**Casos de prueba**

Escenario 1: Se cargan los primeros 18 registros de artículos y todos los datos de ventas.

Escenario 2: Se agregan 6 artículos a una lista de artículos, los ID van del 1 al 6 y los nombres van de la “A” a la “F”.

Escenario 3: Se agregan 8 artículos a una lista de artículos, los ID van del 1 al 8 y los nombres van de la “A” a la “H”.

Escenario 4: Se agregan 4 artículos a una lista de artículos, los ID van del 1 al 4 y los nombres van de la “A” a la “D”.

Escenario 5: Se agregan 9 artículos a una lista de artículos, los ID van del 1 al 9 y los nombres van de la “A” a la “I”.

Objetivo prueba 1: Probar que se agregan todos los artículos y ventas de los archivos de Excel de artículos y ventas, salvo los registros dañados.

|  |  |
| --- | --- |
| Clase | FuercitaBruta |
| Método | CargarDatos() |
| Escenario | Escenario1 |
| Valores de entrada | Todos los registros |
| Resultado | Se agregaron 18 artículos y 149299 ventas |

Objetivo prueba 2: Probar que se agregan los artículos y las ventas y que sus valores correspondan según el orden en el que están en el archivo de Excel.

|  |  |
| --- | --- |
| Clase | FuercitaBruta |
| Método | CargarDatos() |
| Escenario | Escenario1 |
| Valores de entrada | Todos los registros |
| Resultado | El valor del artículo en la posición 10 de la lista de artículos del programa corresponde al del archivo de artículos de Excel y el valor de la venta en la posición 17 corresponde al de la venta de Excel. |

Objetivo prueba 3: Probar que el número de combinaciones que hace el método Combinations corresponde al valor respectivo de la regla de combinación (A**C**B)

|  |  |
| --- | --- |
| Clase | FuercitaBruta |
| Método | Combinations(IEnumerable<T>, int) |
| Escenario | Escenario2 |
| Valores de entrada | Los 6 artículos del escenario y sus combinaciones de 2 |
| Resultado | 15, resultado de 6**C**2 |

Objetivo prueba 4: Probar con una cantidad mayor de artículos las combinaciones para determinar si se están realizando de forma correcta.

|  |  |
| --- | --- |
| Clase | FuercitaBruta |
| Método | Combinations(IEnumerable<T>, int) |
| Escenario | Escenario3 |
| Valores de entrada | Los 8 artículos del escenario y sus combinaciones de 3 |
| Resultado | La primera combinación es {A, B, C} y la última combinación es {F, G, H} correspondiente al orden de combinación de elementos del algoritmo. |

Objetivo prueba 5: Probar con una cantidad mínima el método de ConjuntoPotencia y verificar que sus primeros conjuntos de elementos y el ultimo correspondan a la regla del algoritmo.

|  |  |
| --- | --- |
| Clase | FuercitaBruta |
| Método | ConjuntoPotencia(List<Articulo>) |
| Escenario | Escenario4 |
| Valores de entrada | Los 4 artículos del escenario |
| Resultado | Primer conjunto {}  Segundo conjunto {A}  Tercer conjunto {B}  Cuarto Conjunto {A, B}  Ultimo conjunto {A, B, C, D} |

Objetivo prueba 6: Probar con una cantidad mayor el método de ConjuntoPotencia, verificar que la cantidad de subconjuntos corresponda a la regla 2^n y corroborar que su ultimo subconjunto corresponda al de la lista original de artículos.

|  |  |
| --- | --- |
| Clase | FuercitaBruta |
| Método | ConjuntoPotencia(List<Articulo>) |
| Escenario | Escenario5 |
| Valores de entrada | Los 9 artículos del escenario |
| Resultado | 2^n = 512  Ultimo subconjunto {A, B, C, D, E, F, G, H, I} |

Objetivo prueba 7: Probar con una cantidad minima que el método FrequentItemSet retorne la lista de los artículos frecuentados de acuerdo al parámetro suppCountPar

|  |  |
| --- | --- |
| Clase | FuercitaBruta |
| Método | FrequentItemSet(IEnumerable<IEnumerable<Articulo>>, suppCountPar) |
| Escenario | Escenario4 |
| Valores de entrada | Las lista con las 16 combinaciones de los 4 artículos del escenario y un suppCountPar de 0,25 |
| Resultado | Lista {C,D} y {A, D} |