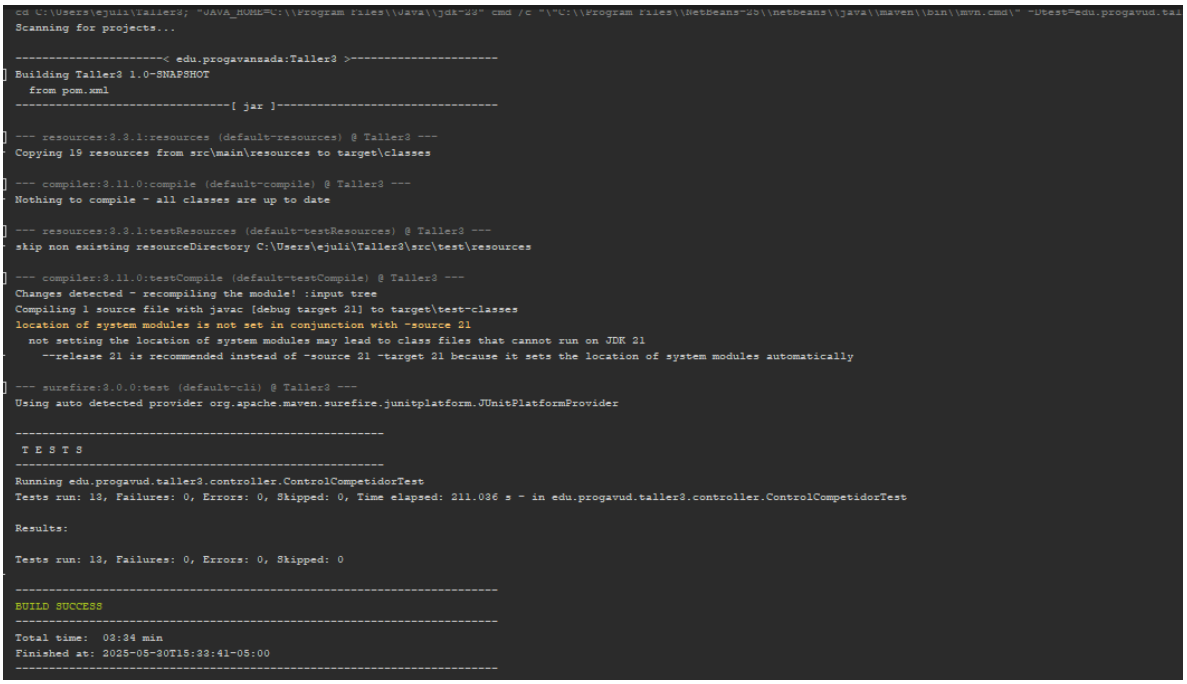
Decidimos realizarle pruebas a la clase ControlCompetidorTest

Evidencia del Dataset con las descripciones de cada parte, la entrada y el resultado esperado

El siguiente análisis presenta los resultados de las pruebas unitarias diseñadas específicamente para evaluar los métodos de la clase ControlCarrera. Cada caso de prueba contempla:

- **Descripción del Test**: Indica el método de ControlCarrera que está siendo evaluado.
- **Entrada**: Valores concretos utilizados para la ejecución del método.
- **Resultado Esperado**: Comportamiento o salida que se espera del sistema una vez ejecutada la prueba.

Imagen 1: Consola de construcción y prueba

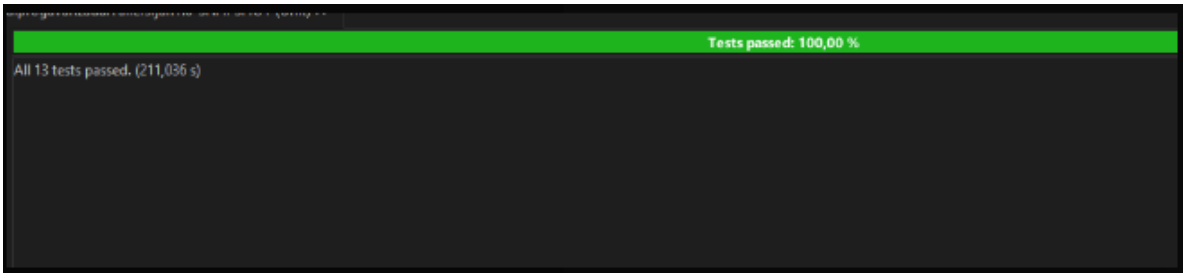


En la consola se observa que:

- La **compilación del proyecto fue exitosa** (BUILD SUCCESS).
- Se **ejecutaron 13 pruebas unitarias** correspondientes a los distintos métodos de ControlCarreraTest.

- **Todas las pruebas pasaron exitosamente**, sin errores ni fallos (Failures: 0, Errors: 0).
- El **tiempo total de ejecución** fue de aproximadamente 2.24min
- Se incluye la **fecha y hora exacta** de ejecución, lo cual garantiza la trazabilidad del proceso de validación.

Imagen 2: Panel visual de resultados de pruebas



En la pestaña **Test Results** del IDE se confirma gráficamente que **el 100% de las pruebas fueron superadas con éxito** (13/13). Entre los métodos validados se encuentran:

- CrearCompetidor()
- GetCompetidor()
- DefinirGanadorUnicoGanador()
- DefinirGanadorConEmpate()
- DefinirGanadorMejorTiempo()
- DefinirGanadorAbsolutoUnico()
- DefinirGanadorAbsolutoConEmpate()
- DefinirGanadorAbsolutoSinVictorias()
- GetNumeroCompetidores()
- DefinirGanadorTiemposCero()
- OrdenCompetidores()
- FormatoTiempoEnSegundos()

En esta clase de prueba se aprovechan las capacidades de **JUnit 5** para organizar la configuración y garantizar que cada escenario de prueba parta de un estado

controlado y reproducible. A continuación, se describen las principales anotaciones usadas:

@BeforeEach

El método setUp() anotado con @BeforeEach prepara un nuevo objeto de ControlCompetidor antes de cada prueba. Para ello:

1. Se instancia un nuevo objeto ControlCompetidor.
2. Se agregan varios objetos de tipo Competidor con atributos configurados manualmente (nombres, tiempos y victorias).
3. Se asegura que cada prueba comience desde un entorno limpio, sin interferencia de pruebas anteriores.

Ausencia de @BeforeAll y @AfterAll

No se utilizan estas anotaciones debido a que las pruebas no requieren recursos compartidos o configuraciones globales. Cada prueba funciona de forma aislada y no depende de configuraciones persistentes.

Ausencia de @AfterEach

No es necesario realizar tareas de limpieza adicionales, ya que las instancias creadas se destruyen al finalizar cada método de prueba, liberadas automáticamente por el recolector de basura. Además, no se utilizan recursos como archivos, hilos o conexiones externas que requieran cierre manual.