

Informe de Laboratorio 03

Tema: PYTHON

Nota

Estudiante	Escuela	Asignatura
Gabriel Soto Ccoya gsotocco@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 02 Semestre: III Código: 20220583

Laboratorio	Tema	Duración
03	PYTHON	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	Del 29 Mayo 2023	Al 08 Junio 2023

1. Tarea

- 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.
- Usando únicamente los métodos de los objetos de la clase Picture, dibuje las figuras (invocando el método "draw()").

2. Equipos, materiales y temas utilizados

- Sistema operativo Windows 11.
- Librería PyGame.
- Python 3.10.11.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional (GabSoto).
- TextMaker / GhostScript / MikTex
- Programación Orientada a Objetos.
- Manejo de matrices unidimensionales.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/GabSoto/Pweb02-Lab03.git>
- URL para el laboratorio 04 en el Repositorio GitHub.
- <https://github.com/GabSoto/Pweb02-Lab03/tree/main/Laboratorio%2003>

4. Implementación de los métodos

4.1. Método *horizontalMirror()*

El presente método nos pedía invertir una pieza del tablero 180°, para lograr esto se recorre el array desde atrás mientras que se va guardando estos elementos en un nuevo array.

Listing 1: horizontalMirror.py

```
1 def horizontalMirror(self):
2     horizontal = []
3     a=0
4     for i in range (len(self.img)):
5         a-=1
6         horizontal.append(self.img[a])
7     return Picture(horizontal)
```

4.2. Método *negative()*

El siguiente método nos pedía invertir el color de cualquier pieza, esto usando el método ya proporcionado `_invColor()`. Luego se reemplaza cada fila de los arrays por su contracolor.

Listing 2: negative.py

```
1 def negative(self):
2     negative = []
3     for row in self.img:
4         inverted_row = [self._invColor(color) for color in row]
5         negative.append(inverted_row)
6     return Picture(negative)
```

4.3. Método *join()*

La función de este método era poder fusionar dos imágenes en una, colocando en la derecha la imagen recibida (p). Para lograr esto, se convirtió las imágenes en arrays con el `.img` para luego agregar ambos en un array `new_img`.

Listing 3: join.py

```
1 def join(self, p):
2
3     new_img = []
4
5     for i in range(len(self.img)):
6         new_img.append(list(self.img[i]) + list(p.img[i]))
```

```
7
8     return Picture(new_img)
```

4.4. Método *up()*

Este método nos pedía guardar una imagen sobre otra, esto se logró usando el método `.append`, dónde agregamos el array de la imagen al final.

Listing 4: up.py

```
1 def up(self, p):
2     up = p.img
3
4     for i in range(len(self.img)):
5         up.append(self.img[i])
6     return Picture(up)
```

4.5. Método *under()*

Esta función era para poner una imagen sobre otra, en este caso se reemplaza todos los Strings vacíos con el array de la imagen del parámetro.

Listing 5: under.py

```
1 def under(self, p):
2
3     under = []
4
5     for i in range(len(self.img)):
6         str = ""
7         for j in range(len(self.img[i])):
8             if(self.img[i][j] == " "):
9                 str+=p.img[i][j]
10            else:
11                str+=self.img[i][j]
12
13     under.append(str)
14
15     return Picture(under)
```

4.6. Método *horizontalRepeat()*

Este método constaba en imprimir (n) veces el mismo elemento de forma horizontal, funciona usando un bucle en base a (n) y el método `join()`.

Listing 6: horizontalRepeat.py

```
1 def horizontalRepeat(self, n):
2     repeated_img = self.img
3
4     for i in range(n):
5         repeated_img = self.join(Picture(repeated_img)).img
6
7     return Picture(repeated_img)
```

4.7. Método *verticalRepeat()*

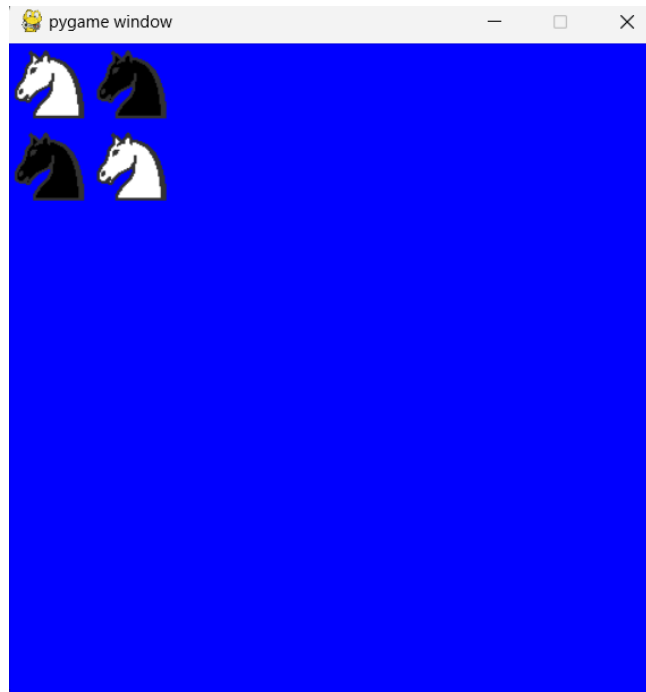
Este método constaba en imprimir (n) veces el mismo elemento de forma vertical, funciona usado un bucle en base a (n) y el método up().

Listing 7: verticalRepeat.py

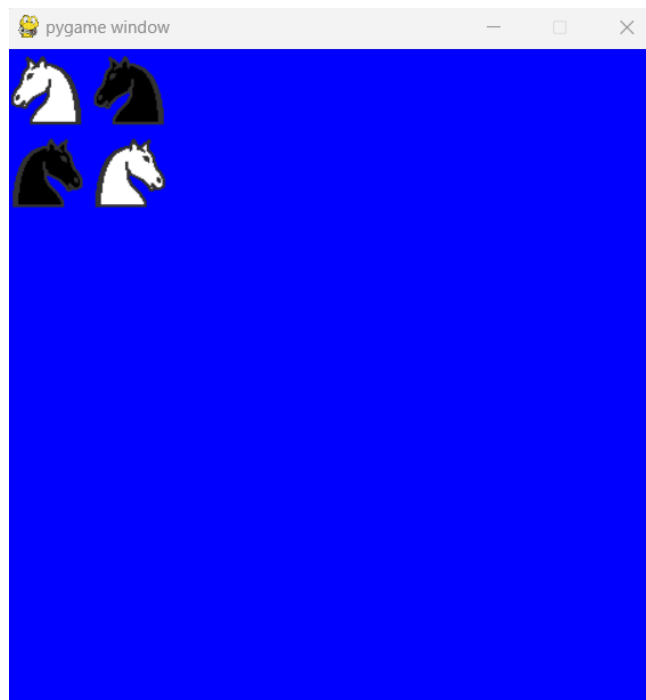
```
1 def verticalRepeat(self, n):  
2     repeated_img = self.img  
3  
4     for i in range(n):  
5         repeated_img = self.up(Picture(repeated_img)).img  
6  
7     return Picture(repeated_img)
```

5. Ejecuciones

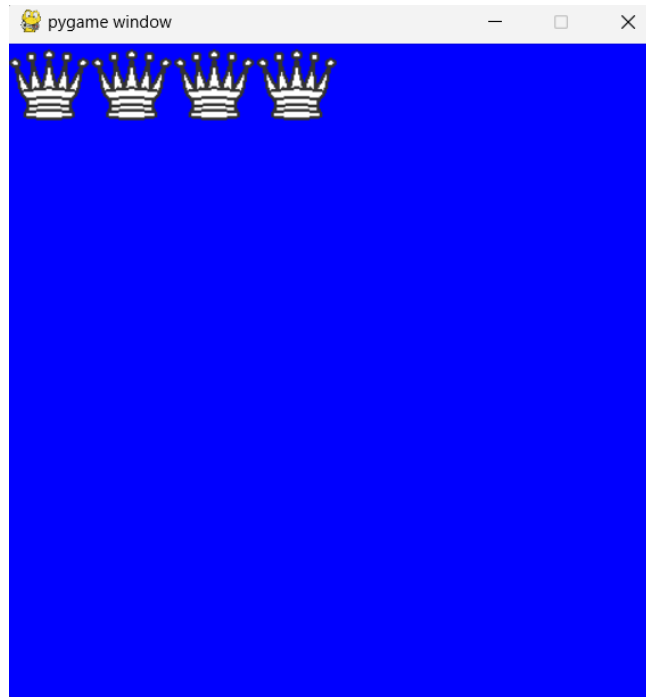
5.1. Ejercicio A:



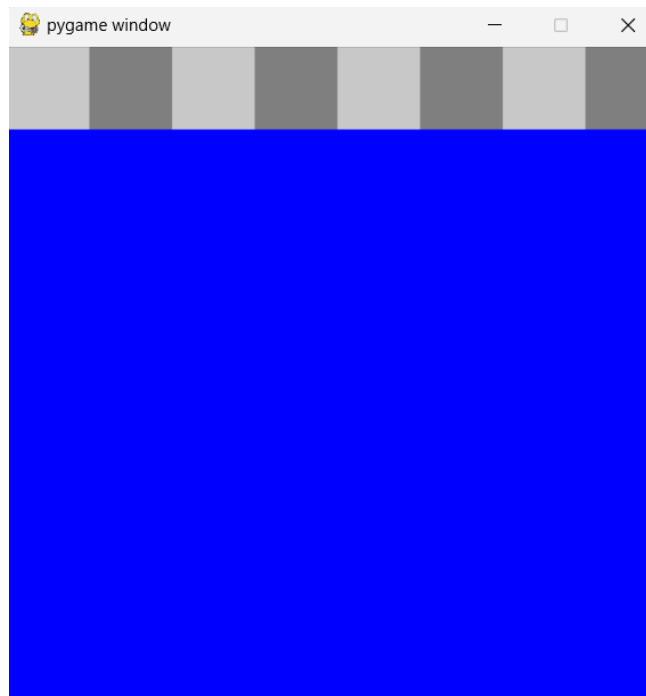
5.2. Ejercicio B:



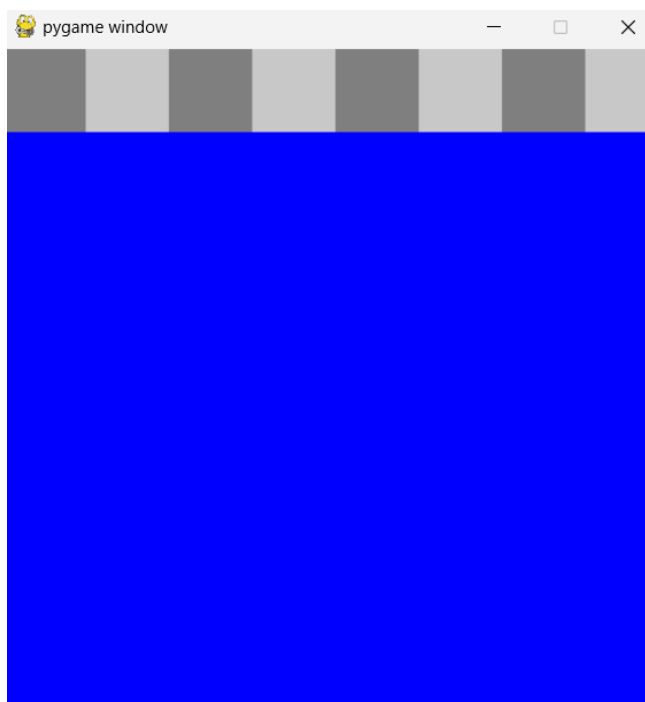
5.3. Ejercicio C:



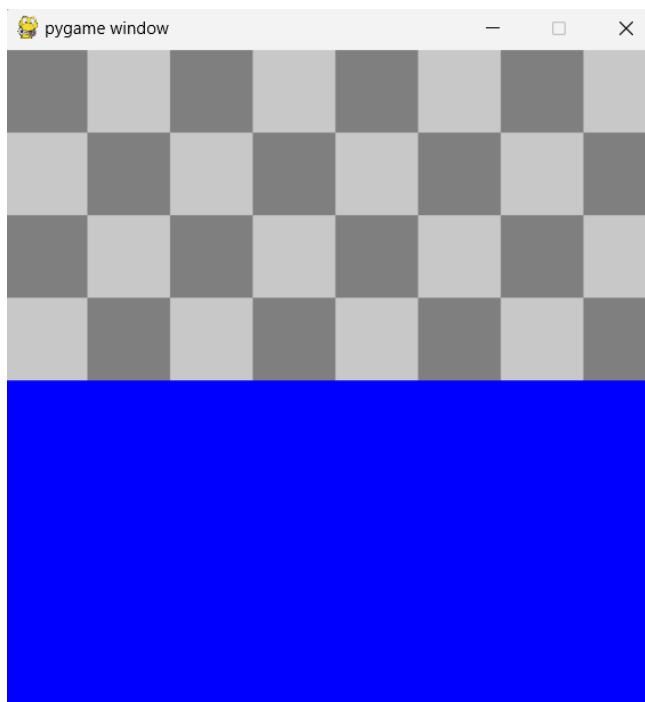
5.4. Ejercicio D:



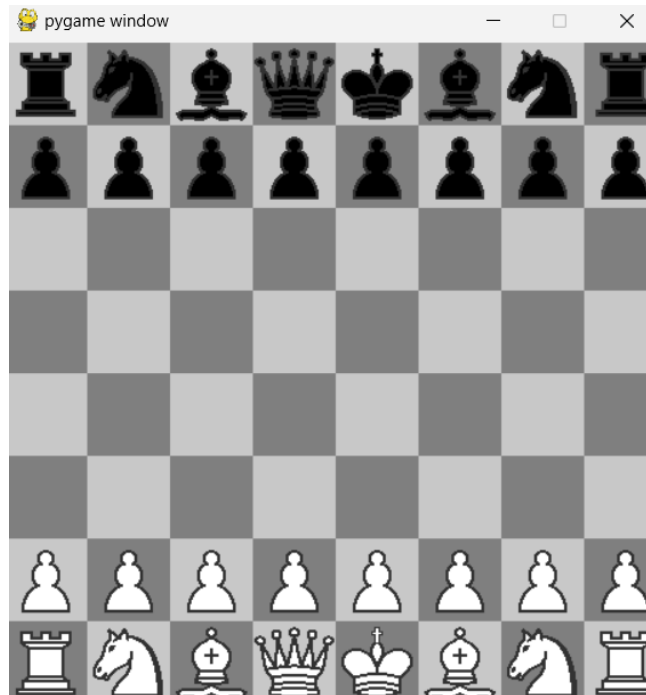
5.5. Ejercicio E:



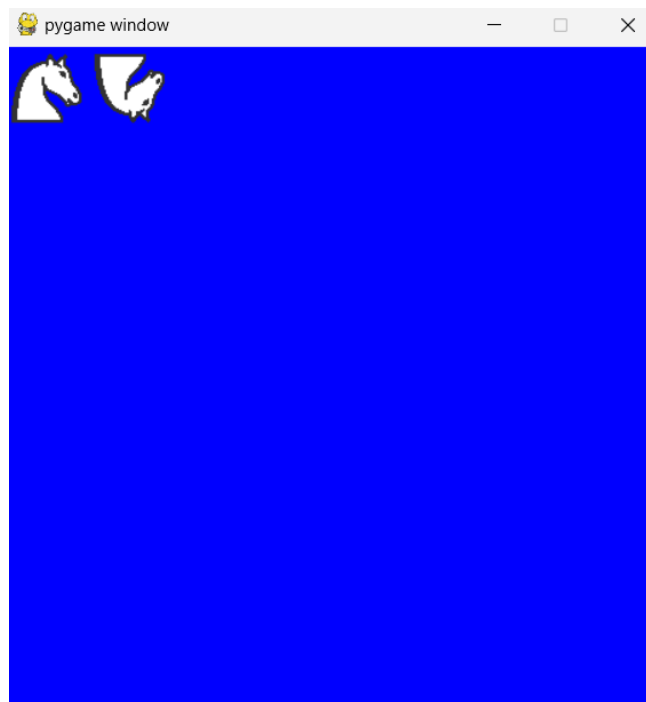
5.6. Ejercicio F:



5.7. Ejercicio G:



5.8. Usando el método *horizontalMirror()*



5.9. Estructura de laboratorio 03

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

```
Pweb2-Lab03/
|-- Laboratorio 04
| |-- chessPictures.py
| |-- colors.py
| |-- Ejercicio2a.py
| |-- Ejercicio2b.py
| |-- Ejercicio2c.py
| |-- Ejercicio2d.py
| |-- Ejercicio2e.py
| |-- Ejercicio2f.py
| |-- Ejercicio2g.py
| |-- interpreter.py
| |-- picture.py
| |-- pieces.py
| |-- prac.py
| |
| |-- __pycache__
| |   |-- chessPictures.cpython-310.pyc
| |   |-- colors.cpython-310.pyc
| |   |-- interpreter.cpython-310.pyc
| |   |-- picture.cpython-310.pyc
| |   |-- pieces.cpython-310.pyc
| |
|-- Latex
| |-- Laboratorio_04.aux
| |-- Laboratorio_04.log
| |-- Laboratorio_04.out
| |-- Laboratorio_04.pdf
| |-- Laboratorio_04.synctex.gz
| |-- Laboratorio_04.tex
| |
| |-- img
| |   |-- logo_abet.png
| |   |-- logo_episunsa.png
| |   |-- logo_unsa.jpg
| |   |-- Picture A.png
| |   |-- Picture B.png
| |   |-- Picture C.png
| |   |-- Picture D.png
| |   |-- Picture E.png
| |   |-- Picture F.png
| |   |-- Picture G.png
| |   |-- Picture H.png
| |
| |-- src
| |   |-- horizontalMirror.py
| |   |-- horizontalRepeat.py
| |   |-- join.py
| |   |-- negative.py
| |   |-- under.py
| |   |-- up.py
| |   |-- verticalRepeat.py
```

5.10. Commits de laboratorio 03

Listing 8: Commits

```
commit 87a022325ef113ff075fd0092fc1e47be42348f6 (HEAD -> main, origin/main)
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 22:06:44 2023 -0500

    Update DISPLAY and Ejercicio2g.py

commit 0a466385d51032ee2a1e4c823fad2af30ae50c
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 14:47:32 2023 -0500

    Terminando el ejercicio g e implementando metodo under()

commit 2ef3d41f10eaebd145ae2d9b76129ceb7e2174b9
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 12:54:11 2023 -0500

    Terminando el metodo horizontalMirror() e implementando un ejercicio mas para mostrar su
    funcionalidad.

commit 21d0a3319ffa12b82ed50981d7fc4417d48655d6
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 12:28:27 2023 -0500

    Terminando el ejercicio f e implementando el metodo verticalRepeat()

commit f16732866a68383cbfdded968b415e5bc007669b8
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 12:25:37 2023 -0500

    Reutilizamos el mismo metodo horizontalRepeat para este ejercicio.

commit ebfedabf5e9ce2db15067b4efaf2b0c9c277dd9
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 12:24:13 2023 -0500

    Implementando el metodo horizontalRepeat() y terminando el Ejercicio d

commit 8bb3088a660b7a0c5b48a16c7159e248ddb0a61
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 12:22:55 2023 -0500

    corrigien el metodo up por el metodo under

commit 82b84503508477969bc6eaac74791fe7e9237c61
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 11:39:05 2023 -0500

    Terminando el ejercicio c e implementando el metodo horizontalRepeat()

commit ae9db59ea28960ed8e16f42c24f469174efab599
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 10:42:18 2023 -0500
```

Elaboracion del ejercicio b

```
commit 65e1c2b6fc485fae77fa36e495bb0e56d0c0f1eb
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Tue Jun 6 07:41:25 2023 -0500
```

Terminando el ejercicio a e implementando los metodos under() y join()

```
commit 7239b1d2ee08c160d4d969273fa63977da7140d8
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Mon Jun 5 21:35:16 2023 -0500
```

Implementando los metodos join() y under()

```
commit 1e7b36822133423c4ba833bf75f83ad9ac7ae0c0
Author: GabSoto <gsotocco@unsa.edu.pe>
Date: Mon Jun 5 20:44:16 2023 -0500
```

```
Implementando la funcion negative()
(END)
```

6. Preguntas:

¿Qué son los archivos *.pyc?

Los archivos con extensión '.pyc' en Python son generados por el intérprete de Python como una forma de almacenar el bytecode compilado del código fuente. Cuando un archivo de Python se ejecuta por primera vez, el intérprete compila el código fuente a bytecode y guarda ese bytecode en un archivo ".pyc". Así evitando la necesidad de compilar el código nuevamente. De manera similar, en Java, los archivos ".class" contienen el bytecode compilado a partir de los archivos de código fuente ".java". Estos archivos ".class" se utilizan para ejecutar el programa de manera eficiente sin tener que recompilar el código fuente en cada ejecución.

¿Para qué sirve el directorio pycache?

Aquí se almacenan los archivos compilados de los módulos: chessPictures.py, colors.py, interpreter.py, picture.py y piezas.py. Estos archivos compilados, que se encuentran en el directorio "pycache", permiten evitar la necesidad de recompilarlos cada vez que se ejecuten otros archivos que los importen.

¿Cuáles son los usos y lo que representa el subguión en Python?

El subguión (.) en Python tiene varios usos y significados adicionales además de ser utilizado en constructores y para indicar métodos privados de una clase (como en la clase Picture). Algunos de estos usos incluyen:

- **Traducción de cadenas:** En la función `str.maketrans()` o en el método `str.translate()`, puedes utilizar el subguión para indicar que un carácter debe ser eliminado o dejado sin cambios durante una operación de traducción de cadenas.

Listing 9: Ejemplo

```
translation_table = str.maketrans("aeiou", "____")
```

```
resultado = "Hola, mundo!".translate(translation_table)
print(resultado)
```

Ouput >> H_l_, m_nd_!

- **Ignorar valores en desempaquetado:** Al desempaquetar una secuencia o iterador, puedes utilizar el subguión para ignorar los valores que no necesitas.

Listing 10: Ejemplo

```
a, _, c = (1, 2, 3)
```

7. Rúbricas

7.1. 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.

7.2. 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	X	3	
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	X	4	
Total		20		19	

8. Referencias

- Instalación de Pygame. <https://www.youtube.com/watch?v=RRir10k0pfI>
- Instalación de Latex. <https://www.youtube.com/watch?v=GNp5IuWSZA0&t=25s>
- Introducción a Python. https://www.w3schools.com/python/python_reference.asp