Testing

Presentado por:

Sergio Tovar Vasquez

Emiliano Guerra

Julian David Murillo Rodriguez

Profesor: Oscar Eduardo Alvarez Rodriguez

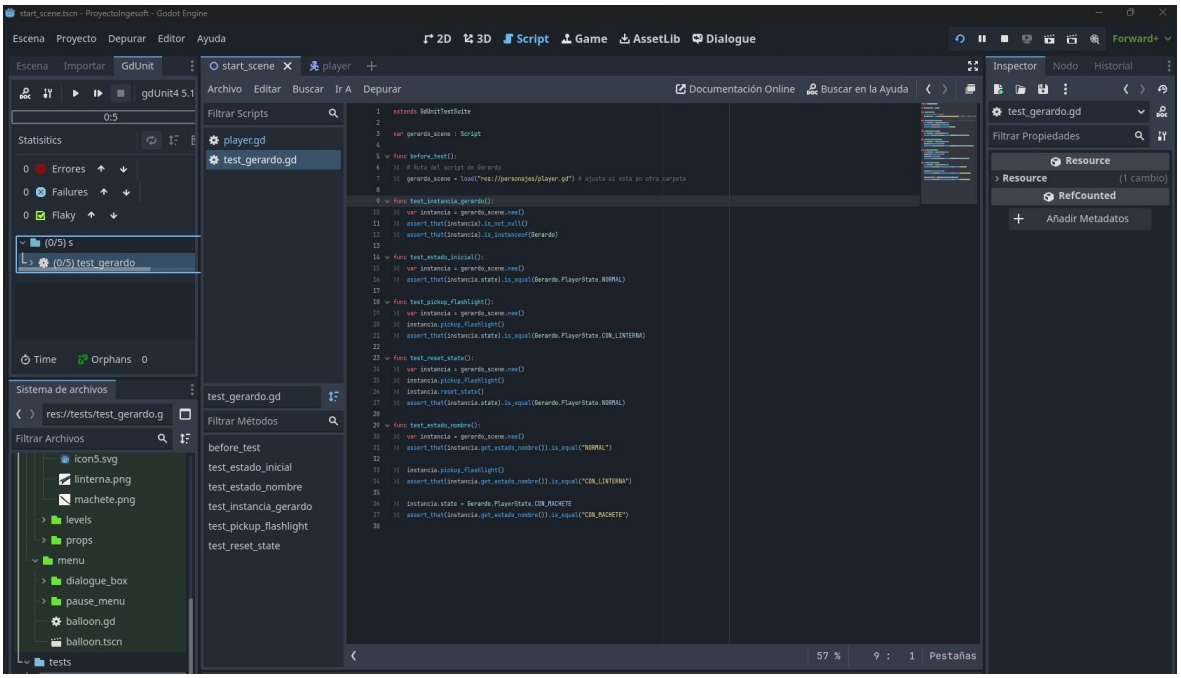
Universidad Nacional de Colombia - sede Bogotá

Facultad de Ingeniería

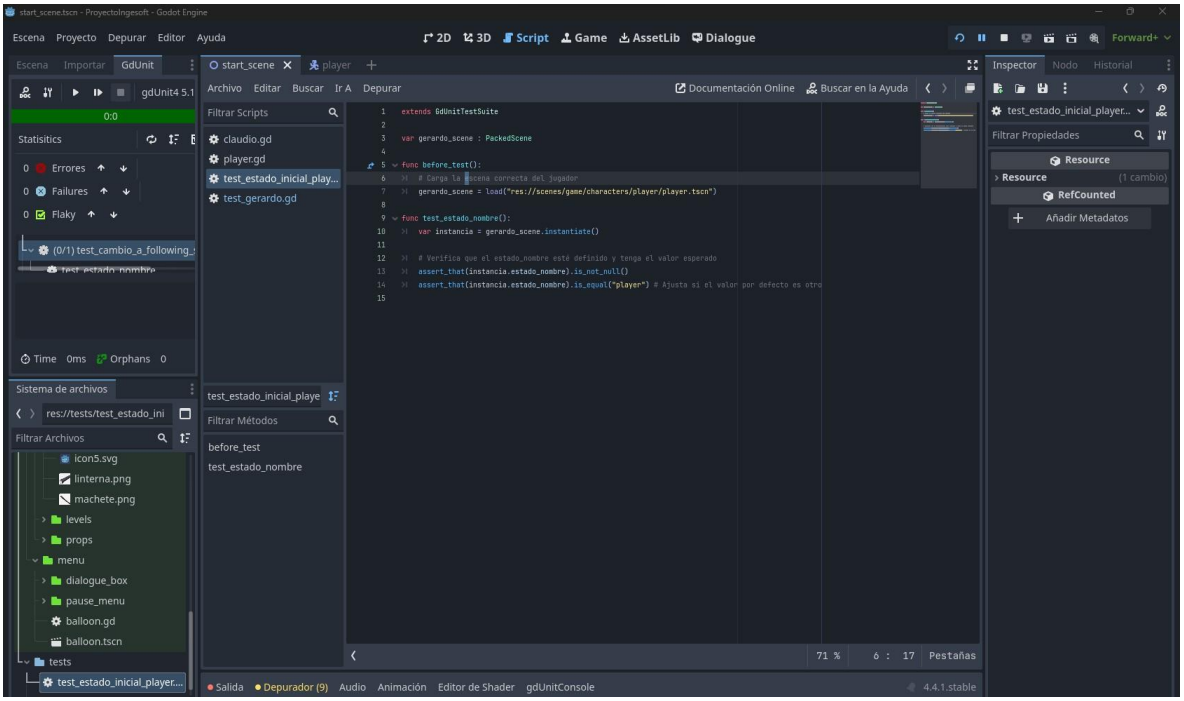
Curso: Ingeniería de Software

Pruebas

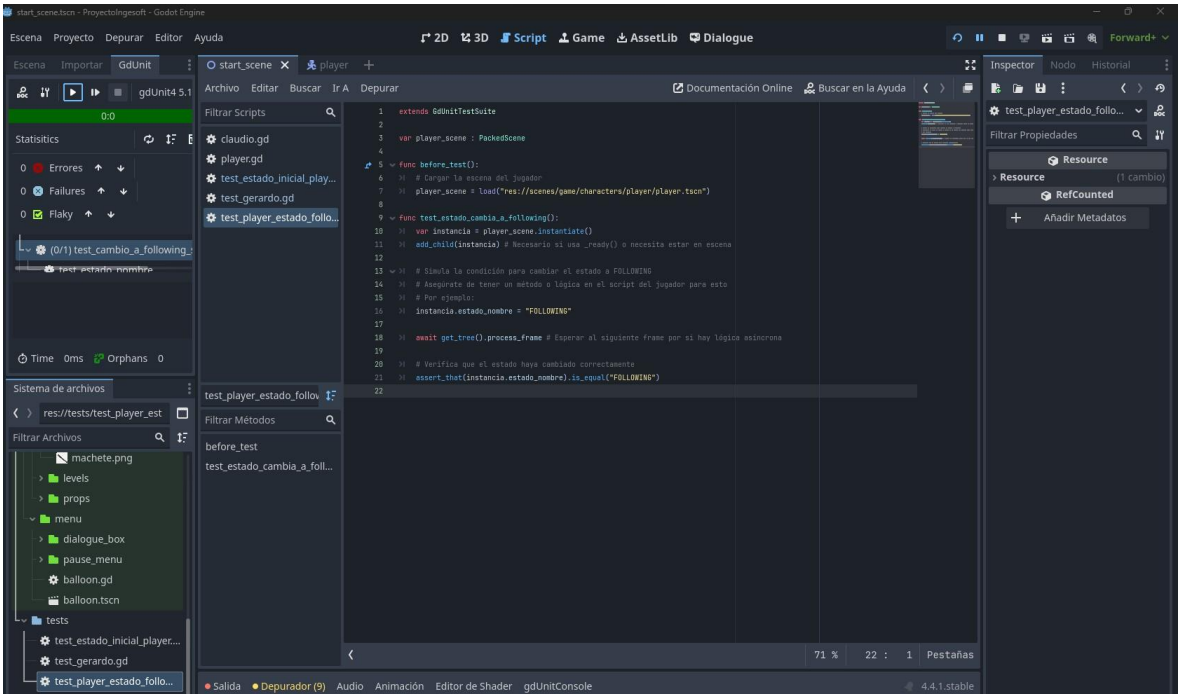
La prueba en el script está verificando cómo un objeto reacciona al cambiar entre diferentes estados o situaciones. Básicamente, se asegura de que el objeto funcione de la manera esperada cuando se le aplican ciertos cambios o interacciones.



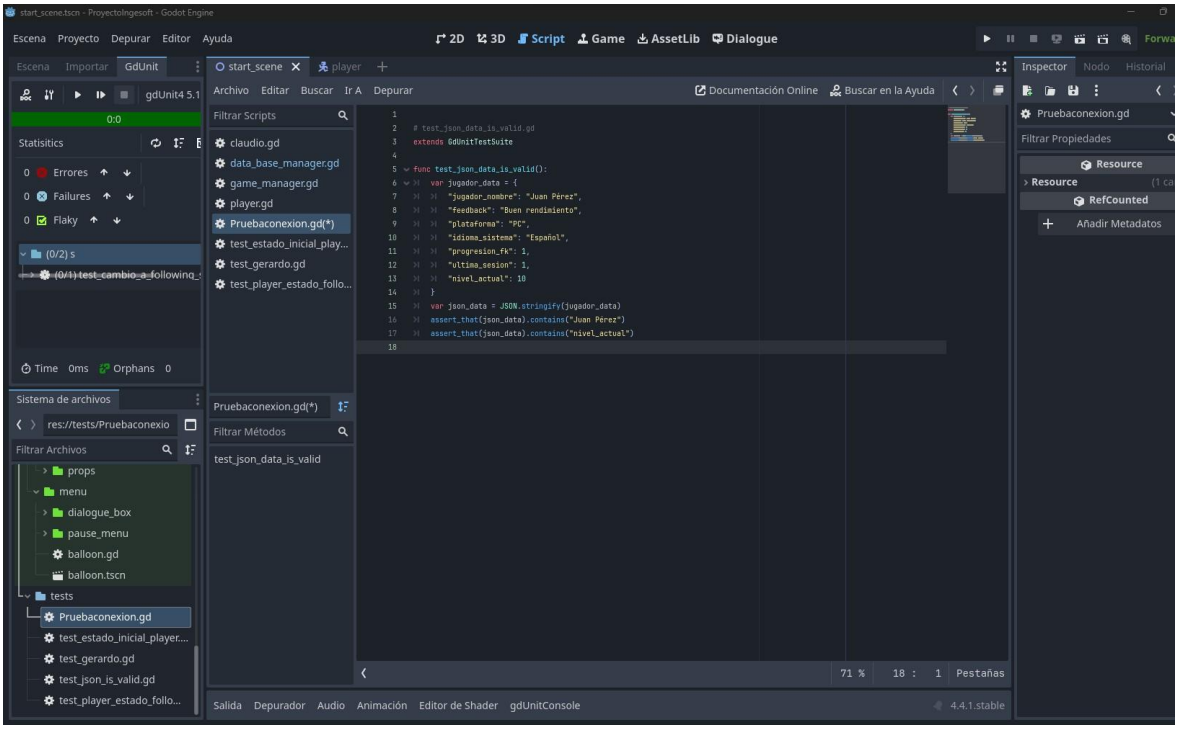
Se verifico que la propiedad estado nombre esté definida y que nombre: estado_nombre tenga como valor "player" al inicio



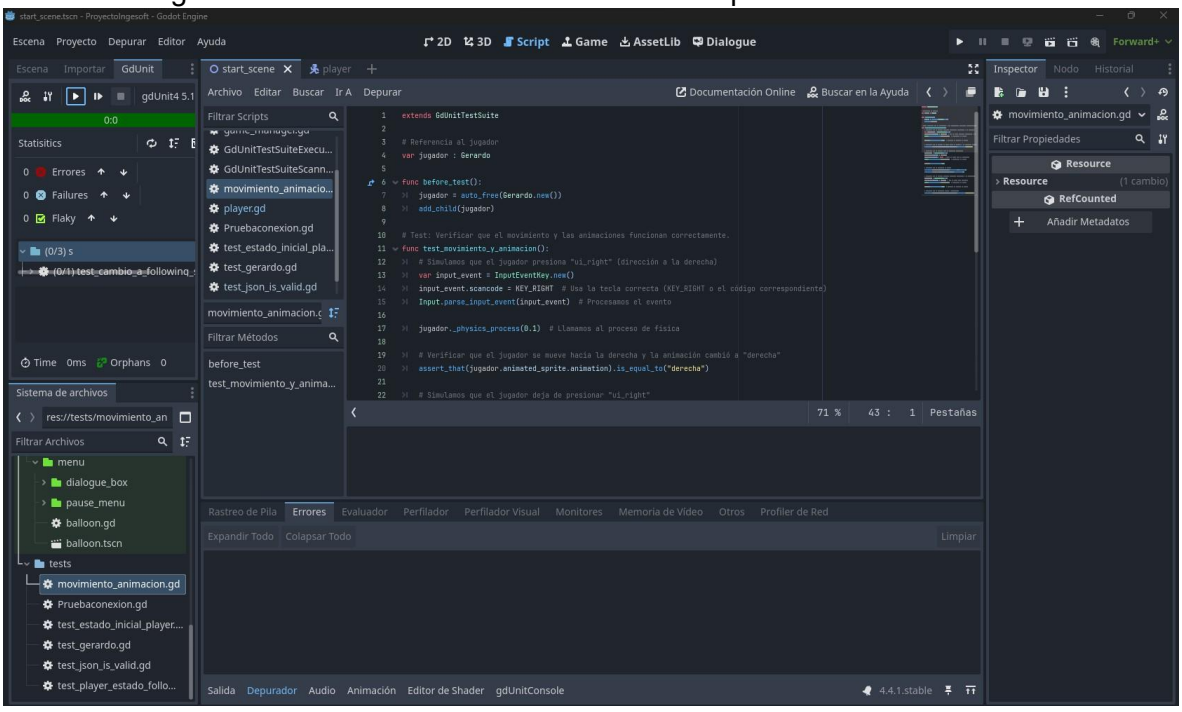
En esta prueba se comprueba que el jugador cambia su estado a “FOLLOWING” como debería cuando se le indica.



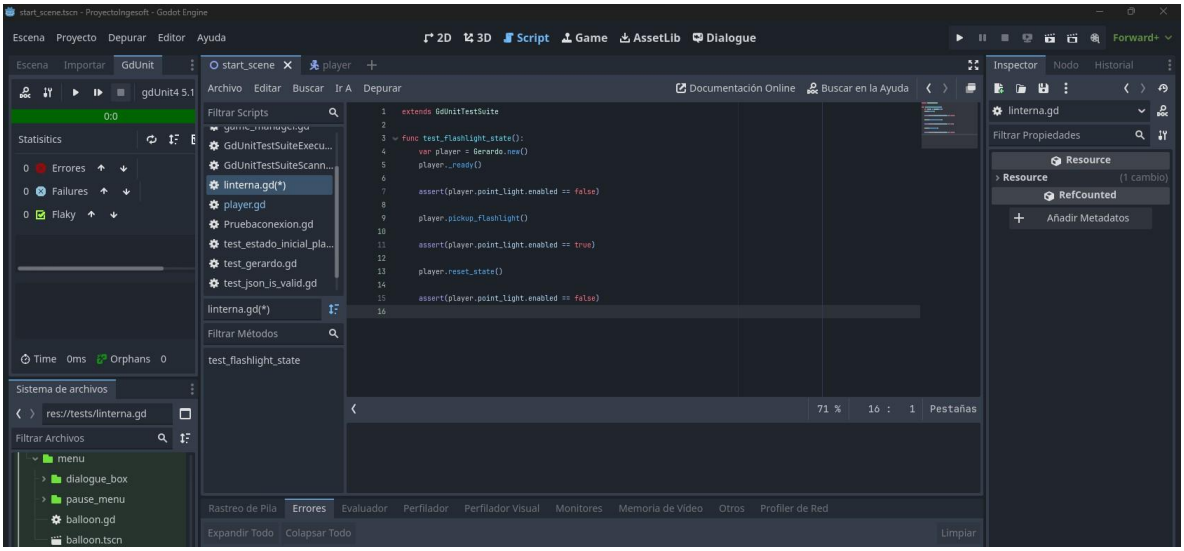
La prueba comprueba que los datos del jugador se convierten correctamente a JSON y que incluyen campos importantes como el nombre y el nivel actual.



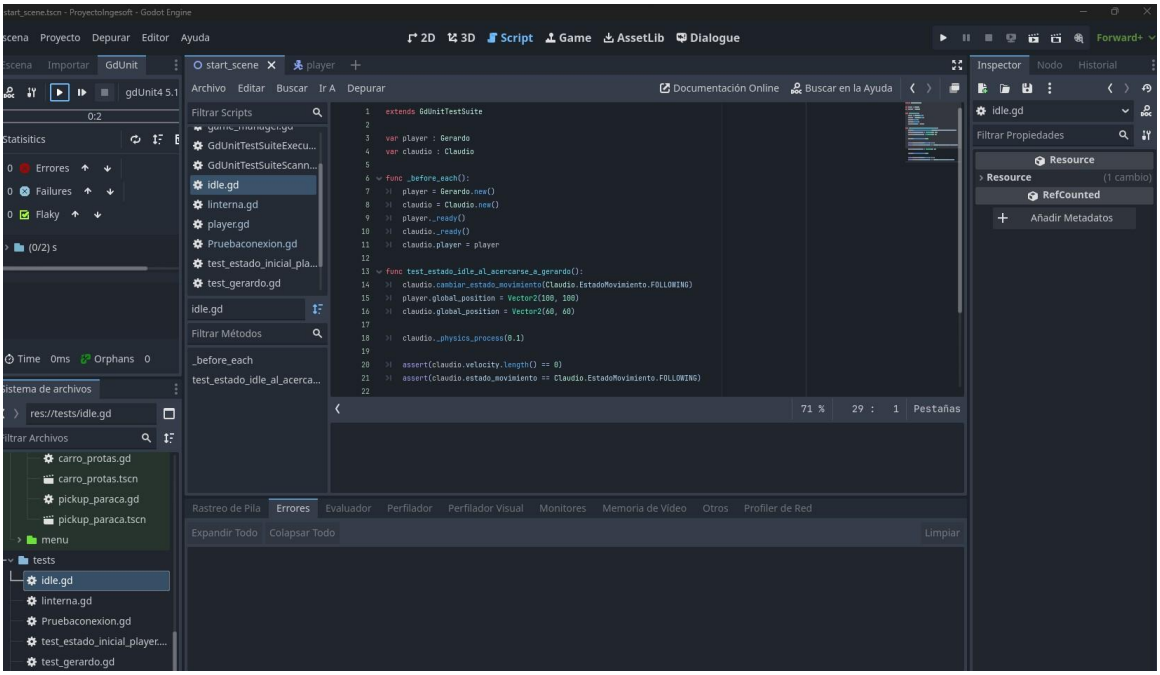
La prueba consiste en verificar que el personaje se mueva correctamente en todas las direcciones y que, al hacerlo, las animaciones del *animated_sprite* cambien de forma adecuada según la dirección del movimiento cuando se presionan las teclas de dirección.



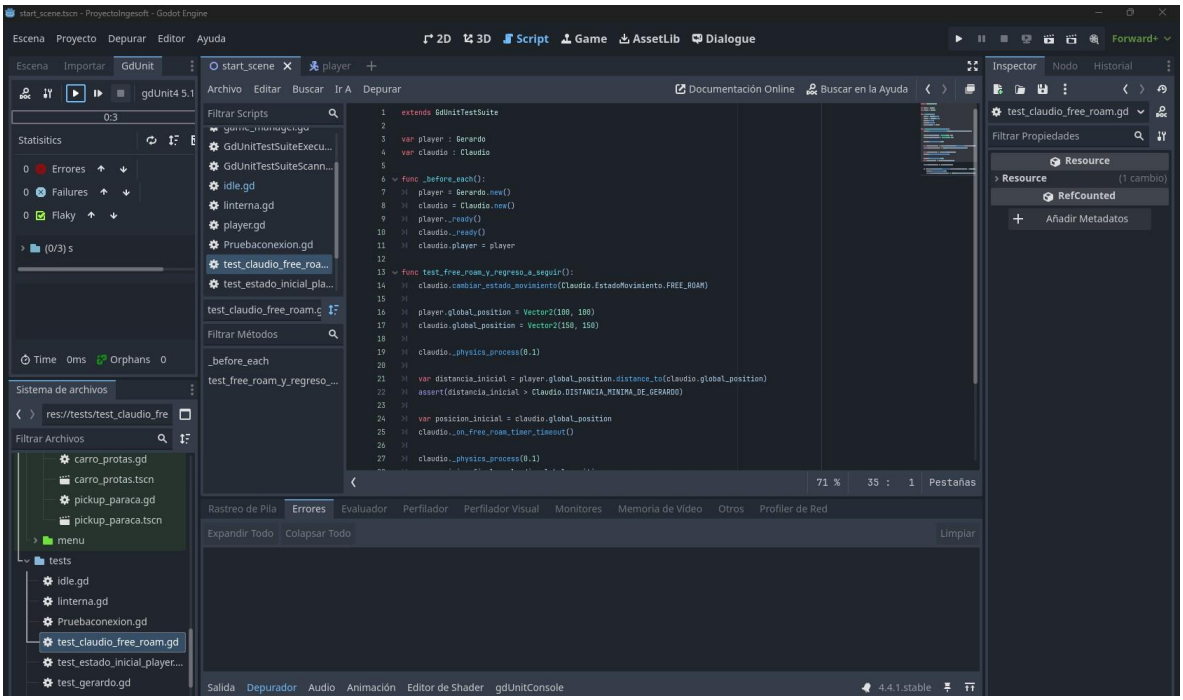
Se probó si la linterna del jugador está encendida y en un estado válido.



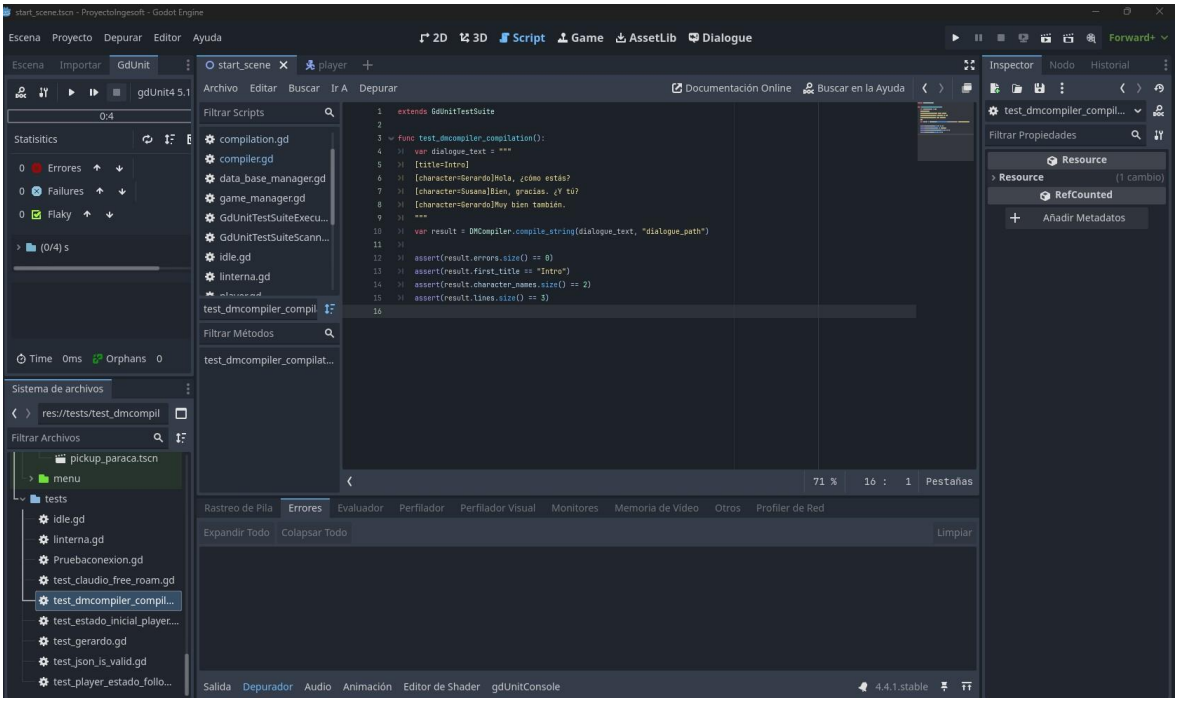
La prueba verifica que *Claudio* cambie correctamente de estado de movimiento a **IDLE** al acercarse a *Gerardo*. Primero, se asegura que *Claudio* sigue a *Gerardo* sin detenerse. Luego, al acercarse lo suficiente, se confirma que su velocidad es 0 y su estado cambia a *IDLE*.



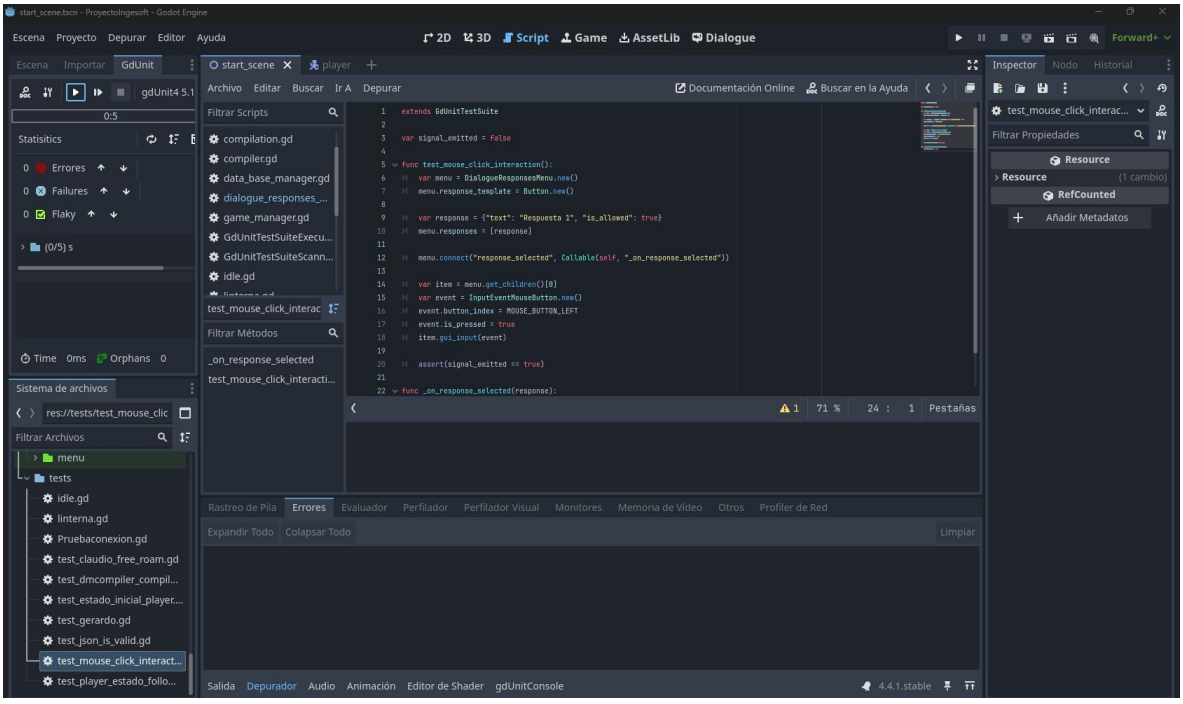
La prueba asegura que *Claudio* puede moverse libremente cuando está en estado FREE_ROAM, manteniendo una distancia mínima de *Gerardo*. Al finalizar el tiempo del "free roam", se verifica que *Claudio* regrese a seguir a *Gerardo* si ha superado la distancia máxima de movimiento libre, cambiando correctamente su estado a FOLLOWING**.**



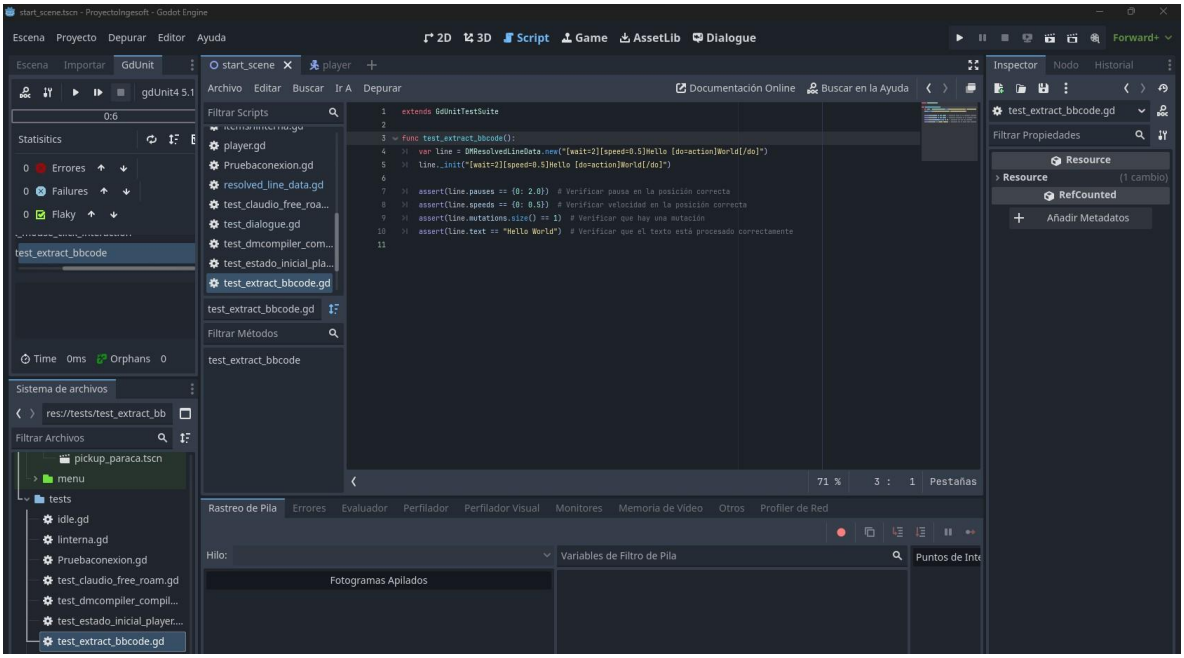
La prueba valida que el compilador de diálogo (DMCompiler) procesa correctamente un texto de diálogo. Verifica que no haya errores en el proceso, que se haya identificado correctamente el título, que se hayan detectado los nombres de los personajes y que se haya generado el número adecuado de líneas de diálogo.



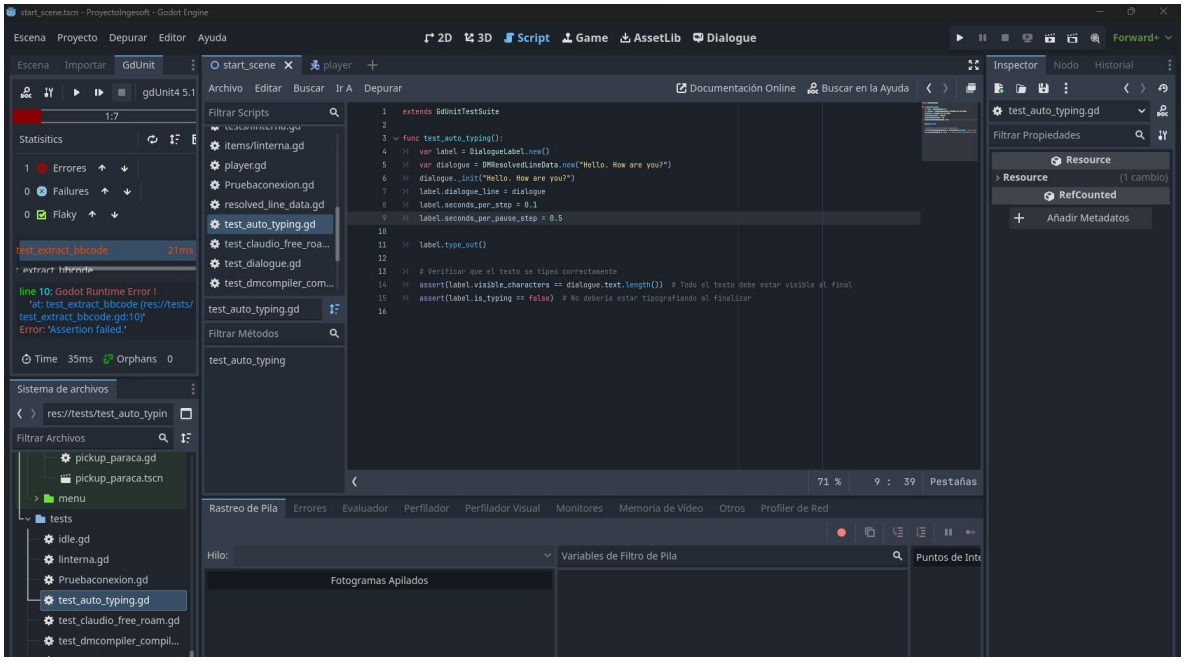
Este test verifica que, al hacer clic en un botón de diálogo, se emite correctamente la señal que indica que una respuesta fue seleccionada. Simula el clic del jugador y comprueba que el juego reacciona como debería.



En esta prueba se verifica que el sistema entienda bien las etiquetas especiales en un texto de diálogo, aplicando pausas, velocidad y acciones, y que el texto final quede limpio.



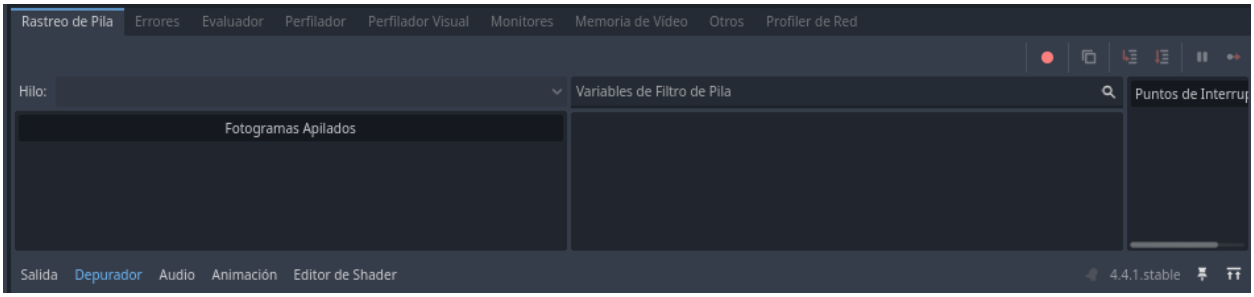
En esta prueba se verifica que el sistema entienda bien las etiquetas especiales en un texto de diálogo, aplicando pausas, velocidad y acciones, y que el texto final quede limpio.

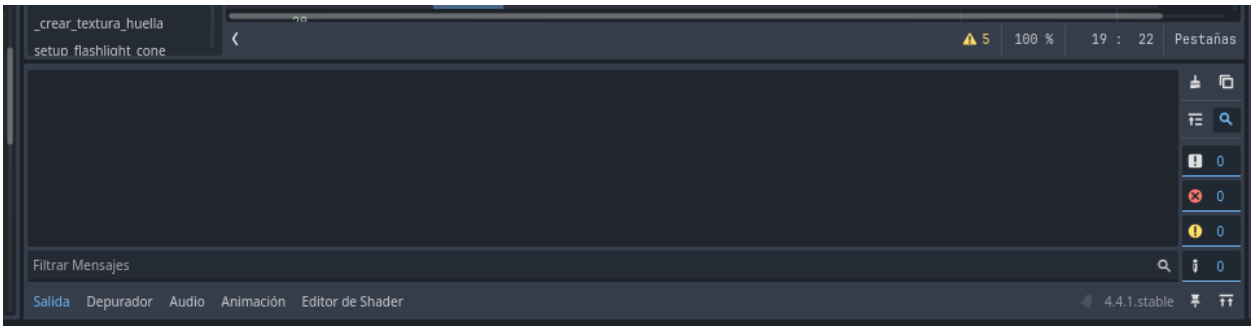


Profe por ultimo Queríamos explicarle brevemente los desafíos que encontramos al intentar implementar un análisis estático de código en nuestro proyecto de Godot. A pesar de que intentamos integrar un linter específico para Godot, encontramos dificultades técnicas debido a la falta de compatibilidad del paquete que queríamos utilizar. Tras varios intentos, no logramos ejecutar la herramienta de manera eficiente, ya que no estaba disponible en el índice oficial de pip y las alternativas no eran fáciles de implementar en el entorno de Godot.

A pesar de estos contratiempos, decidimos proceder de otra manera para garantizar que el código se

mantuviera de acuerdo con buenas prácticas. Utilizamos las herramientas de depuración integradas en Godot y realizamos revisiones manuales del código, lo cual nos permitió asegurar que el código cumpliera con los estándares requeridos. Afortunadamente, conseguimos los resultados deseados y los entregamos sin problemas, aunque de manera diferente a la inicialmente planteada.





Básicamente utilizamos la salida y depurador ya integrado en el mismo Godot para realizar pruebas y verificar que nuestro proyecto no manejara errores tal como se muestra, si tiene algún par de advertencias que ya tenemos en cuenta.