

Informe Trabajo Parcial

**Profesor:**

Leo Ramirez Argume

**Sección:**

WS7A

**Integrantes:**

Mejía, Luis

Plácido, Fabrizio

Urbina, Alexander

Uribe, Braulio

2020-01

1. Descripción del alcance de la solución

El proyecto con el nombre “Mezio” es una página web la cual tiene como meta brindar asistencia a los usuarios con el diseño mobiliario. Aparte de mostrar dónde comprar los muebles que son parte del diseño.

Para conseguir este propósito, tendremos que asegurarnos de que el sistema de diseño funcione como debería. Las pruebas del software tienen una gran importancia para asegurarnos y brindar confianza de que la solución está funcionando en el modo correcto.

1. Reglas de negocio

**RN01**:Los muebles no deben superponerse en el plano.

**RN02**:El usuario no podrá añadir un nuevo mueble si se supera el 75% de espacio ocupado de la plantilla.

**RN03**: En caso de que el usuario no muestre ninguna actividad en el modelo 2d por más de 30 minutos, se le cerrará la sesión.

**RN04**: El usuario debe contar con una cuenta para poder acceder al sitio web.

**RN05**: El usuario solo puede trabajar en un proyecto a la vez.

**RN06**: Solo se puede usar una plantilla por proyecto.

**RN07:** Las plantillas que usarán en cada proyecto son modificables.

**RN08:** El usuario podrá insertar un solo mueble a la vez en el plano.

**RN09**: El usuario solo podrá confirmar el paquete si ha agregado por lo menos un producto.

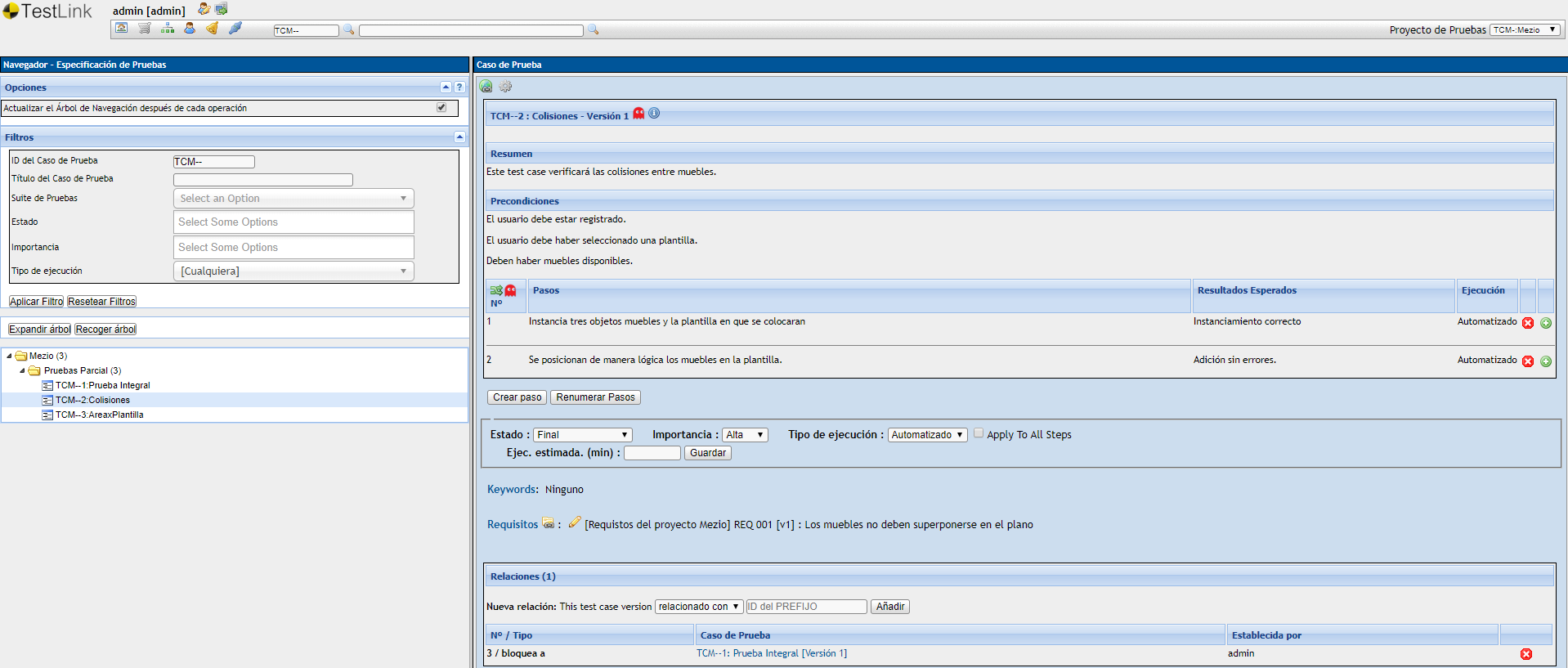
**RN10:** El sistema siempre mostrará muebles que presenten stock.

**RN11:** El sistema recomendará muebles con características predominantes en la mayoría de los muebles que haya agregado.

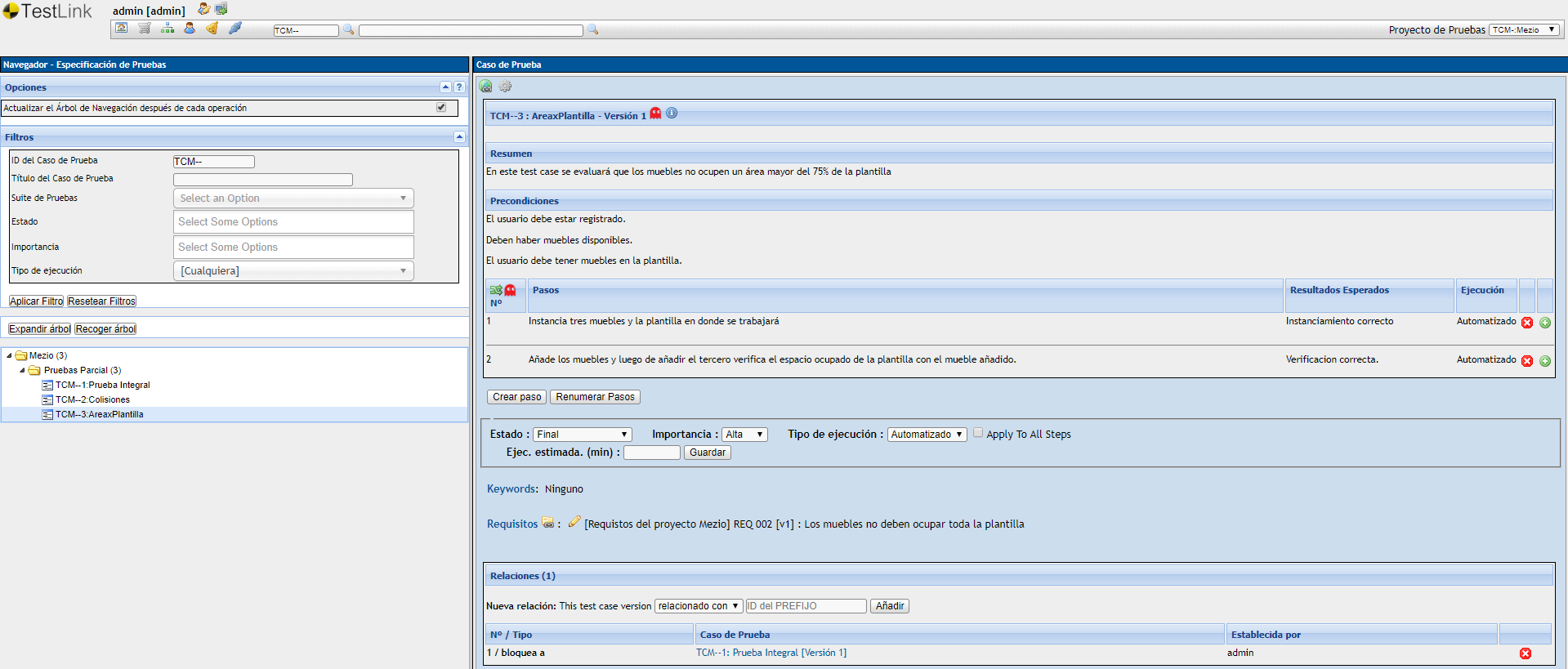
**RN12:** No se podrá generar un reporte a menos que el usuario haya guardado un paquete.

1. Testlink

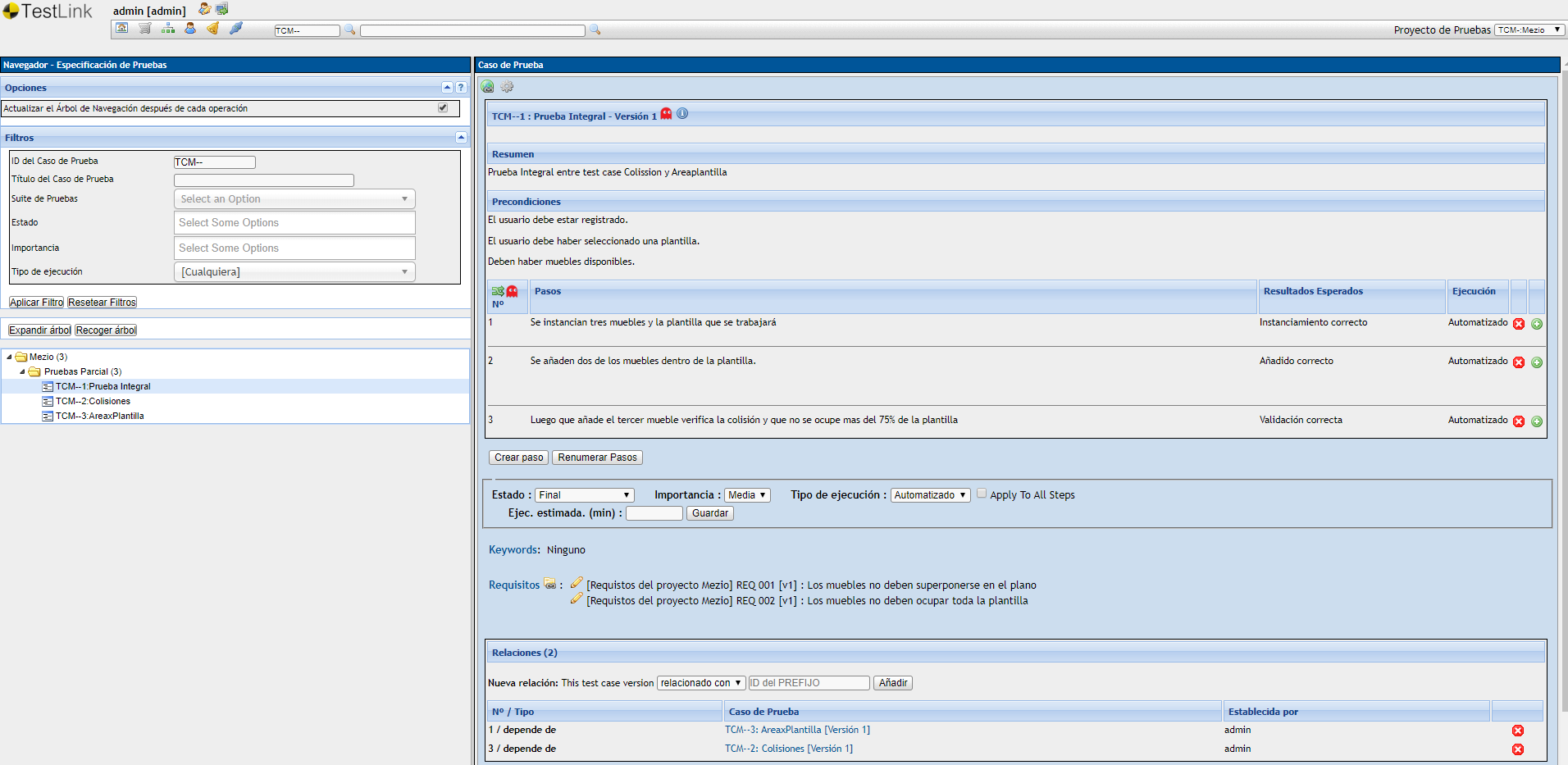
* Prueba Unitaria Colisiones



* Prueba Unitaria AreaxPlantilla



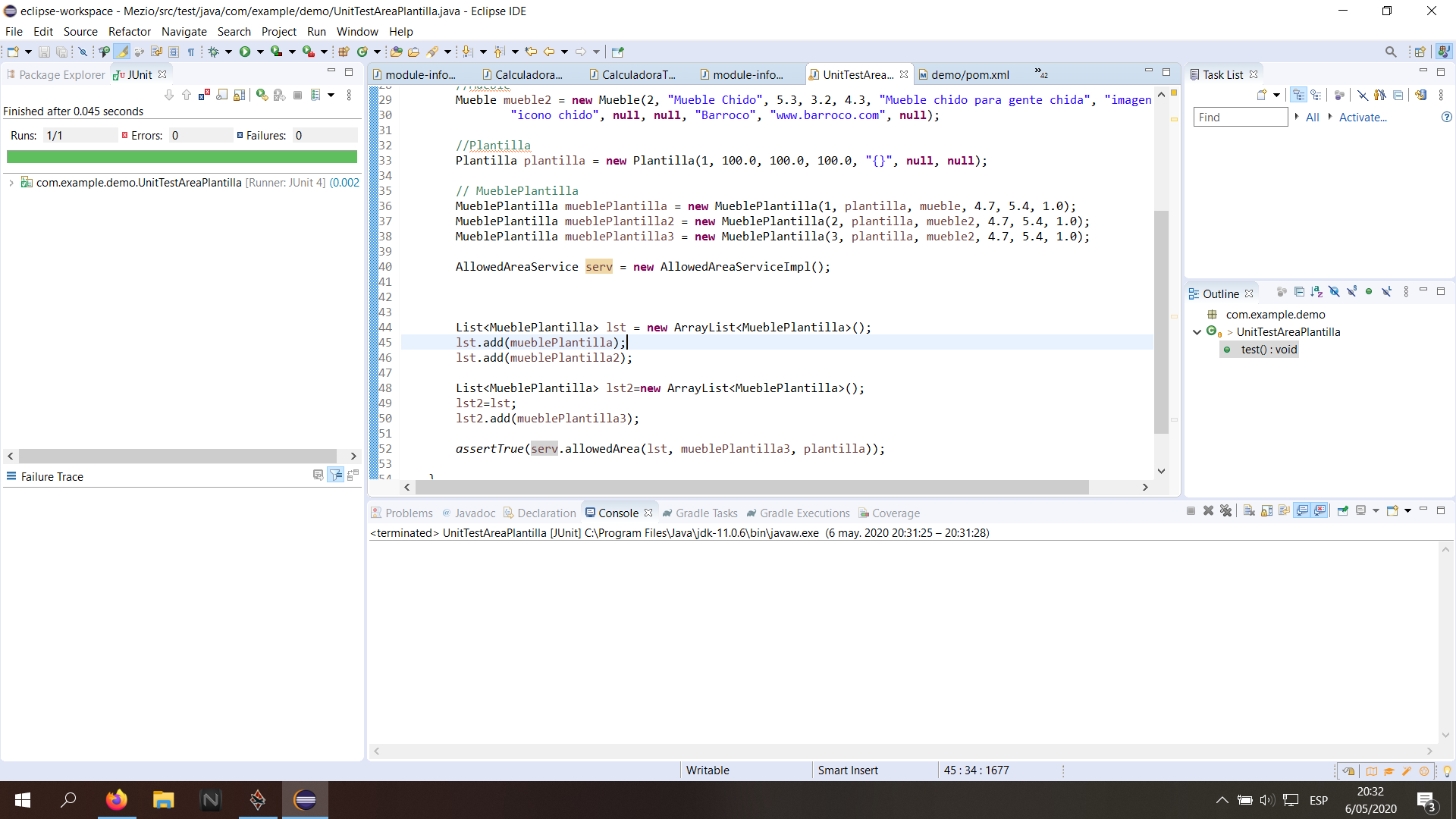
* Prueba Integral

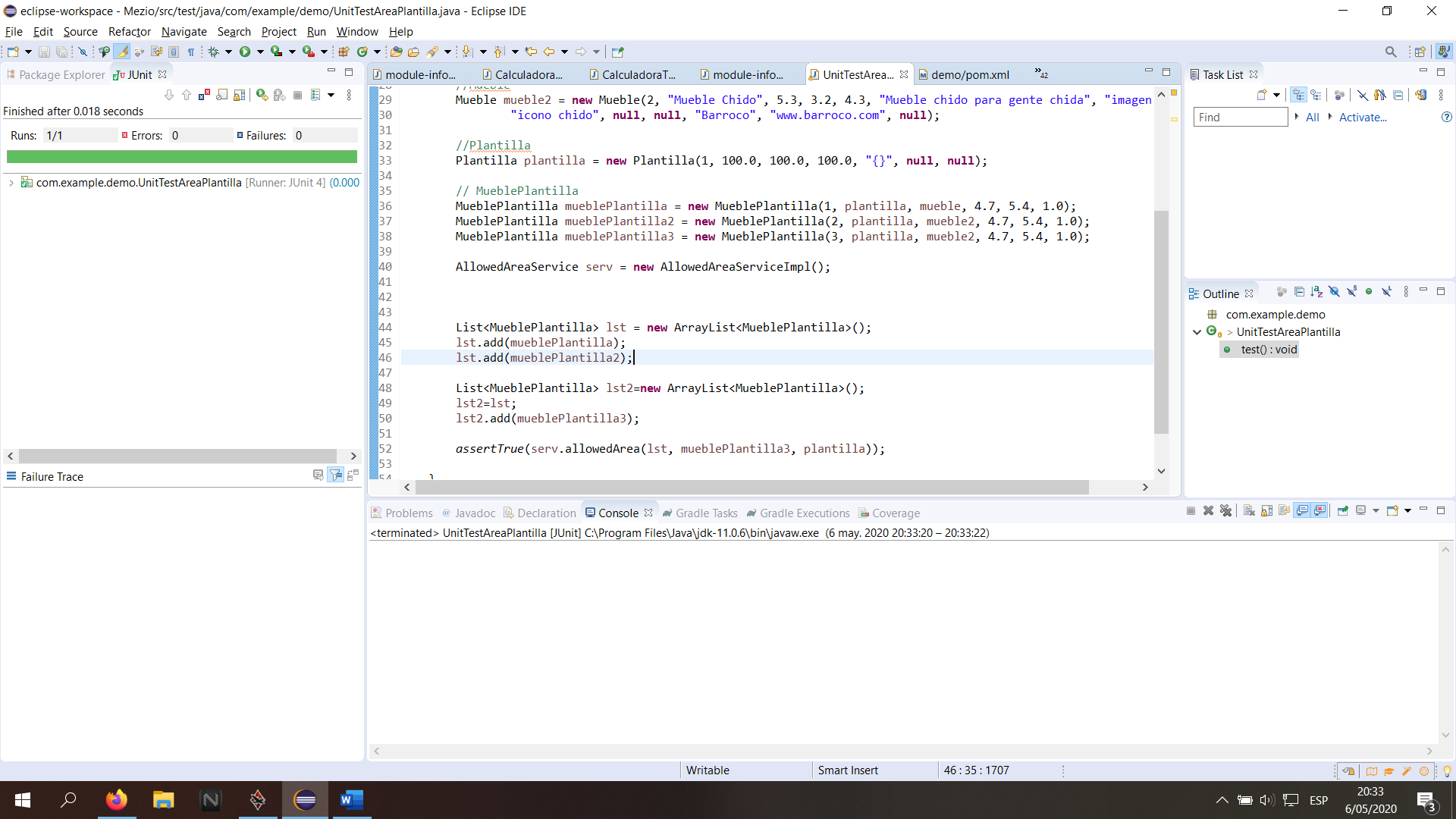


1. Pruebas Unitarias

Errores reportados:

* Se tuvo que reestructurar el proyecto para la aplicación de pruebas unitarias.
* Existieron problemas debido a que se usaba una comparación ‘==’ en vez de .Equals() dado el uso de Double.
* Se tuvo complicaciones para identificar las matches entre servicios y reglas de negocio.





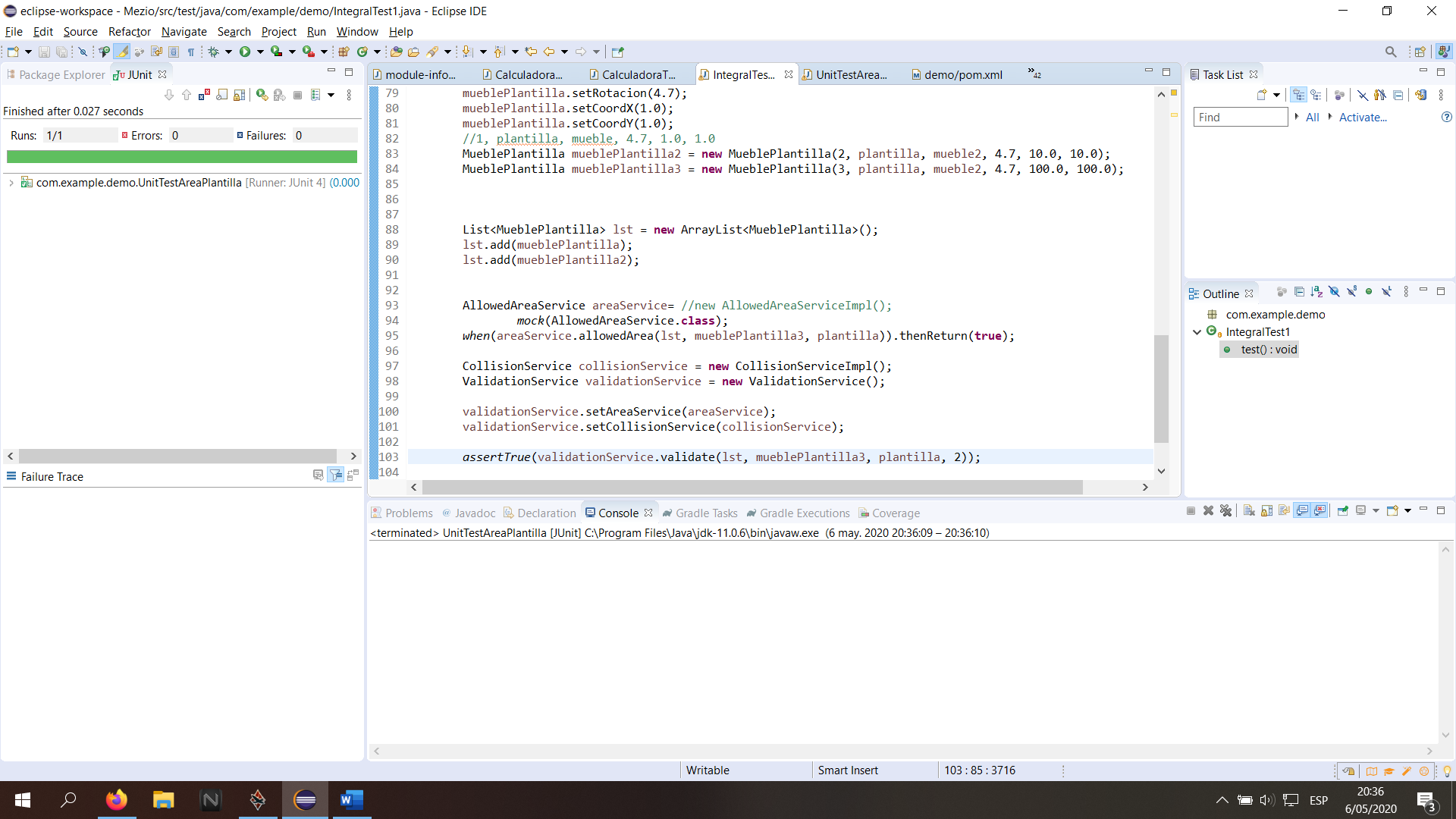
1. Pruebas Integrales

Planteamiento con Mockito:

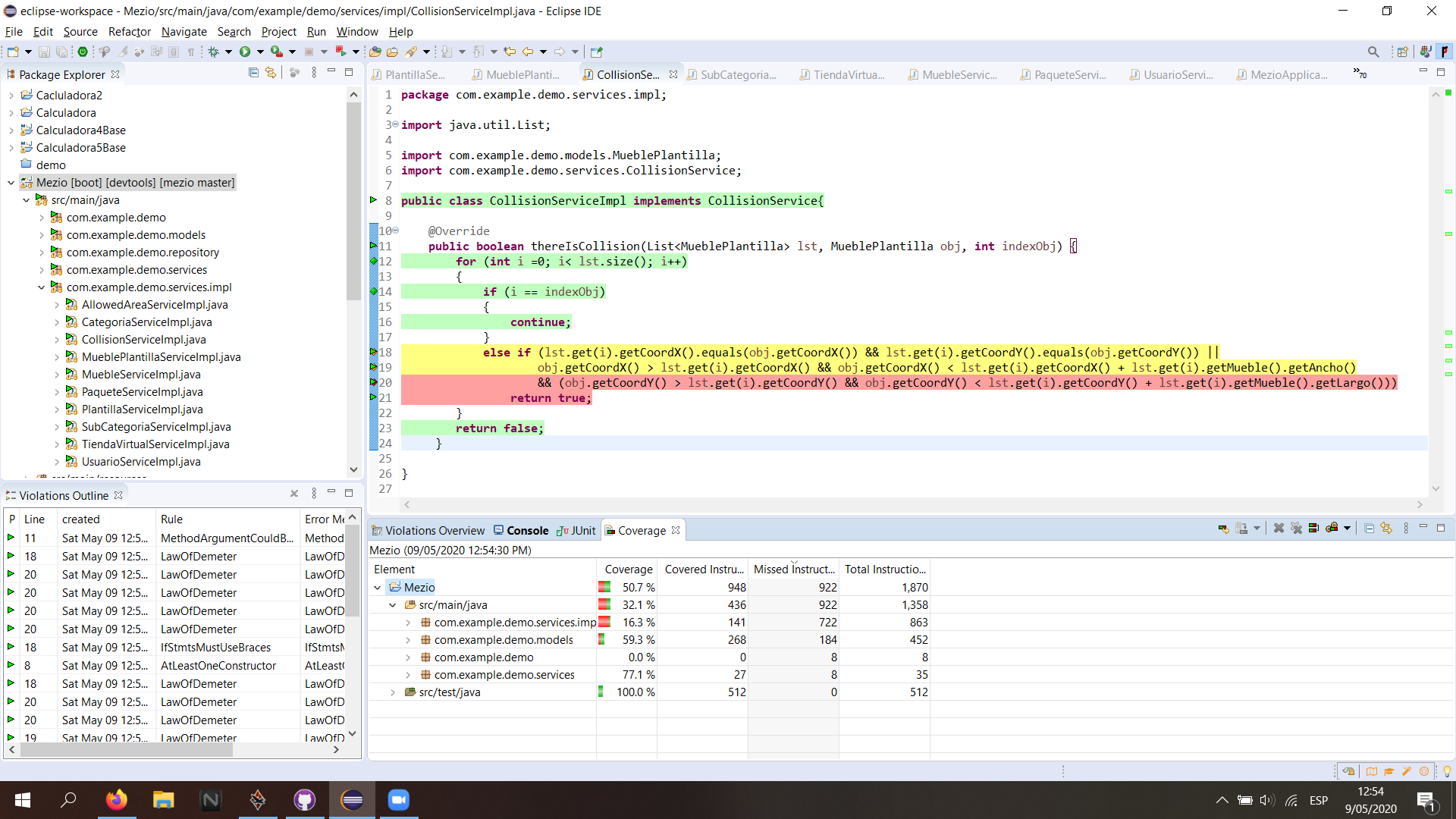
* Se creó un service Validation que integra los services AllowedArea y Collision, sobre el cual se realizó el Test Integral
* Se utilizó Mockito con el Service AllowedArea de ValidationService para que devuelva verdadero cuando se inserte el mueble3 en la plantilla, aunque este exceda el área para muebles permitida.
* La prueba integral resultó exitosa debido al Mockito, pues si se usarán condiciones normales esta terminaría en failure.

Problemas encontrados:

* Debido a la estructuración de las pruebas unitarias, se tuvo que tomar tiempo para reestructurar y que sea posible la integración.
* En un principio se planteó que las funcionalidades solo se encontraran en un service, complicando el uso de Mockito.

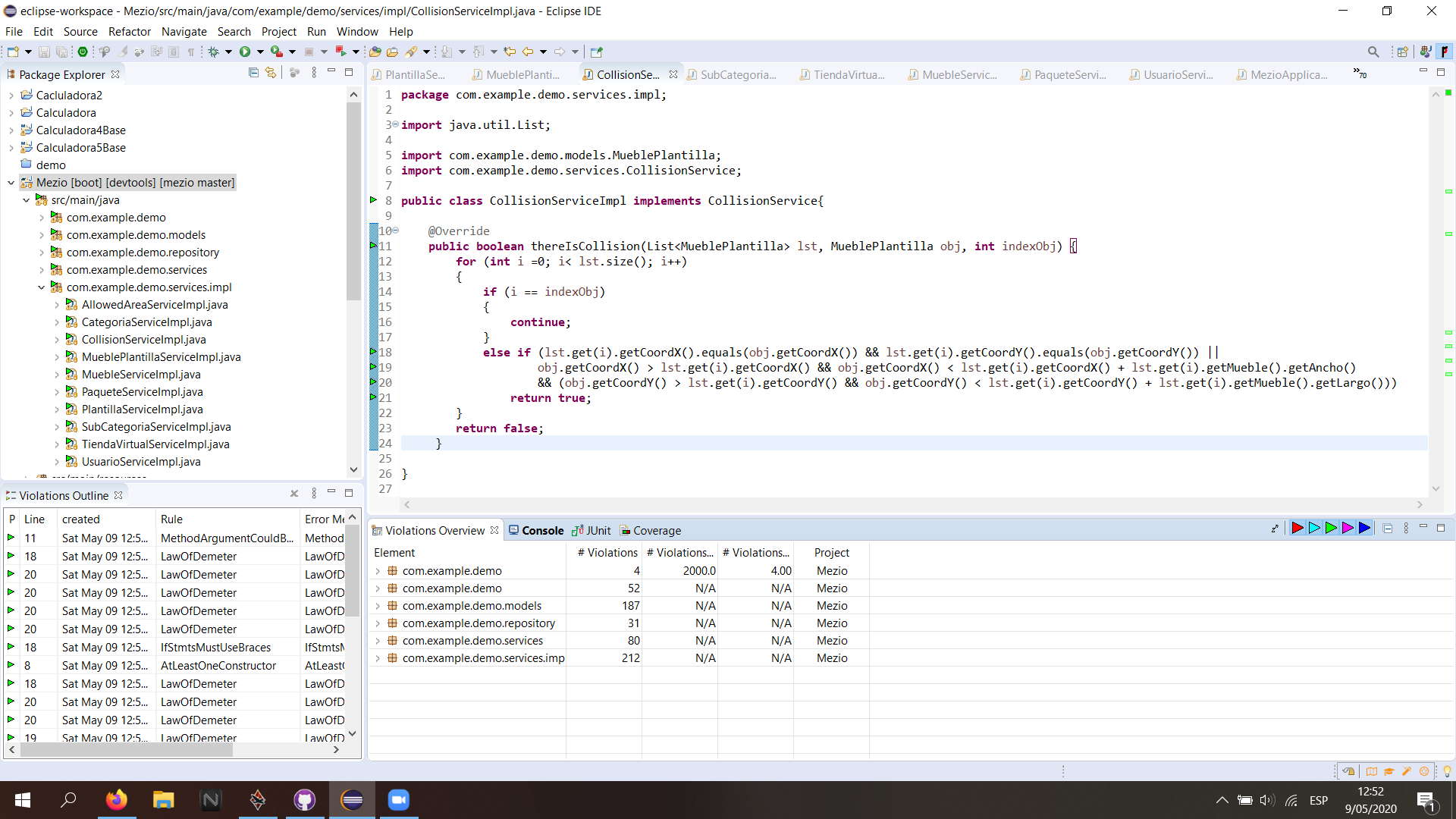


1. Análisis de cobertura



* Se obtuvo un 32.1% de coverage sobre el código, el cual consideramos suficiente pues nuestras pruebas se realizan sobre el core del negocio, la creación de la plantilla de muebles. Por ello, el coverage de este código tiene más peso sobre la aplicación. Sin embargo, se ha tomado en consideración y se aplicarán pruebas que sustenten la implementación de los demás servicios.

1. Auditoría sobre el código



* Se encontró un número alto de violaciones en el código, lo que nos llevó a revisar el contorno de ellas. Luego de revisar las recomendaciones, el grupo encontró que las violaciones no eran de la gravedad que se pensó. Se consideró que gran parte de ellos se debió a que los integrantes no cuentan con un amplio conocimiento de Java. Sin embargo, se determinó a intentar levantar revisiones del código para futuras iteraciones.