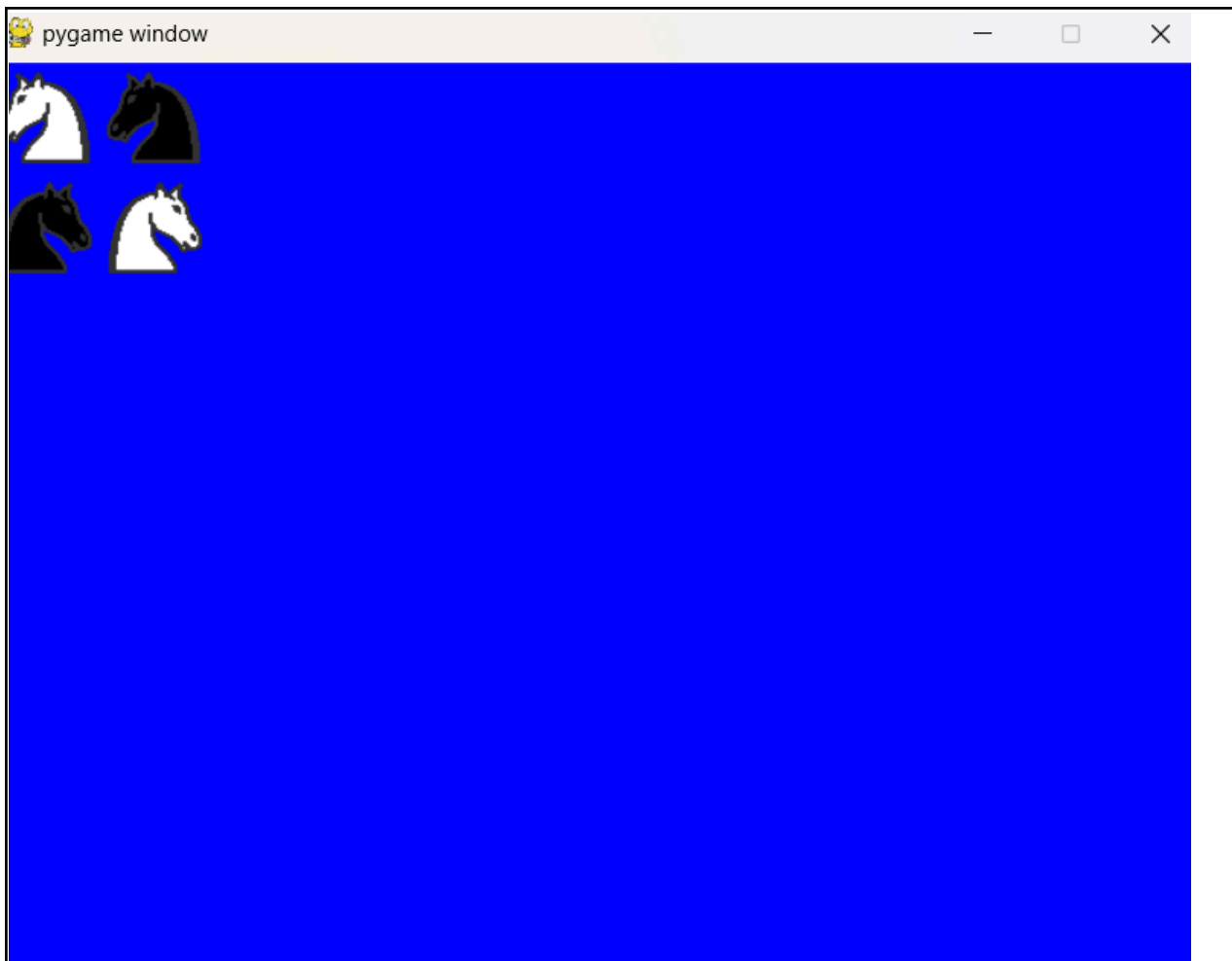
### **EJERCICIO2A: Manipulación de Caballos**



```
Ejercicio2a.py X  colors.py  Ejercicio2b.py  Ejercicio2c.py  Ejercicio2a.py > ...
LaboratorioIVOficial_Marron_Puma_Maritza > EjerciciosPropuestos > Archivos_Tablero > Ejercicio2a.py > ...
1  from interpreter import draw
2  from chessPictures import *
3
4  row_1 = knight.join(knight.negative())
5  row_2 = row_1.negative()
6  board = row_1.under(row_2)
7  draw(board)
```

En el ejercicio 2a se genera un tablero alternando un caballo blanco con uno negro. Se utiliza el método *negative()* para invertir los colores del caballo blanco, obteniendo así un caballo negro. Luego, se crea la fila de caballos utilizando el método *join()* y se invierte verticalmente con *negative()*. Finalmente, se une la fila de caballos con la fila invertida para formar el tablero.

### EJERCICO 2B: Reflejo Vertical



```
Ejercicio2a.py | colors.py | Ejercicio2b.py X | Ejercicio2c.py | Ejecutar | ...
LaboratorioIVOficial_Marron_Puma_Maritz > EjerciciosPropuestos > Archivos_Tablero > Ejercicio2b.py > ...
1  from interpreter import draw
2  from chessPictures import *
3
4  row_1 = knight.join(knight.negative())
5  row_2 = row_1.verticalMirror()
6  board = row_1.under(row_2)
7  draw(board)
```

Aquí se genera un tablero similar al anterior, pero con un reflejo vertical de la fila de caballos. El método **verticalMirror()** se aplica a **row\_1**, reflejando la imagen sobre el eje vertical. Luego, se combinan ambas filas para formar el tablero.

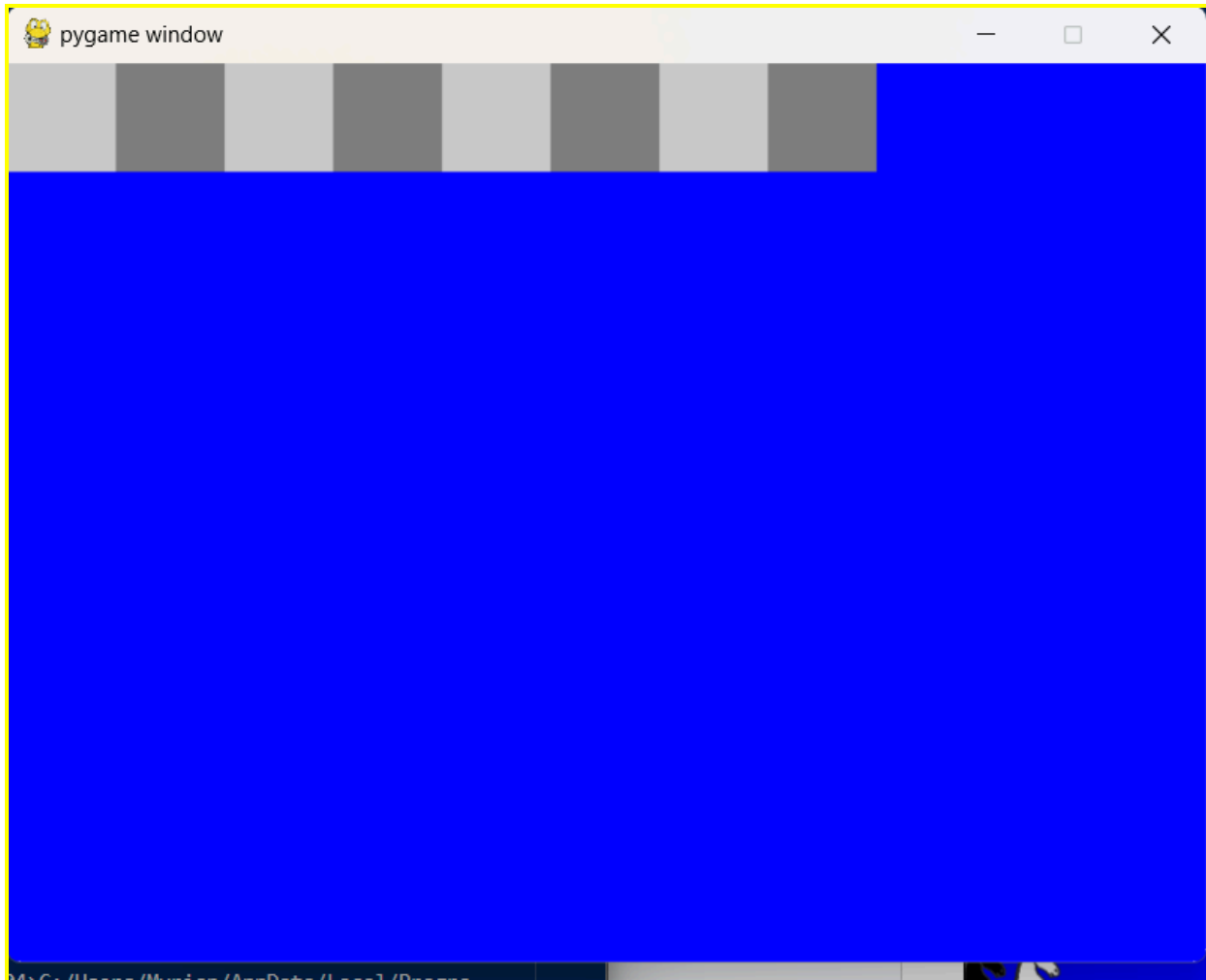
### EJERCICIO 2C: Repetición de Reinas Horizontalmente



```
Ejercicio2a.py | colors.py | Ejercicio2b.py | Ejercicio2c.py X | Ejercicio2c.py
LaboratorioIVOficial_Marron_Puma_Maritz > EjerciciosPropuestos > Archivos_Tablero > Ejercicio2c.py
1  from interpreter import draw
2  from chessPictures import *
3
4  draw(queen.horizontalRepeat(4))
```

Se toma una figura de reina y se repite horizontalmente cuatro veces utilizando el método *horizontalRepeat()*. El resultado es una fila de cuatro reinas, mostrando cómo se pueden manipular las figuras para generar patrones repetitivos.

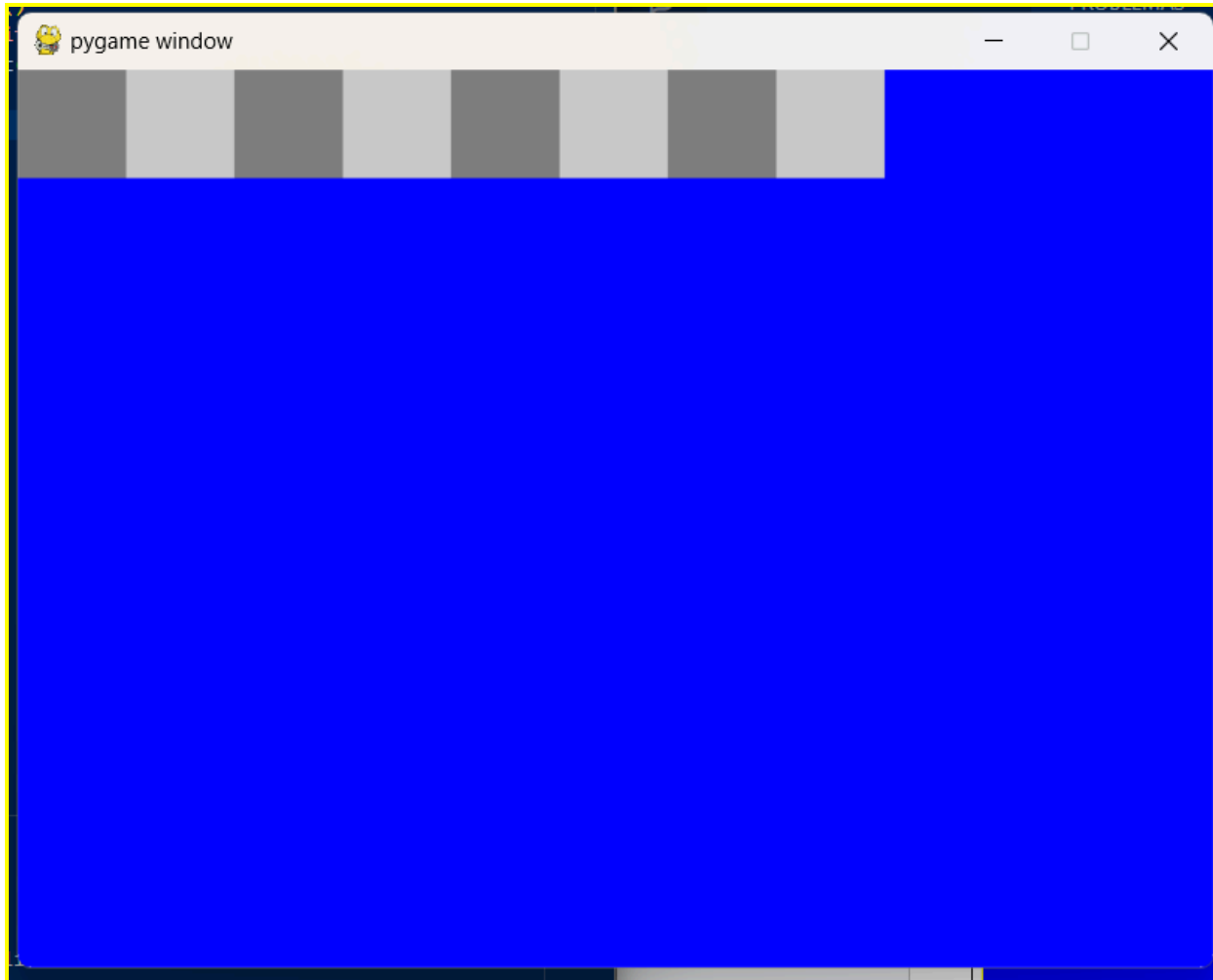
### ***EJERCICIO2D: Tablero de Cuadrados Alternos***



```
py | colors.py | Ejercicio2b.py | Ejercicio2c.py | Ejercicio2d.py X | ▸ ▾ □ ·
LaboratorioIVOOficial_Marron_Puma_Maritza > EjerciciosPropuestos > Archivos_Tablero > Ejercicio2d.py > ...
1  from interpreter import draw
2  from chessPictures import *
3
4  row_1 = square.join(square.negative()).horizontalRepeat(4)
5  board = row_1
6  draw(board)
```

Generó una fila alternando cuadrados blancos y negros, formando un patrón que se repite horizontalmente, utilizando *join()* para combinar un cuadrado y su negativo, y *horizontalRepeat()* para crear múltiples repeticiones de la fila.

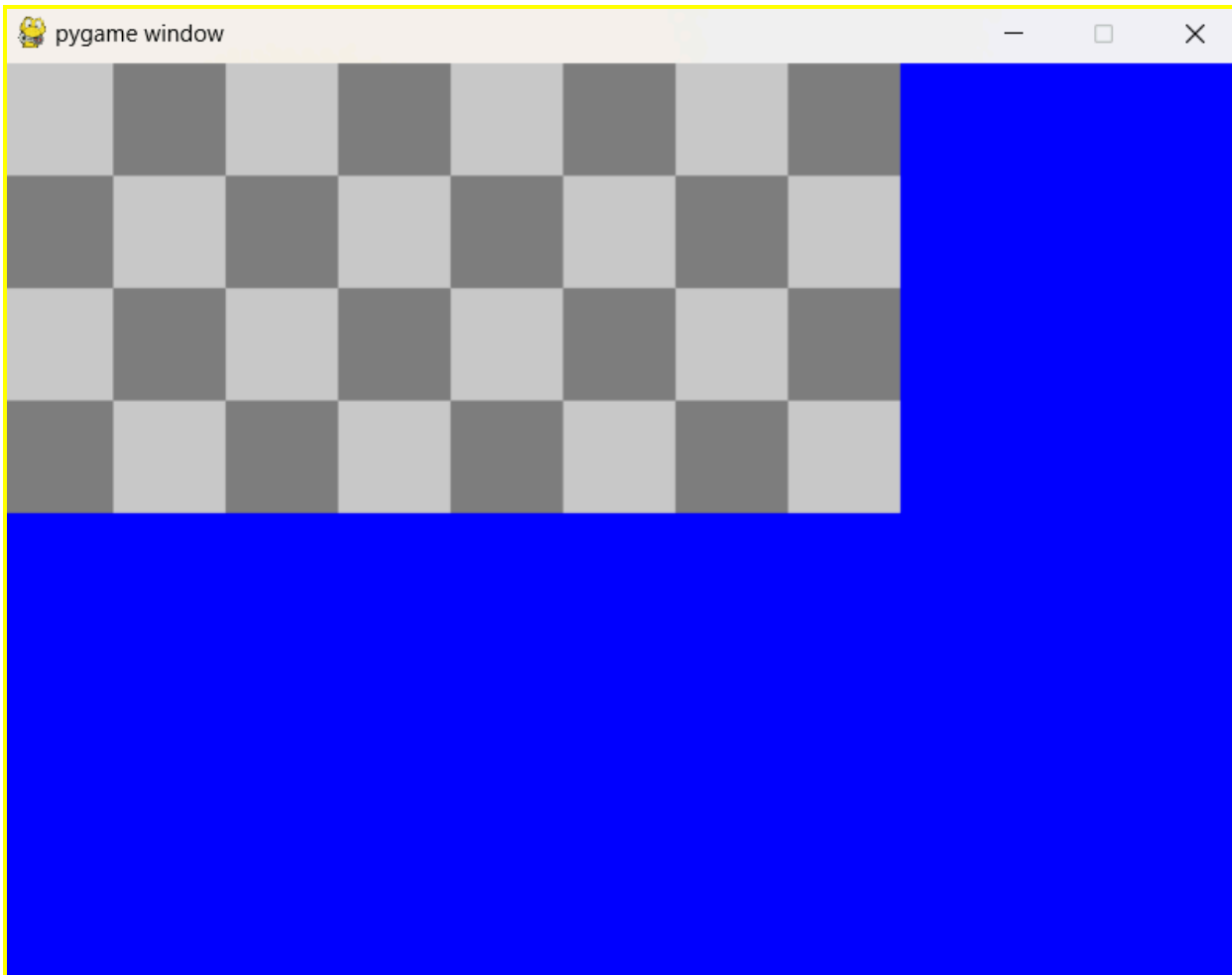
### ***EJERCICIO2E: Alternancia Inversa de Colores***



```
Ejercicio2b.py  Ejercicio2c.py  Ejercicio2d.py  Ejercicio2e.py X  ▸ ▾  □  ⋮
LaboratorioIVOficial_Marron_Puma_Maritz > EjerciciosPropuestos > Archivos_Tablero > Ejercicio2e.py > ...
1  from interpreter import draw
2  from chessPictures import *
3
4  row_1 = square.negative().join(square).horizontalRepeat(4)
5  board = row_1
6  draw(board)
```

En este ejercicio, se crea una fila inversa en términos de colores, comenzando con un cuadrado negro seguido de uno blanco. Repitiendo cuatro veces este patrón utilizando *horizontalRepeat()*.

### **EJERCICIO2F: Tablero Completo de Ajedrez**

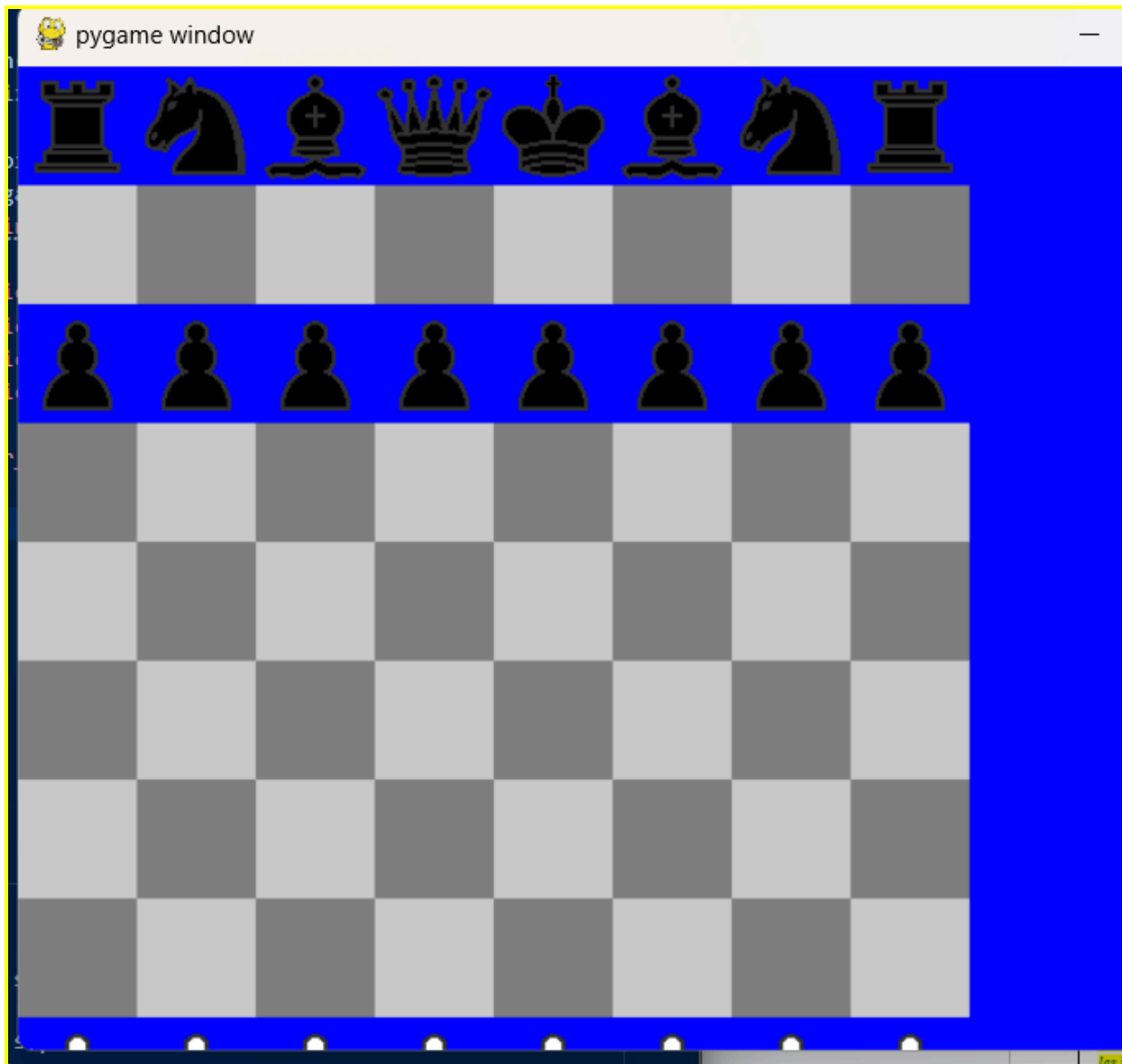


```
Ejercicio2c.py  Ejercicio2d.py  Ejercicio2e.py  Ejercicio2f.py X  ▸ ▾ □ ⋮
LaboratorioIVOOficial_Marron_Puma_Maritza > EjerciciosPropuestos > Archivos_Tablero > Ejercicio2f.py > ...
1  from interpreter import draw
2  from chessPictures import *
3
4  row_whiteFirst = square.join(square.negative()).horizontalRepeat(4)
5  row_blackFirst = square.negative().join(square).horizontalRepeat(4)
6  chess_board = row_whiteFirst.under(row_blackFirst).verticalRepeat(2)
7  draw(chess_board)
```



En este ejercicio se debe generar un tablero de ajedrez completo alternando casillas blancas y negras en filas alternas. **row\_whiteFirst** crea una fila comenzando con un cuadrado blanco, mientras que **row\_blackFirst** comienza con un cuadrado negro. Ambas filas se combinan y se repiten verticalmente utilizando **verticalRepeat()** para completar el tablero.

### **EJERCICIO2G: Posición de Piezas en el Tablero**



```

Ejercicio2b.py  Ejercicio2c.py  Ejercicio2d.py  Ejercicio2e.py  Ejercicio2f.py  Ejercicio2g.py X  interpreter.py  chessPictures.py  picture.py  chessPictures.python-311.pyc  > ~  [ ]
LaboratorioIVOficial_Marron_Puma_Maritz > EjerciciosPropuestos > Archivos_Tablero > Ejercicio2g.py > ...
1  from interpreter import draw
2  from chessPictures import *
3
4  fila_piezas_negras = rock.negative().join(knight.negative().join(bishop.negative().join(queen.negative().join(king.negative().join(bishop.negative().join(knight.negative().join(rock.negative()))))))))
5  fila_peones_negras = pawn.negative().horizontalRepeat(8)
6
7  fila_piezas_blancas = rock.join(knight.join(bishop.join(queen.join(king.join(bishop.join(knight.join(rock)))))))
8  fila_peones_blancas = pawn.horizontalRepeat(8)
9
10 fila_inicial_blanca = square.join(square.negative()).horizontalRepeat(4)
11 fila_inicial_negra = square.negative().join(square).horizontalRepeat(4)
12 filas_vacias_repetidas = fila_inicial_blanca.under(fila_inicial_negra).verticalRepeat(20)
13
14 fila_superior_piezas = fila_inicial_blanca.up(fila_piezas_negras)
15 fila_superior_peones = fila_inicial_negra.up(fila_peones_negras)
16 fila_inferior_peones = fila_inicial_blanca.up(fila_peones_blancas)
17 fila_inferior_piezas = fila_inicial_negra.up(fila_piezas_blancas)
18
19 tablero_ajedrez = fila_superior_piezas.under(fila_superior_peones.under(filas_vacias_repetidas.under(fila_inferior_peones.under(fila_inferior_piezas))))
20 draw(tablero_ajedrez)
21

```

*Este ejercicio tiene como objetivo generar un tablero de ajedrez utilizando imágenes de las piezas y casillas. Se utilizan diversas funciones de la clase **chessPictures** para manipular, unir y organizar las imágenes, creando un diseño visualmente representativo de un tablero de ajedrez completo.*

## Conclusiones

A través de este ejercicio, he aprendido a manipular imágenes y representaciones de piezas de ajedrez de una manera dinámica y creativa. Implementar funciones como `join()`, `under()`, y `negative()` me ha permitido ver cómo se pueden combinar elementos para formar estructuras más complejas, como un tablero de ajedrez. Este proceso de construcción me ha ayudado a apreciar la importancia de la modularidad

La creación de patrones y la disposición de las piezas me ha enseñado a pensar de manera más visual y estratégica, algo que es fundamental en el ajedrez. A medida que configuraba el tablero, comprendí mejor cómo cada pieza tiene su propia función y cómo interactúan entre sí en el juego.

## REFERENCIAS Y BIBLIOGRAFÍA

r

*Referencias de las clases proporcionadas (Picture, chessPictures) para la creación de imágenes y tableros.*

Contenido y demostración		Puntos	Checklis t	Estudiant e	Profeso r
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).	3	x	1	
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	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	x	1	
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	2	
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		14	

### ¡OJO! Anexar a su informe

#### RUBRICA PARA EL CONTENIDO DEL INFORME Y DEMOSTRACIÓN

El alumno debe marcar o dejar en blanco en celdas de la columna Checklist si cumplió 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

