



# Introducción

# En este informe se instruirá del cómo se pudo llevar a cabo el proyecto y los códigos solicitados por la tercera cátedra de Programación Orientada a Objetos.

# Será descrito el proceso de creación de un juego RPG con python y ruby, con las solicitudes dadas para su creación, como son la implementación de subclases, un menú de combate y las características básicas de un juego RPG, también mostrando los problemas que surgieron a medida que se creó dicho proyecto.

# Desarrollo del juego

Para este proyecto se utilizaron herramientas como son las librerías random y los pilares fundamentales de P.O.O, (herencia,polimorfismo,encapsulación,abstracción).

**Lenguajes utilizados:** Python, Ruby.

Librería: Random (Hace que las variables sean aleatorias)

**Juego a desarrollar:**

Se desarrolló como un juego en donde caes en un mundo Isekai (muere o desaparece el protagonista de la realidad real, para cumplir otra misión, expiar su culpa o salvar al mundo en otro tiempo o realidad paralela.) y en este caso fue llamado a este nuevo mundo, para destronar al Rey Demonio (Villano).

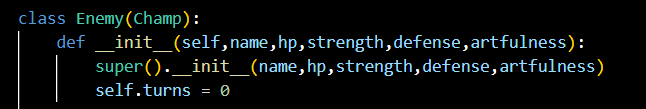
**Problemáticas con Python:** Se tuvo algunos percances al momento de llamar a las funciones, porque no se noto al principio que era más fácil y cómodo usar el encapsulamiento a la hora de llamar a funciones. También al ser un lenguaje interpretado se dificulto la identificación de errores con rapidez, pues en la ejecución se mostraban a medida que se leía el programa en vez de arrojar todos al mismo tiempo.

**Líneas de códigos o parte relevantes del código:**

En la línea n°83 se definen los valores para las estadísticas del enemigo, fue diseñado de forma que en cada encuentro el enemigo tenga estadísticas diferentes de forma que no siempre termine con el mismo resultado.

e = Enemy(name= "Rey demonio",hp = randint(80,150),strength=randint(30,60),defense=randint(10,20),artfulness=randint(10,30))

Aquí se ve una sección del código escrito en python en la que se hace uso de la herencia para crear la clase “Enemigo”:



**En donde utilizó la IA o cuál fue el aporte de esta en su informe o creación de código**

Y así fue como se utilizó:

Se usó [chat.openai.com](https://chat.openai.com) a la hora de transcribir el código a ruby debido a la inexperiencia que se tenía con el lenguaje en cuestión. Además de consultar dudas sobre el uso del lenguaje a la hora de complementar el código.

**Problemáticas con (Ruby):**

Como mayor inconveniente que se puede mencionar, fue en la implementación de los submenús en los menús principales, al momento de llamar un menú dentro de un menú principal, se tuvo que tener “delicadeza” a la hora de “cerrar” los métodos y constructores, por ejemplo para el método “def” o “if”, era necesario cerrarlo correctamente con el valor de retorno que era “end” para finalizar el bloque de cierre.

**Conclusión:**

Se podría decir que al momento de desarrollar la idea del proyecto, con una lluvia de ideas y creaciones de distintos códigos a la par que se avanzó, pudimos tomar en cuenta que el desarrollo formó una gran parte para congeniar la idea final, y pulir de tal manera, la idea que se quería mostrar.

Hablando sobre el uso de los lenguajes, aunque ya se tenía conocimiento de uno de ellos (Python), dio bastantes inconvenientes a la hora de usarlo como un “Orientado a objetos”, por el simple hecho que no se identificaban, que a simple vista no se distinguen, como las comillas,las definiciones de parámetros o simplemente porque no se escribía la opción de mostrar en pantalla. Mencionando que en Ruby también surgían los mismos tipos de errores.

Se destaca igualmente, que fue un trabajo que sirvió para desenvolverse de mejor manera en un lenguaje y un paradigma de la programación específicos.

# Python y características

El primer lenguaje elegido para nuestro grupo fue python, este es un lenguaje interpretado multiparadigma lo que le permite ser capaz de programar en distintos estilos de programación.

**Tipo:**Interpretado

**Paradigmas:** -Orientada a objetos

-Imperativo

-Funcional

-Reflexivo

Este posee un tipado dinámico por lo que una variable es capaz de tomar valores de distintos tipos de datos.

Una característica muy reconocible de este lenguaje es que el contenido de las funciones, clases, bucles, etc, se delimitan según su tabulación o también conocidos como indentación.

# Ruby y características

Como segundo lenguaje tenemos ruby, este es un lenguaje multiparadigma pero mayormente dirigido a un paradigma orientado a objetos, siendo que todos sus tipos de datos son objetos y todas las funciones son métodos.

**Tipo:** Interpretado

**Paradigmas:**-Orientado a objetos

-Reflexivo

Al igual que en python este lenguaje posee un tipado dinámico siendo capaz de tomar una variable con diferentes tipos de datos. También es Open Source y multiplataforma, se descarga gratuitamente desde la página oficial y se puede ejecutar en diferentes sistemas operativos.

Sigue el “principio de la menor sorpresa”, lo que significa que el lenguaje debe comportarse de tal manera que minimice la confusión de los usuarios experimentados, simplificando el trabajo de programación. Por lo que se dice que es de alto nivel, al a veces parecer que simplemente se está escribiendo en inglés y estar más lejano al lenguaje de máquina.