

Informe LABORATORIO 04

Tema: Python

Nota

Estudiante	Escuela	Asignatura
Jeanpier Michaelson Valera Yana jvalera@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación web 2 Semestre: III Código: 20212110

Laboratorio	Tema	Duración
04	Python	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	Del 29 Mayo 2023	Al 02 Junio 2023

1. Tarea

- URL GitHub de Tarea del Ajedrez <https://github.com/rescobedoq/pw2/tree/main/labs/lab04/Tarea-del-Ajedrez>.
- En esta tarea usted pondra en practica sus conocimientos de programacion en Python para dibujar un tablero de Ajedrez.
- La parte grafica ya esta programada, usted solo tendra que concentrarse en las estructuras de datos subyacentes.
- Con el codigo proporcionado usted dispondra de varios objetos de tipo Picture para poder realizar su tarea.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux 23 lunar 64 bits Kernell 6.2.
- VIM 9.0.
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.
- Programación Orientada a Objetos.
- Algoritmo de ordenamiento por inserción

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/valera0wner/pwb2-lab-c-23a.git>
- URL para el laboratorio 01 en el Repositorio GitHub.
- <https://github.com/valera0wner/pwb2-lab-c-23a/tree/main/Lab04-pweb2>

4. Actividades con el repositorio GitHub

4.1. Creando e inicializando repositorio GitHub

- Como es el cuarto laboratorio se creo la carpeta en github GitHub.
- Se realizaron los siguientes comandos en la computadora:

Listing 1: Creando carpeta de trabajo

```
$ mkdir Lab04-pweb2
```

Listing 2: Dirigiéndonos al directorio de trabajo

```
$ cd Lab04-pweb2
```

Listing 3: Haciendo exercises

```
$ mkdir exercises  
$ code esEscalar.py  
$ code esUnitaria.py  
$ code test_esEscalar.py  
$ code test_esUnitaria.py  
$ git add README.md  
$ git commit -m "agregando exercises con archivos .py"  
$ git push -u origin main
```

4.2. Commits

Listing 4: Haciendo el ejercicio2a

```
$ code Ejercicio2a.py  
$ git add .  
$ git commit -m "Haciendo el ejercicio2a, haciendo arrays"  
$ git push -u origin main
```

Showing 1 changed file with 6 additions and 2 deletions.

Lab04-pweb2/Tarea-del-Ajedrez/Ejercicio2a.py	
@@ -1,6 +1,10 @@	
1 - from interpreter import *	1 + from interpreter import draw
2 from chessPictures import *	2 from chessPictures import *
3 from picture import Picture	3 from picture import Picture
4	4
5 - draw(knight.verticalMirror())	5 + pieces = [king, knight, queen]
	6 +
	7 +
	8 + for piece_img in pieces:
	9 + draw(piece_img)
6	10

- Se creo el archivo **.gitignore** para no considerar los archivos dentro de myenv **Scripts/** que son innecesarios hacer seguimiento.

Listing 5: Creando .gitignore

```
$ vim .gitignore
```

Listing 6: .gitignore

```
my_env/lib/*
my_env/Scripts/*
my_env/Include/*
```

Listing 7: Commit: Creando .gitignore para archivos dentro del entorno virtual

```
$ git add .
$ git commit -m "Creando .gitignore para archivos "
$ git push -u origin main
```

Code Blame 8 lines (7 loc) · 123 Bytes

```

1
2 my_env/bin/*
3 my_env/lib/*
4 my_env/Scripts/*
5 my_env/Include/*
6 my_env/src/__pycache__/*
7 *.pyc
8 Tarea-del-Ajedrez/__pycache__/*
```

- Para el siguiente commit se implemento el algoritmo de invertir el color, usando un metodo anteriormente creado que hacia el uso de un archivo llamado colors.py .
- El código utilizado es el siguiente:

```
10 def _invColor(self, color):
11     if color not in inverter:
12         return color
13     return inverter[color]
14
```

```
33 def negative(self):
34     """ Devuelve un negativo de la imagen """
35     negativo=[]
36
37     for value in self.img:
38         linea=''
39         for caracter in value:
40             linea = linea+self._invColor(caracter)
41         negativo.append(linea)
42
43     return Picture(negativo)
```

- Se utilizó para el primer archivo de Ejercicio2a.

```
1 from interpreter import draw
2 from chessPictures import *
3
4
5 arriba = knight.join(knight.negative())
6 abajo = knight.negative().join(knight)
7
8 img = arriba.up(abajo)
9
10 draw(img)
```

- archivo .py ejecutado en la consola.



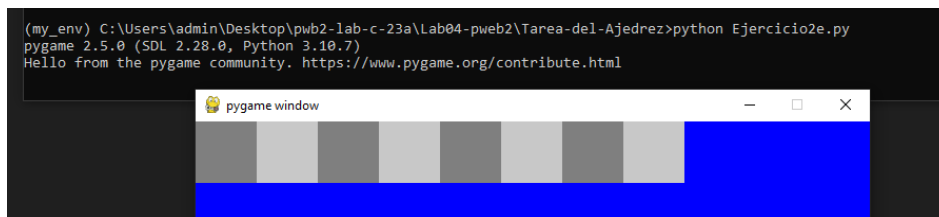
Listing 8: 4to commit

```
$ git commit -m "Agregando metodos y ultimas modificaciones"
```

- Aca hago el ejercicio2e y ejercicio2d, cada uno de ellos me genera una fila de cuadros con colores casi invertidos, uno empieza por el color negativo del otro, estos cuadrados son 7 y sus colores se van intercambiando



- Segundo ejemplo



4.3. Estructura de laboratorio 04

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

El número de serie del volumen es 7E34-14B3

C:.

```
.gitignore
chessPictures.py
colors.py
Ejercicio2a.py
Ejercicio2b.py
Ejercicio2c.py
Ejercicio2d.py
Ejercicio2e.py
Ejercicio2f.py
Ejercicio2g.py
interpreter.py
picture.py
pieces.py
requirements.txt

__pycache__
  chessPictures.cpython-310.pyc
  colors.cpython-310.pyc
  interpreter.cpython-310.pyc
  picture.cpython-310.pyc
  pieces.cpython-310.pyc
```

5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

5.2. 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 2: 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 3: 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			
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	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			
Total		20		12	

6. Referencias

- <https://www.w3schools.com/java/default.asp>
- <https://www.geeksforgeeks.org/insertion-sort/>