

UNIVERSIDAD GERARDO BARRIOS
FACULTAD DE CIENCIAS Y TECNOLOGIA



CATEDRA:

PROGRAMACION COMPUTACIONAL III

CATEDRATICO:

WILLIAN ALEXIS MONTES GIRON

ACTIVIDAD:

PARCIAL II

ESTUDIANTES

EDWIN JOSÉ PARADA CAMPOS (**SMTR043624**)

ROBERTO CARLOS CHÁVEZ CAMPOS (**SMTR044024**)

YADER ENMANUEL ROMERO (**SMTR095924**)

CARLOS FELIPE SOTO MAYORGA (**SMTR079724**)

Índice

Introducción	3
Objetivos	4
Librerías de Python	5
Ejemplos de librerías.	5
Trabajo por realizar.	6
¿En qué consiste la librería y para qué se usa?	7
¿Para qué se usa?.....	7
Funciones más relevantes y utilizadas de la librería	7
Casos de uso en los que la librería se ha utilizado en la vida real.....	8
Juegos hechos con Pygame.....	9
Referencias bibliográficas	11

Introducción

En este documento se aborda el concepto y la utilidad de las librerías en Python, entendidas como colecciones de módulos que agrupan funciones, clases y herramientas ya desarrolladas para agilizar tareas comunes. A través de ejemplos como NumPy, TensorFlow, Pandas o Matplotlib, se demuestra cómo estas librerías permiten ahorrar tiempo, reducir errores y acceder a soluciones optimizadas sin partir de cero.

Para ejemplificar la aplicación práctica de estos conceptos, el grupo nombardo “Pac-Man” ha decidido centrar el trabajo en Pygame, una biblioteca de código abierto diseñada para la creación de videojuegos en Python. Se describe su funcionamiento básico como, controles de usuario, gestión de gráficos y sonido, bucles de juegos para usarlo en proyecto del parcial que es la recreación del clásico PacMan.

El dominio de librerías populares no solo acelera el desarrollo de prototipos y soluciones finales, sino que también constituye una habilidad esencial en entornos académicos. Al comprender en profundidad cómo funcionan y cómo integrarlas de manera eficiente, los desarrolladores pueden centrarse en la lógica de negocio y la creatividad, dejando atrás tareas que a veces como estudiantes se nos hacen difíciles y complejas. Este trabajo busca, por tanto, aumentar nuestro conocimiento sobre el lenguaje de Python

Objetivos

Objetivo general:

- Comprender el uso de librerías en Python el uso de una librería de preferencia del equipo.

Objetivos específicos:

- Conocer en que casos de la vida real se pueden usar las diferentes librerías que la comunidad de python ofrece.
- Analizar e implementar las funciones clave de Pygame para gráficos, sonido y eventos.

Librerías de Python

Una librería de Python es un conjunto de módulos que contienen funciones, clases y herramientas ya escritas por otros desarrolladores para facilitar tareas específicas. En lugar de escribir todo desde cero, puedes importar una librería y aprovechar su funcionalidad.

¿Para qué sirven?

Las librerías te permiten:

- Ahorrar tiempo y esfuerzo.
- Evitar errores comunes.
- Acceder a soluciones optimizadas y probadas.

Ejemplos de librerías.

1. NumPy

- **Qué es:** Una librería para cálculos numéricos.
- **Función principal:** Manejo eficiente de arrays multidimensionales y operaciones matemáticas como álgebra lineal, estadísticas y transformaciones.
- **Ejemplo de uso:** Procesamiento de datos científicos o simulaciones físicas.

2. TensorFlow

- **Qué es:** Una librería de código abierto para machine learning y redes neuronales.
- **Función principal:** Crear, entrenar y desplegar modelos de aprendizaje automático, especialmente redes profundas.
- **Ejemplo de uso:** Reconocimiento de imágenes, predicción de texto, clasificación de datos.

3. Pandas

- **Qué es:** Una librería para análisis y manipulación de datos estructurados.
- **Función principal:** Trabajar con tablas (DataFrames), limpiar datos, agrupar, filtrar y transformar información.
- **Ejemplo de uso:** Análisis de datos financieros, encuestas, registros médicos.

4. PyQt

- **Qué es:** Un conjunto de herramientas para crear interfaces gráficas (GUI) usando Qt en Python.
- **Función principal:** Diseñar ventanas, botones, menús y componentes interactivos.
- **Ejemplo de uso:** Aplicaciones de escritorio con visualización de datos o formularios.

5. Matplotlib

- **Qué es:** Una librería para crear gráficos y visualizaciones.
- **Función principal:** Generar gráficos de líneas, barras, dispersión, histogramas, etc.
- **Ejemplo de uso:** Mostrar resultados de experimentos, visualizar tendencias o estadísticas.

6. smtplib

- **Qué es:** Una librería estándar de Python para enviar correos electrónicos.
- **Función principal:** Conectarse a servidores SMTP y enviar mensajes desde scripts.
- **Ejemplo de uso:** Automatizar notificaciones por correo, enviar reportes o alertas.

7. Pillow

- **Qué es:** Una librería para procesamiento de imágenes.
- **Función principal:** Abrir, modificar, guardar y aplicar filtros a imágenes.
- **Ejemplo de uso:** Redimensionar fotos, convertir formatos, aplicar efectos visuales

Trabajo por realizar.

Nuestro grupo a tomado a bien hacer un trabajo usando la librería de Pygame, es una librería bastante conocida y útil en el campo de la creación de videojuegos, además pensamos que hacer este tipo de trabajos nos ayudara a conocer nuevas formas de trabajar usando Python.

¿En qué consiste la librería y para qué se usa?

Como hemos dicho anteriormente nosotros usaremos Pygame, es una biblioteca de código abierto que proporciona herramientas para desarrollar videojuegos en Python. Fue creada para facilitar el acceso a gráficos, sonido y eventos sin necesidad de dominar programación avanzada. Se usa para manejar gráficos, sonido, animaciones y controles de usuario de forma sencilla y eficiente.

¿Para qué se usa?

Pygame permite:

- Dibujar gráficos en pantalla: como personajes, fondos, objetos y efectos visuales.
- Cotrolar el teclado, mouse y joystick: para que el jugador interactúe con el juego.
- Reproducir sonidos y música: incluyendo efectos especiales y bandas sonoras.
- Crear animaciones: como movimientos suaves de personajes o transiciones.
- Diseñar bucles de juego: que actualizan la pantalla y responden a eventos en tiempo real.

En nuestro caso recrearemos el juego de PacMan, es un juego sencillo y bastante conocido, además que Pygame permite crear este tipo de juegos.

Funciones más relevantes y utilizadas de la librería

A continuación, se presentan al menos ocho funciones que utilizaremos en el trabajo para recrear el juego de PacMan, aunque puede que añadamos más si el juego lo requiere.

- `pygame.init()`: inicia todos los módulos de Pygame, preparando la librería para su uso.
- `pygame.quit()`: cierra todos los módulos de Pygame y libera los recursos asociados.
- `pygame.display.set_mode(resolution)`: crea la ventana o superficie principal de dibujo con la resolución especificada (tupla ancho×alto).
- `pygame.display.update([rect_list])`: refresca la pantalla; si se pasa una lista de rectángulos, solo actualiza esas áreas, mejorando el rendimiento.
- `pygame.event.get()`: recupera todos los eventos pendientes (teclado, ratón, cierre de ventana) y los devuelve como una lista.

- `pygame.key.get_pressed()`: devuelve un listado booleano del estado actual de todas las teclas, ideal para detectar pulsaciones continuas.
- `pygame.time.Clock().tick(fps)`: retrasa el bucle de juego para que no supere la tasa de frames por segundo indicada, manteniendo una velocidad consistente.
- `pygame.image.load(path)`: carga una imagen desde disco y devuelve un Surface, base para renderizar sprites y fondos.
- `pygame.transform.scale(surface, size)`: redimensiona un Surface a un nuevo tamaño, útil para adaptar assets a diferentes resoluciones.
- `pygame.draw.rect(surface, color, rect, width=0)`: dibuja un rectángulo (relleno si `width=0`) en la superficie indicada, clave para visualizaciones y colisiones.
- `pygame.mixer.Sound(path)`: carga un efecto de sonido desde archivo; combinado con `.play()` permite reproducir audio puntual.
- `pygame.mixer.music.load(path)` / `pygame.mixer.music.play(loops)`: gestiona la música de fondo, cargando y reproduciendo pistas con bucle opcional.
- `pygame.sprite.Group()`: crea un contenedor para múltiples sprites, facilitando llamadas masivas a `.update()` y `.draw()`.
- `pygame.sprite.Sprite()`: clase base para crear sprites con atributos `image` y `rect`, integrándose con los grupos de Pygame.

Casos de uso en los que la librería se ha utilizado en la vida real.

- **Educación y talleres:** Se emplea en colegios, universidades y bootcamps para enseñar programación de forma lúdica, con ejemplos interactivos que van desde simulaciones de física hasta minijuegos guiados.
- **Prototipado rápido en Game Jams:** Equipos y desarrolladores independientes lo eligen para armar demos en 48–72 horas, aprovechando su loop de juego, manejo de sprites y entrada de usuario listos para usar.
- **Simulaciones interactivas:** Investigadores y estudiantes construyen visualizaciones de algoritmos, modelos de tráfico o agentes autónomos con Pygame como interfaz visual a sus cálculos.
- **Interfaces gráficas ligeras:** Pequeñas aplicaciones de kioscos, prototipos de señalética digital o paneles de control usan Pygame cuando no requieren toda la complejidad de un framework GUI convencional.

Juegos hechos con Pygame.

Frets on Fire: Un juego de ritmo musical de código abierto, muy similar al popular Guitar Hero. Los jugadores utilizan el teclado para "tocar" las notas de las canciones que aparecen en pantalla. Es uno de los ejemplos más famosos y exitosos de lo que se puede lograr con Pygame.

Solar-Wolf: Un juego de acción y arcade en el que controlas una nave espacial que debe recoger cubos de energía mientras esquiva minas y ataques enemigos. Es un buen ejemplo de un juego de estilo retro con una jugabilidad adictiva.

Dangerous Dave: Si bien el original es un clásico, existen numerosas recreaciones y clones hechos en Pygame como un ejercicio de aprendizaje y homenaje. Este juego de plataformas de acción muestra la capacidad de Pygame para manejar niveles basados en baldosas, colisiones y movimiento de personajes.

Save the Date: Una novela visual galardonada que, aunque pueda parecer simple en su concepto, explora temas complejos a través de la narrativa y la elección del jugador. Demuestra la flexibilidad de Pygame más allá de los géneros de acción y arcade.



Conclusión

El presente trabajo ha mostrado cómo las librerías en Python facilitan el desarrollo de proyectos al ofrecer módulos especializados que simplifican tareas complejas. Tras revisar ejemplos esenciales como NumPy, Pandas y Matplotlib, se profundizó en Pygame para ilustrar su potencial en la creación de videojuegos.

La implementación de funciones clave de Pygame como gestión de gráficos, sonido, eventos y sprites permitira recrear de forma estructurada el clásico PacMan, aplicando buenas prácticas de modularidad, control de recursos y bucles de juego. Este proceso no solo reforzara conceptos teóricos, sino que también proporcionó experiencia práctica en diseño de interfaces interactivas.

Finalmente, el análisis de casos reales de uso y de proyectos destacados desarrollados con Pygame evidencia la versatilidad de la biblioteca, tanto en entornos educativos como en prototipado rápido y simulaciones. Con este ejercicio, se fortalece la capacidad de integrar librerías de Python en aplicaciones concretas, potenciando la creatividad y la eficiencia en futuros desarrollos.

Referencias bibliográficas

- <https://immune.institute/blog/librerias-python-que-son/>
- <https://immune.institute/blog/librerias-python-que-son/>
- <https://wiki.python.org/moin/PyGame>
- <https://www.geeksforgeeks.org/python/pygame-tutorial/>
- <https://pypi.org/project/pygame/>
- <https://www.pygame.org/news>
- <https://www.pygame.org/docs/>
- <https://en.wikipedia.org/wiki/Pygame>
- <https://thepythoncode.com/article/creating-pacman-game-with-python>