Plan de Pruebas Data QA

# Introducción:

Este documento tiene como objetivo planear, analizar, diseñar, preparar, ejecutar, evaluar y realizar informe de pruebas a un documento de 6120 datos.

**Descripción del Conjunto de Datos (ventas.csv):**

El archivo ventas.csv contiene las siguientes columnas:

### Tipado de las Columnas:

* fecha\_venta (**Fecha**): Fecha en que se realizó la venta (YYYY-MM-DD)
* id\_producto (**Entero**): Identificador único del producto.
* nombre\_producto (**String**): Nombre del producto.
* categoria (**String**): Categoría a la que pertenece el producto.
* precio (**Flotante**): Precio unitario del producto.
* cantidad\_vendida (**Entero**): Cantidad de unidades vendidas.
* total\_venta (**Flotante**): Importe total de la venta.
* nombre\_cliente (**String**): Nombre del cliente.
* region (**String**): Región geográfica de la venta.
* metodo\_pago (**String**): Método de pago utilizado.

### Objetivos de las Pruebas:

Garantizar que los datos de ventas.csv cumple con los estándares de calidad que se necesitan para el análisis confiable y una buena decisión de toma de decisiones, con esas pruebas se espera:

* Validar la precisión de los datos
* Validar que cada tipo de columna sea correcta
* Validar que no exista valores inválidos, vacíos o duplicados
* Validar que coherentemente la lógica de cada uno de los campos
* Detectar problemas de calidad en información crucial para el análisis y decisiones comerciales
* Validar que cada campo tiene el formato esperado

### Alcance de las pruebas:

Se ejecutan pruebas para validar cada columna del archivo ventas.csv con el fin de buscar problemas de calidad. En cada columna, se va a identificar problemas explícitos, por ejemplo, valores fuera del rango, formato de datos incorrecto, caracteres especiales, datos nulos o duplicados, entre otros.

# Metodología utilizada

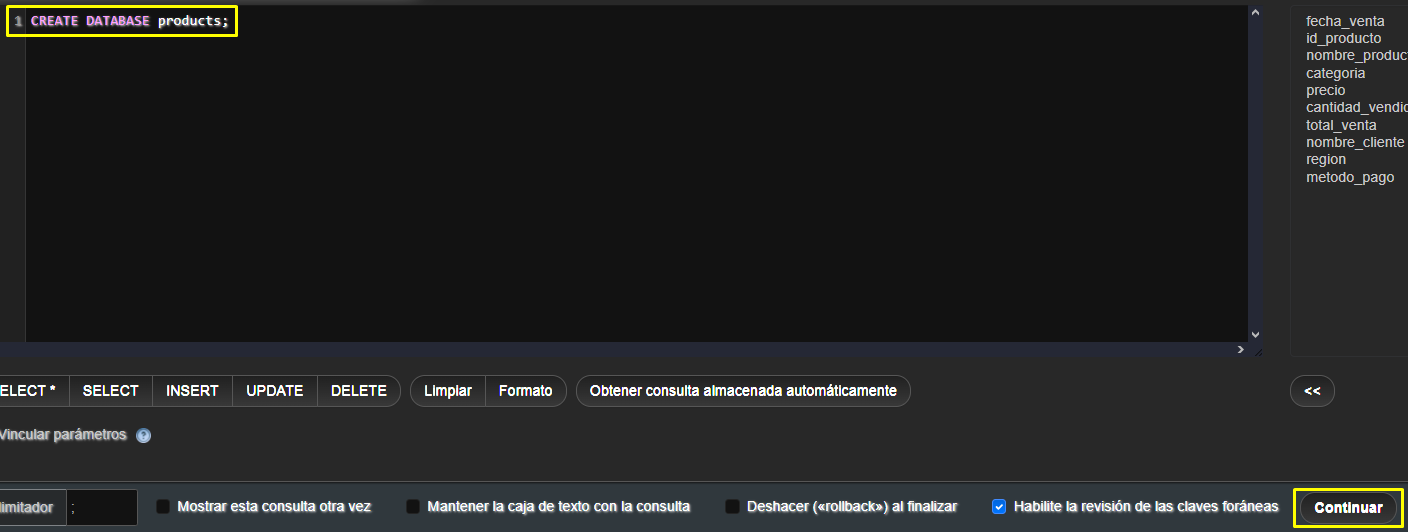
Para llevar a cabo las pruebas se utilizará la metodología ágil scrum, ejecutando los principios de trabajo como historias de usuario, iteraciones cortas y entrega continua de resultados.

Entre las herramientas que se van a usar están Python, Pandas, Pytest, Excel, Csv.

# Carga de información de ventas.csv a base de datos

Se realiza el cargue de la información al gestor de base de datos php MyAdmin para que sea mucho más fácil la administración y gestión de la misma y así se garantice la integridad de la información de las pruebas que se van a ejecutar posteriormente

1. Creación de la base de datos: CREATE DATABASE products;



Base de datos creada



2. Creación de la tabla ventas

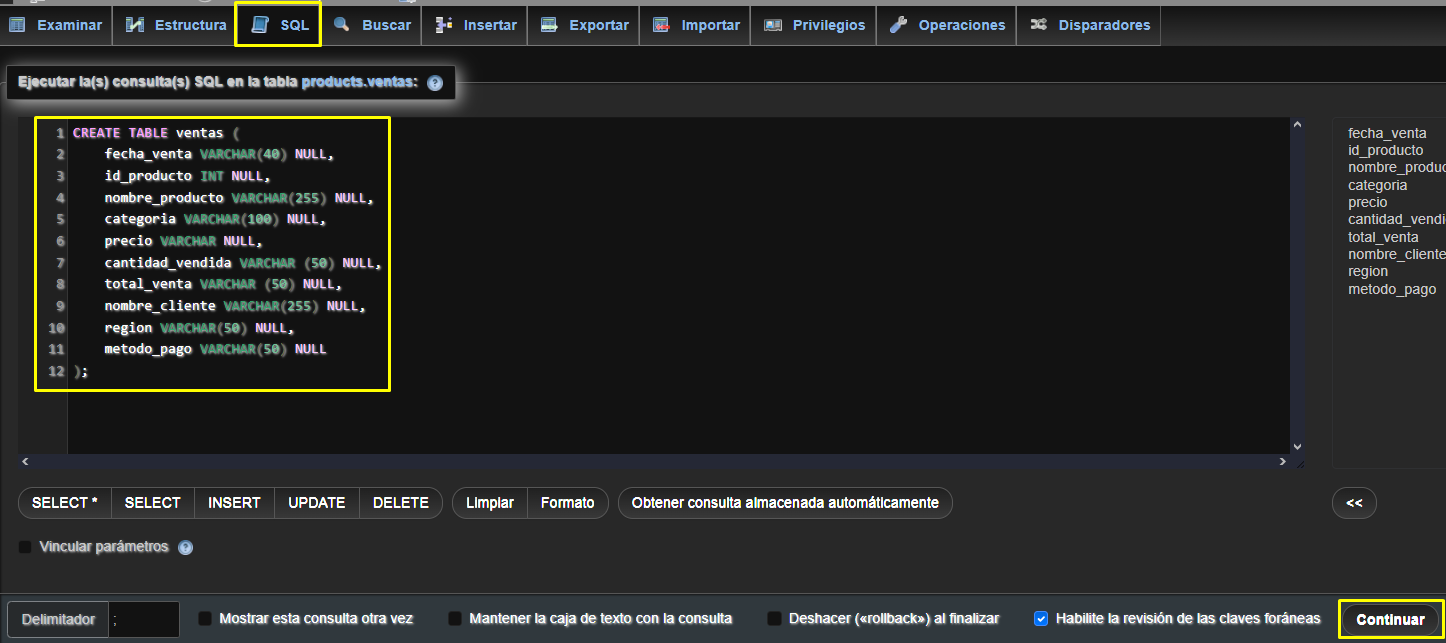
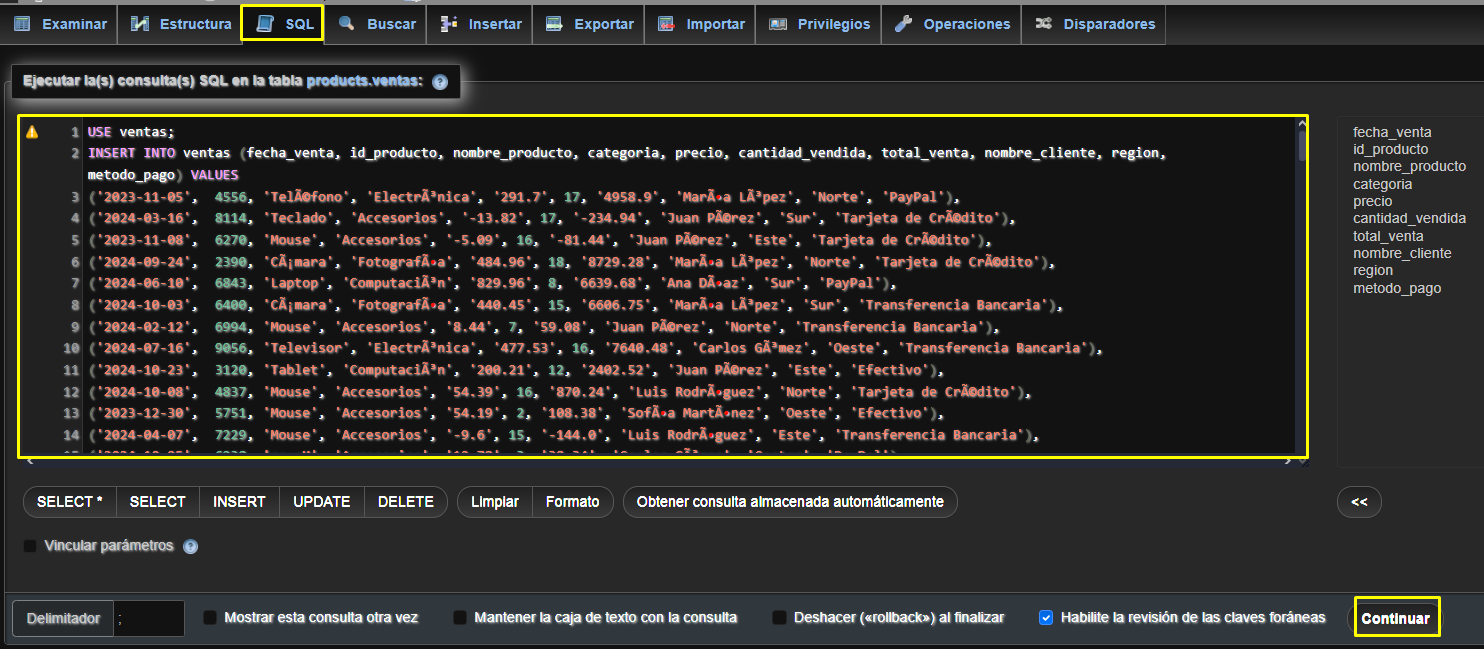


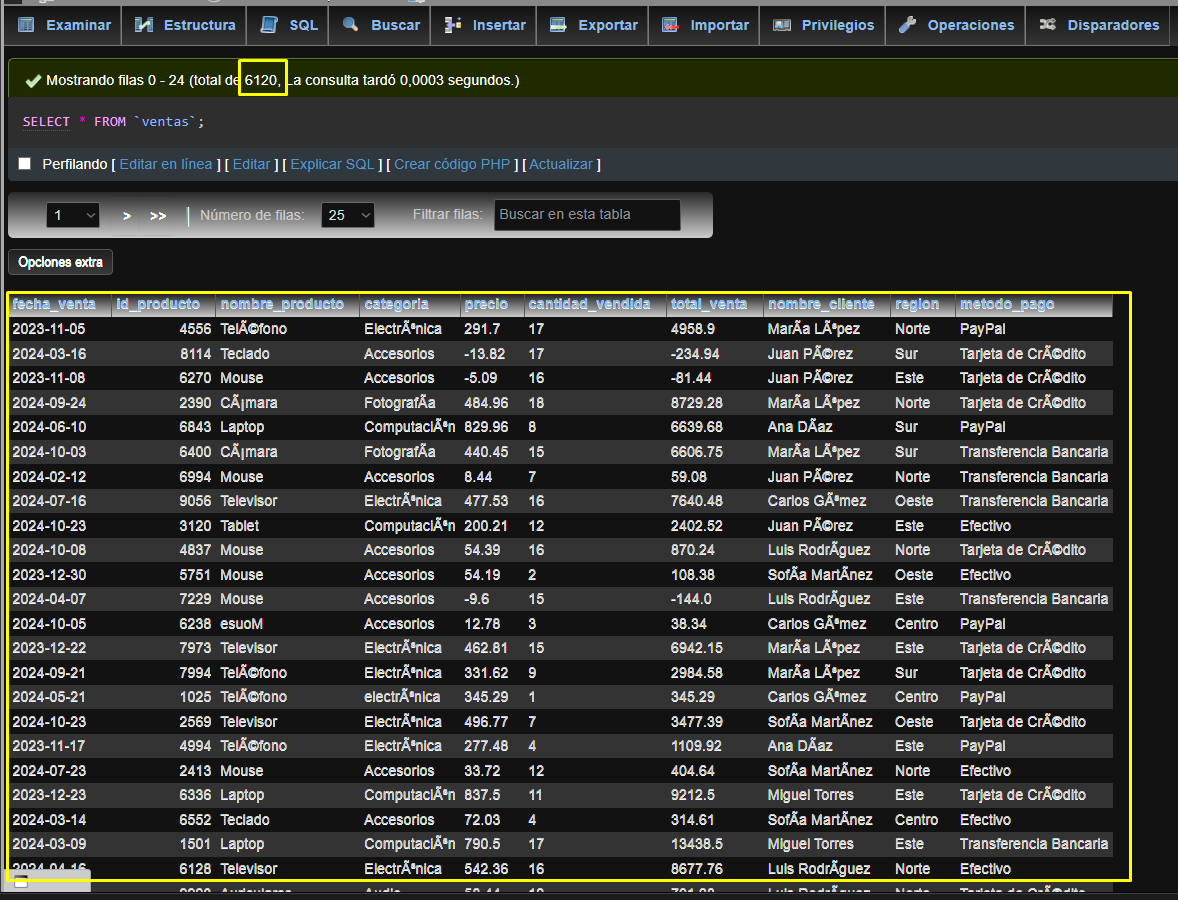
Tabla creada



3. Inserción de los datos para la tabla ventas



Datos ingresados



# Historias de usuario

|  |  |
| --- | --- |
| Id HU | Descripción HU |
| 1 | Como analista de calidad de datos,  quiero identificar y analizar las fechas de venta en el dataset, asegurándome de que cumplan con formatos y valores válidos, para garantizar la integridad, coherencia y utilidad de la información en los reportes de ventas. |
| 2 | Como analista de calidad de datos,  quiero garantizar que los valores del campo id\_producto sean únicos, coherentes y correctos, para asegurar la trazabilidad de los productos y evitar errores en el manejo de datos relacionados. |
| 3 | Como analista de calidad de datos,  quiero asegurar que los valores en la columna nombre\_producto cumplan con las reglas de formato, coherencia y datos válidos, para garantizar que los productos estén correctamente registrados y se puedan analizar sin errores. |
| 4 | Como analista de calidad, quiero validar la integridad y coherencia de los datos en el campo categoria, para asegurar que los registros sean precisos y cumplan con los estándares de calidad necesarios para análisis confiables. |
| 5 | Como analista de calidad de datos, quiero asegurar que los valores del campo precio sean válidos y consistentes, para garantizar la precisión en los reportes y análisis financieros derivados de esta información. |
| 6 | Como analista de calidad, quiero validar las reglas de negocio para el campo cantidad\_vendida, para garantizar que los datos cumplen con los criterios de integridad y calidad requeridos. |
| 7 | Como analista de calidad, quiero asegurarme de que los datos en el campo total\_venta cumplan con los requisitos de calidad, para garantizar que las validaciones respalden la precisión y confiabilidad del análisis de datos. |
| 8 | Como analista de datos, quiero realizar pruebas de calidad en el campo nombre\_cliente, para asegurar que cumple con los requisitos de negocio y calidad. |
| 9 | **Como** analista de datos, **quiero** asegurar la calidad de los datos en el campo "región", **para** garantizar que la información sea consistente, precisa y útil para la toma de decisiones empresariales. |
| 10 | Como analista de datos, quiero realizar un análisis exhaustivo de los valores en el campo metodo\_pago,  para asegurar la calidad de los datos mediante la identificación de inconsistencias, valores especiales, y vacíos, y garantizar que cumplen con los requisitos del negocio para una correcta categorización de los métodos de pago. |

# Casos de prueba

## Campo fecha\_venta

Lista de casos de prueba para cada una de las historias de usuario que se definieron en el punto 4

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Localizar y contar los registros que cumplen con el formato de fecha YYYY-MM-DD. | 1. Validar que tanto en el archivo ‘ventas1.csv’ como en la base de datos existen registros con formato de fecha ‘YYYY-MM-DD’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 2 | Determinar el número total de registros que cumplen con el formato YYYY-MM-DD. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con formato de fecha YYYY-MM-DD 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 3 | Verificar si hay exactamente 5835 registros con el formato YYYY-MM-DD. | 1. Validar que tanto en el archivo ‘ventas1.csv’ como en la base de datos existen efectivamente 5835 registros con formato de fecha YYYY-MM-DD. 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 4 | Localizar registros que utilicen el formato de fecha DD/MM/YYYY. | 1. Validar que tanto en el archivo ‘ventas1.csv’ como en la base de datos existen registros con formato de fecha DD/MM/YYYY 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 5 | Determinar la cantidad de registros con el formato de fecha DD/MM/YYYY. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con formato de fecha DD/MM/YYYY 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 6 | Verificar si existen 234 registros con el formato DD/MM/YYYY. | 1. Validar que tanto en el archivo ‘ventas.csv’ como en la base de datos existen efectivamente 234 registros con el formato de fecha DD/MM/YYYY. 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 7 | Localizar y contar los registros que contengan el texto 'Fecha desconocida'. | 1. Validar que tanto en el archivo ‘ventas1.csv’ como en la base de datos existen registros con el texto “Fecha desconocida” 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 8 | Determinar el número total de registros con el texto 'Fecha desconocida'. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con el texto “Fecha desconocida” 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 9 | Verificar si existen 4 registros con el texto 'Fecha desconocida'. | 1. Validar que tanto en el archivo ‘ventas.csv’ como en la base de datos existen efectivamente 4 registros con el texto 'Fecha desconocida'. 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 10 | Localizar registros con formatos de fecha diferentes al estándar YYYY-MM-DD. | 1. Validar que tanto en el archivo ‘ventas1.csv’ como en la base de datos existen registros con el formato de fecha distinto a ‘YYYY-MM-DD’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 11 | Determinar cuántos registros tienen un formato de fecha diferente a YYYY-MM-DD. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con formato de fecha distinto a ‘YYYY-MM-DD’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 12 | Verificar si existen 238 registros con formatos distintos a YYYY-MM-DD. | 1. Validar que tanto en el archivo ‘ventas1.csv’ como en la base de datos existen efectivamente 238 registros con formato de fecha distinto a ‘YYYY-MM-DD.’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 13 | Localizar los registros en fecha\_venta que estén vacíos o sean nulos. | 1. Validar que tanto en el archivo ‘ventas1.csv’ como en la base de datos existen registros vacíos o nulos 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 14 | Determinar la cantidad de registros vacíos o nulos en fecha\_venta. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay vacíos o nulos 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 15 | Verificar si existen exactamente 47 registros vacíos o nulos en fecha\_venta. | 1. Validar que tanto en el archivo ‘ventas1.csv’ como en la base de datos existen efectivamente 47 registros vacíos o nulos 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 16 | Revisar la columna fecha\_venta para asegurarse de que no haya fechas mayores a la actual. | 1. Validar que tanto en el archivo ‘ventas1.csv ‘ no hay registros con fecha superior a la actual. 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 17 | Verificar si hay exactamente 890 registros con el formato YYYY-MM-DD. | 1. Validar tanto en el archivo ‘ventas1.csv ‘como en la base de datos que no hay 890 registros con el formato de fecha ‘YYYY-MM-DD.’ 2. Validar que tanto en la consola como en el archivo html se muestre el error por el cual falló el caso de prueba 3. Validar que el caso de prueba tome el estado “failed” |
| 18 | Verificar si existen 300 registros con el formato DD/MM/YYYY. | 1. Validar tanto en el archivo ‘ventas1.csv ‘como en la base de datos que no hay 300 registros con el formato de fecha DD/MM/YYYY. 2. Validar que tanto en la consola como en el archivo html se muestre el error por el cual falló el caso de prueba 3. Validar que el caso de prueba tome el estado “failed” |
| 19 | Verificar si existen 55 registros con el texto 'Fecha desconocida'. | 1. Validar tanto en el archivo ‘ventas1.csv ‘como en la base de datos que no hay 55 registros con el texto 'Fecha desconocida'. 2. Validar que tanto en la consola como en el archivo html se muestre el error por el cual falló el caso de prueba 3. Validar que el caso de prueba tome el estado “failed” |
| 20 | Verificar si existen 333 registros con formatos distintos a YYYY-MM-DD. | 1. Validar tanto en el archivo ‘ventas1.csv ‘como en la base de datos que no hay 333 registros con formato de fecha distinto a YYYY-MM-DD. 2. Validar que tanto en la consola como en el archivo html se muestre el error por el cual falló el caso de prueba 3. Validar que el caso de prueba tome el estado “failed” |
| 21 | Verificar si existen exactamente 987 registros vacíos o nulos en fecha\_venta. | 1. Validar tanto en el archivo ‘ventas1.csv ‘como en la base de datos que no hay 987 registros vacíos o nulos 2. Validar que tanto en la consola como en el archivo html se muestre el error por el cual falló el caso de prueba 3. Validar que el caso de prueba tome el estado “failed” |

## Campo id\_producto

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Identificar qué valores en la columna id\_producto son únicos. | Identificar los id\_productos únicos. |
| 2 | Contar cuántos valores únicos existen en la columna id\_producto. | Validar cuántos id\_producto únicos hay. |
| 3 | Confirmar que el número de valores únicos en la columna id\_producto coincide con un valor específico (n). | Validar que hay n cantidad de id\_producto únicos. |
| 4 | Identificar qué registros tienen valores duplicados en la columna id\_producto. | Identificar los id\_producto duplicados. |
| 5 | Contar cuántos valores duplicados existen en la columna id\_producto. | Validar cuántos id\_producto duplicados hay. |
| 6 | Confirmar que el número de valores duplicados en la columna id\_producto coincide con un valor específico (n). | Validar que hay n cantidad de id\_producto duplicados. |
| 7 | Comprobar que no existan registros nulos o vacíos en la columna id\_producto. | Validar que no haya datos nulos o vacíos. |
| 8 | Asegurar que todos los valores de la columna id\_producto sean numéricos. | Validar que no haya datos string (texto). |
| 9 | Asegurar que si hay duplicados en id\_producto, estén vinculados al mismo nombre\_producto. | Validar que si el id\_producto se duplica, ese id\_producto debe estar asociado al mismo nombre\_producto. |
| 10 | Verificar que no existan registros en la columna id\_producto con valores negativos. | Validar que no haya id\_producto menores que cero. |
| 11 | Asegurar que todos los valores en la columna id\_producto sean positivos. | Validar que id\_producto sea mayor a cero. |

## Campo nombre\_producto

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Identificar si hay registros con caracteres especiales en la columna nombre\_producto. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que existen registros que contienen caracteres especiales en el campo nombre\_producto 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 2 | Contar los registros que contienen caracteres especiales en la columna nombre\_producto. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con caracteres especiales en el campo nombre\_producto 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 3 | Validar que hay exactamente 1341 registros con caracteres especiales en nombre\_producto | 1. Validar que hay 1341 registros para nombre\_producto con caracteres especiales 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 4 | Identificar si existen registros en la columna nombre\_producto con el valor "Teclado". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros con nombre\_producto ‘Teclado’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 5 | Contar cuántos registros en la columna nombre\_producto tienen el valor "Teclado". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con nombre\_producto ‘Teclado’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 6 | Validar si la cantidad de registros con el nombre "Teclado" corresponde a 655 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 655 registros para nombre\_producto con caracteres especiales 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 7 | Identificar si existen registros en la columna nombre\_producto con el valor "Mouse". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros con nombre\_producto ‘Mouse’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 8 | Contar cuántos registros en la columna nombre\_producto tienen el valor "Mouse". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con nombre\_producto ‘Mouse’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 9 | Validar si la cantidad de registros con el nombre "Mouse" corresponde a 718 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 718 registros para nombre\_producto ‘Mouse’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 10 | Identificar si existen registros en la columna nombre\_producto con el valor "Auriculares". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros con nombre\_producto ‘Auriculares’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 11 | Contar cuántos registros en la columna nombre\_producto tienen el valor "Auriculares". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con nombre\_producto ‘Auriculares’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 12 | Validar si la cantidad de registros con el nombre\_producto "Auriculares" corresponde a 643 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 643 registros para nombre\_producto ‘Auriculares’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 13 | Contar los registros en la columna nombre\_producto que comienzan con la letra "A". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en nombre\_producto que empiezan con la letra ‘A’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 14 | Validar que el número de registros que comienzan con "A" corresponde a 643 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 643 registros en nombre\_producto que empiezan con la letra ‘A’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 15 | Contar los registros en la columna nombre\_producto que comienzan con la letra "E". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en nombre\_producto que empiezan con la letra ‘E’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 16 | Validar que el número de registros que comienzan con "E" corresponde a 21 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 21 registros en nombre\_producto que empiezan con la letra ‘E’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 17 | Contar los registros en la columna nombre\_producto que comienzan con la letra "T". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en nombre\_producto que empiezan con la letra ‘T’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 18 | Validar que el número de registros que comienzan con "T" corresponde a un 2671 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 2671 registros en nombre\_producto que empiezan con la letra ‘T’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 19 | Contar los registros en la columna nombre\_producto que comienzan con la letra "S". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en nombre\_producto que empiezan con la letra ‘S’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 20 | Validar que el número de registros que comienzan con "S" corresponde a un valor específico (n). | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 14 registros en nombre\_producto que empiezan con la letra ‘S’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 21 | Contar los registros en la columna nombre\_producto que comienzan con la letra "M". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en nombre\_producto que empiezan con la letra ‘M’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 22 | Validar que el número de registros que comienzan con "M" corresponde a 718 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 718 registros en nombre\_producto que empiezan con la letra ‘M’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 23 | Revisar la columna nombre\_producto para identificar si hay registros con valores nulos o vacíos. | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que no hay datos nulos o vacíos en el campo nombre\_producto 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 24 | Revisar la columna nombre\_producto para asegurarse de que no contenga valores numéricos. | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que no hay datos numéricos en el campo nombre\_producto 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 25 | Revisar la columna nombre\_producto y contar cuántos registros contienen el carácter especial Ã¡. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con este carácter especial: Ã¡ en el campo nombre\_producto 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 26 | Revisar la columna nombre\_producto y contar cuántos registros contienen el carácter especial Ã© | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con este carácter especial: Ã© en el campo nombre\_producto 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |

## Campo categoria

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Identificar registros con caracteres especiales en la columna categoria. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que existen registros que contienen caracteres especiales en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 2 | Contar los registros que contienen caracteres especiales en la columna categoria. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con caracteres especiales en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 3 | Validar que hay exactamente 3398 registros con caracteres especiales en el campo categoria | 1. Validar que hay 3398 registros con caracteres especiales en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 4 | Identificar registros que no tienen caracteres especiales en la columna categoria. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que existen registros que no contienen caracteres especiales en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 5 | Contar los registros que no contienen caracteres especiales en la columna categoria. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay sin caracteres especiales en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 6 | Validar que hay exactamente 2686 registros que no contienen caracteres especiales en el campo categoria | 1. Validar que hay 2686 registros sin caracteres especiales en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 7 | Identificar si existen registros en la columna ‘categoria’ con el valor "Audio". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros con ‘categoria’ ‘Audio’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 8 | Contar cuántos registros en la columna ‘categoria’ tienen el valor "Audio". | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con ‘categoria’ ‘Audio’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 9 | Validar si la cantidad de registros con el nombre "Audio" corresponde a 655 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 655 registros para ‘categoria’ ‘Audio’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 10 | Identificar si existen registros en la columna ‘categoria’ con el valor ‘Oficina’ | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros con ‘categoria’ ‘Oficina’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 11 | Contar cuántos registros existen en la en la columna ‘categoria’ con el valor ‘Oficina’ | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos cuántos registros hay con ‘categoria’ ‘Oficina’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 12 | Validar si la cantidad de registros de ‘categoria’ con el nombre "Oficina" corresponde a 624 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 624 registros para ‘categoria’ ‘Oficina’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 13 | Identificar registros en la columna ‘categoria’ comienzan con la letra ‘F’. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en el campo ‘categoria’ que empiezan con la letra ‘F’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 14 | Contar los registros en la columna ‘categoria’ que comienzan con la letra ‘F’ | 1. Contar tanto en el archivo ventas1.csv como en la base de datos los registros en el campo ‘categoria’ que empiezan con la letra ‘F’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 15 | Validar si la cantidad de registros que comienzan con la letra "F" corresponde a 670 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 670 registros que comienzan con la letra ‘F’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 16 | Identificar registros en la columna ‘categoria’ comienzan con la letra ‘A’. | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en el campo ‘categoria’ que empiezan con la letra ‘A’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 17 | Contar los registros en la columna ‘categoria’ que comienzan con la letra ‘A’ | 1. Contar tanto en el archivo ventas1.csv como en la base de datos los registros en el campo ‘categoria’ que empiezan con la letra ‘A’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 18 | Validar si la cantidad de registros que comienzan con la letra "A" corresponde a 2062 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 2062 registros que comienzan con la letra ‘A’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 19 | Identificar registros en la columna ‘categoria’ vacíos o nulos | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros vacíos en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 20 | Contar los registros en la columna ‘categoria’ vacíos u nulos | 1. Contar tanto en el archivo ventas1.csv como en la base de datos que hay registros vacíos en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 21 | Validar si la cantidad de registros vacíos corresponde a 36 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 36 registros vacíos en el campo ‘categoria’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 22 | Identificar registros en la columna ‘categoria’ que tienen el carácter ‘Ã’ | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en el campo ‘categoria’ que tienen el carácter ‘Ã’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 23 | Contar los registros en la columna ‘categoria’ que tienen el carácter ‘Ã’ | 1. Contar tanto en el archivo ventas1.csv como en la base de datos que hay registros en el campo ‘categoria’ que tienen el carácter ‘Ã’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 24 | Validar si la cantidad de registros que tienen el carácter ‘Ã’ corresponde a 670 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 36 registros en el campo ‘categoria’ que tienen el carácter ‘Ã’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 25 | Identificar registros en la columna ‘categoria’ que tienen el carácter ‘Ã³’ | 1. Validar tanto en el archivo ‘ventas1.csv’ como en la base de datos que hay registros en el campo ‘categoria’ que tienen el carácter ‘Ã³’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 26 | Contar los registros en la columna ‘categoria’ que tienen el carácter ‘Ã³’ | 1. Contar tanto en el archivo ventas1.csv como en la base de datos que hay registros en el campo ‘categoria’ que tienen el carácter ‘Ã³’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |
| 27 | Validar si la cantidad de registros que tienen el carácter ‘Ã³’ corresponde a 2728 | 1. Validar tanto en el archivo ventas1.csv como en la base de datos que hay 36 registros en el campo ‘categoria’ que tienen el carácter ‘Ã³’ 2. Validar que tanto en la consola como en el archivo html el caso de prueba esté en estado “passed” |

## Campo precio

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Determinar cuál es el valor más alto registrado en la columna precio. | 1. Validar el valor más alto del campo ‘precio’ tanto en el documento ‘ventas1.csv’ como en la base de datos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 2 | Asegurarse de que el valor más alto en la columna precio es 849.91 | 1. Validar que el valor más alto del campo ‘precio’ tanto en el documento ‘ventas1.csv’ como en la base de datos es el 849.91 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 3 | Determinar cuál es el valor más bajo registrado en la columna precio. | 1. Validar el valor más bajo tanto en el documento ‘ventas1.csv’ como en la base de datos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 4 | Asegurarse de que el valor más bajo en la columna precio es -844.33 | 1. Validar que el valor más alto del campo ‘precio’ tanto en el documento ‘ventas1.csv’ como en la base de datos es el -844.33 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 5 | Identificar y contar registros con valores positivos en la columna precio. | 1. Contar los valores positivos tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 6 | Asegurarse de que hay 5368 registros positivos en la columna precio | 1. Validar que hay 5368 registros positivos tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 7 | Identificar y contar registros con valores negativos en la columna precio. | 1. Contar los valores negativos tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 8 | Asegurarse de que hay 492 registros positivos en la columna precio | 1. Validar que hay 492 registros negativos tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 9 | Identificar y contar registros vacíos o nulos en la columna precio. | 1. Contar los valores vacíos o nulos tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 10 | Asegurarse de que hay 35 registros positivos en la columna precio | 1. Validar que hay 35 registros vacíos o nulos tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 11 | Comprobar que existen registros con valores negativos en la columna 'precio'. | 1. Validar que efectivamente tanto en el documento ‘ventas1.csv’ como en la base de datos existente registros negativos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘failed’ |
| 12 | Confirmar que no hay registros con valores negativos en la columna 'precio'. | 1. Validar que efectivamente tanto en el documento ‘ventas1.csv’ como en la base de datos si hay registros negativos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘failed’ |
| 13 | Confirmar que todos los valores en la columna 'precio' son numéricos. | 1. Validar que tanto en el documento ‘ventas1.csv’ como en la base de datos no todos los registros son numéricos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘failed’ |
| 14 | Confirmar que todos los valores en la columna precio son numéricos. | 1. Validar que tanto en el documento ‘ventas1.csv’ como en la base de datos no todos los registros son numéricos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘failed’ |
| 15 | Identificar y contar registros que están como tipo de dato string en la columna precio. | 1. Contar los valores positivos tanto en el documento ‘ventas1.csv’ como en la base de datos en el campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 16 | Asegurarse de que hay 6085 registros como tipo de dato string en la columna precio | 1. Validar que hay 6085 registros como string tanto en el documento ‘ventas1.csv’ como en la base de datos en el campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’ |
| 17 | Identificar y contar cuántos registros tienen valores entre 100 y 200 en la columna precio. | 1. Validar cuántos registros hay entre 100 y 200 tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’. |
| 18 | Asegurarse de que hay 923 registros que están entre 100 y 200 en la columna precio. | 1. Validar que hay 923 registros entre 100 y 200 tanto en el documento ‘ventas1.csv’ como en la base de datos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’. |
| 19 | Identificar y contar los registros que comienzan con $ en la columna precio. | 1. Validar cuántos registros hay que empiezan con $ tanto en el documento ‘ventas1.csv’ como en la base de datos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’. |
| 20 | Asegurarse de que hay 225 registros que empiezan con $ en la columna precio. | 1. Validar que hay 225 registros que empiezan con $ tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’. |
| 21 | Identificar y contar los registros que comienzan con - en la columna precio. | 1. Validar cuántos registros hay que empiezan con - tanto en el documento ‘ventas1.csv’ como en la base de datos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’. |
| 22 | Asegurarse de que hay 492 registros que empiezan con - en la columna precio. | 1. Validar que hay 492 registros que empiezan con - tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’. |
| 23 | Identificar y contar los registros que comienzan con - en la columna precio. | 1. Validar cuántos registros hay que empiezan con - tanto en el documento ‘ventas1.csv’ como en la base de datos 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’. |
| 24 | Asegurarse de que hay 507 registros que empiezan con - en la columna precio. | 1. Validar que hay 507 registros que empiezan con - tanto en el documento ‘ventas1.csv’ como en la base de datos del campo ‘precio’ 2. Validar que el caso de prueba tanto en el reporte\_precio.html y en la consola tenga el estado ‘passed’. |

## Campo cantidad\_vendida

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Detectar el registro con el valor más alto en el campo cantidad\_vendida. | 1. Validar que el sistema devuelve correctamente el valor más alto registrado en el campo. |
| 2 | Determinar el registro con el valor más bajo en la columna cantidad\_vendida. | 1. Validar el valor más bajo en la columna cantidad\_vendida se identifica correctamente. |
| 3 | Confirmar que existen valores positivos en la columna cantidad\_vendida. | 1. Validar que se confirma la existencia de valores positivos en la columna. |
| 4 | Determinar cuántos registros contienen valores positivos en la columna cantidad\_vendida. | 1. Validar que el sistema cuenta correctamente los registros con valores positivos. |
| 5 | Validar que no existan valores positivos en la columna cantidad\_vendida. | 1. Validar que el sistema confirma correctamente que no existen valores positivos. |
| 6 | Validar que hay registros con valores negativos en la columna cantidad\_vendida. | 1. Validar que el sistema confirma la existencia de valores negativos en la columna. |
| 7 | Validar que no existan valores negativos en la columna cantidad\_vendida. | 1. Validar que el sistema confirma correctamente que no existen valores negativos. |
| 8 | Determinar cuántos registros contienen valores negativos en la columna cantidad\_vendida. | 1. Validar que el sistema cuenta correctamente los registros con valores negativos. |
| 9 | Validar que no existan valores nulos o vacíos en la columna cantidad\_vendida. | 1. Validar que el sistema confirma correctamente que no hay datos nulos o vacíos. |
| 10 | Validar que no existan cadenas de texto o strings en la columna cantidad\_vendida. | 1. Validar que el sistema confirma correctamente que no hay cadenas de texto o strings. |
| 11 | Validar que haya n cantidad de valores positivos en la columna cantidad\_vendida. | 1. Validar que el sistema identifica correctamente que hay n cantidad de valores positivos. |
| 12 | Validar que haya n cantidad de valores negativos en la columna cantidad\_vendida. | 1. Validar que el sistema identifica correctamente que hay n cantidad de valores negativos. |
| 13 | Validar que haya n cantidad de valores nulos o vacíos en la columna cantidad\_vendida. | 1. Validar que el sistema identifica correctamente que hay n cantidad de valores nulos o vacíos. |
| 14 | Validar que haya n cantidad de cadenas de texto o strings en la columna cantidad\_vendida. | 1. Validar que el sistema identifica correctamente que hay n cantidad de cadenas de texto o strings. |
| 15 | Validar cuántos valores se encuentran dentro del rango de 100 a 200 en la columna cantidad\_vendida. | 1. Validar que el sistema identifica correctamente cuántos datos se encuentran en el rango de 100 a 200. |
| 16 | Validar que existen datos mayores a 100 en la columna cantidad\_vendida. | 1. Validar que el sistema confirma correctamente que existen datos mayores a 100. |
| 17 | Validar que existen datos mayores a 1000 en la columna cantidad\_vendida. | 1. Validar que el sistema confirma correctamente que existen datos mayores a 1000. |
| 18 | Validar que no existan valores mayores a 100 en la columna cantidad\_vendida. | 1. Validar que el sistema confirma correctamente que no hay datos mayores a 100. |
| 19 | Validar que no existan valores mayores a 1000 en la columna cantidad\_vendida. | 1. Validar que el sistema confirma correctamente que no hay datos mayores a 1000. |
| 20 | Validar que haya n cantidad de valores positivos en la columna cantidad\_vendida en el rango de 100 a 200 | 1. Validar que el sistema identifica correctamente que hay n cantidad de datos entre 100 y 200. |
| 21 | Validar que haya n cantidad de valores mayores a 100 en la columna cantidad\_vendida. | 1. Validar que el sistema identifica correctamente que hay n cantidad de valores mayores a 100. |
| 22 | Validar que haya n cantidad de valores mayores a 1000 en la columna cantidad\_vendida. | 1. Validar que el sistema identifica correctamente que hay n cantidad de valores mayores a 1000. |

## Campo total\_venta

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Localizar el valor más alto registrado en el campo total\_venta. | 1. Validar el valor más alto. |
| 2 | Localizar el valor más bajo registrado en el campo total\_venta. | 1. Validar el valor más bajo. |
| 3 | Confirmar que existen valores mayores a 0 en la columna total\_venta. | 1. Validar que hay valores positivos. |
| 4 | Contar cuántos valores mayores a 0 hay en la columna total\_venta. | 1. Validar cuántos datos con valores positivos hay. |
| 5 | Confirmar que no hay valores mayores a 0 en la columna total\_venta. | 1. Validar que no hay valores positivos. |
| 6 | Confirmar que existen valores menores a 0 en la columna total\_venta. | 1. Validar que hay datos con valores negativos. |
| 7 | Confirmar que no hay valores menores a 0 en la columna total\_venta. | 1. Validar que no haya datos con valores negativos. |
| 8 | Contar cuántos valores menores a 0 hay en la columna total\_venta. | 1. Validar cuántos datos con valores negativos hay. |
| 9 | Verificar que no existan valores nulos o vacíos en la columna total\_venta. | 1. Validar que no haya datos nulos o vacíos. |
| 10 | Confirmar que no existen cadenas de texto o strings en la columna total\_venta. | 1. Validar que no haya cadenas de texto o string. |
| 11 | Contar cuántos valores positivos existen en la columna total\_venta. | 1. Validar qué hay n cantidad de datos positivos. |
| 12 | Contar cuántos valores negativos existen en la columna total\_venta. | 1. Validar qué hay n cantidad de datos negativos. |
| 13 | Contar cuántos valores nulos o vacíos hay en la columna total\_venta. | 1. Validar qué hay n cantidad de datos nulos o vacíos. |
| 14 | Contar cuántas cadenas de texto o strings existen en la columna total\_venta. | 1. Validar qué hay n cantidad de datos como string o cadena de caracteres. |
| 15 | Determinar cuántos valores están entre 100 y 200 en la columna total\_venta. | 1. Validar cuántos datos hay entre 100 y 200. |
| 16 | Confirmar que hay valores mayores a 100 en la columna total\_venta. | 1. Validar que hay datos > 100. |
| 17 | Confirmar que hay valores mayores a 1000 en la columna total\_venta. | 1. Validar que hay datos > 1000. |
| 18 | Confirmar que no hay valores mayores a 100 en la columna total\_venta. | 1. Validar que no hay datos > 100. |
| 19 | Confirmar que no hay valores mayores a 1000 en la columna total\_venta. | 1. Validar que no hay datos > 1000. |
| 20 | Contar cuántos valores están entre 100 y 200 en la columna total\_venta. | 1. Validar qué hay n cantidad de datos entre 100 y 200. |
| 21 | Contar cuántos valores mayores a 100 existen en la columna total\_venta. | 1. Validar qué hay n cantidad de datos > 100. |
| 22 | Contar cuántos valores mayores a 1000 existen en la columna total\_venta. | 1. Validar qué hay n cantidad de datos > 1000. |

## Campo nombre\_cliente

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Identificar registros en la columna nombre\_cliente que contienen caracteres especiales. | 1. Validar datos con caracteres especiales. |
| 2 | Asegurar que existen registros sin caracteres especiales en la columna nombre\_cliente. | 1. Validar datos sin caracteres especiales. |
| 3 | Confirmar que existen registros con el nombre Miguel Torres en la columna nombre\_cliente. | 1. Validar que hay datos con el nombre Miguel Torres. |
| 4 | Determinar cuántos registros tienen el nombre Miguel Torres en la columna nombre\_cliente. | 1. Validar cuántos datos hay con el nombre Miguel Torres. |
| 5 | Confirmar que hay n cantidad de registros con el nombre Miguel Torres. | 1. Validar que hay n cantidad para Miguel Torres. |
| 6 | Confirmar que existen registros con el nombre Carlos en la columna nombre\_cliente. | 1. Validar que hay datos con el nombre Carlos. |
| 7 | Determinar cuántos registros tienen el nombre Carlos en la columna nombre\_cliente. | 1. Validar cuántos datos hay con el nombre Carlos. |
| 8 | Confirmar que hay n cantidad de registros con el nombre Carlos. | 1. Validar que hay n cantidad para Carlos. |
| 9 | Confirmar que existen registros con el nombre Ana en la columna nombre\_cliente. | 1. Validar que hay datos con el nombre Ana. |
| 10 | Determinar cuántos registros tienen el nombre Ana en la columna nombre\_cliente. | 1. Validar cuántos datos hay con el nombre Ana. |
| 11 | Confirmar que hay n cantidad de registros con el nombre Ana. | 1. Validar que hay n cantidad para Ana. |
| 12 | Determinar cuántos registros empiezan con la letra A en la columna nombre\_cliente. | 1. Validar cuántos datos hay que empiecen con la letra A. |
| 13 | Confirmar que hay n cantidad de registros que empiezan con la letra A. | 1. Validar que hay n cantidad con la letra A. |
| 14 | Confirmar que no hay registros nulos o vacíos en la columna nombre\_cliente. | 1. Validar que no haya datos nulos o vacíos. |
| 15 | Confirmar que no hay valores numéricos en la columna nombre\_cliente. | 1. Validar que no haya datos numéricos. |
| 16 | Contar cuántos registros tienen el carácter especial Ã³ en la columna nombre\_cliente. | 1. Validar cuántos datos hay con este carácter especial: Ã³. |
| 17 | Contar cuántos registros tienen el carácter especial Ã en la columna nombre\_cliente. | 1. Validar cuántos datos hay con este carácter especial: Ã |
| 18 | Contar cuántos registros tienen el carácter especial Ã© en la columna nombre\_cliente. | 1. Validar cuántos datos hay con este carácter especial: Ã© |

## Campo region

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Confirmar que no hay valores con caracteres especiales en la columna región. | 1. Validar que no haya datos con caracteres especiales. |
| 2 | Detectar registros en la columna región que contienen caracteres especiales. | 1. Validar que hay datos con caracteres especiales. |
| 3 | Verificar que al menos un registro contiene el valor "Norte" en la columna región. | 1. Validar que hay datos con el nombre "Norte". |
| 4 | Determinar cuántos registros tienen el valor "Norte" en la columna región. | 1. Validar número total de registros con el valor "Norte". |
| 5 | Confirmar que hay n cantidad de registros con el nombre Norte. | 1. Validar que hay n cantidad para Norte. |
| 6 | Confirmar que al menos un registro contiene el valor "Sur" en la columna región. | 1. Validar que hay datos con el nombre "Sur". |
| 7 | Determinar cuántos registros tienen el valor "Sur" en la columna región. | 1. Validar número total de registros con el valor "Sur". |
| 8 | Confirmar que hay n cantidad de registros con el nombre Sur. | 1. Validar que hay n cantidad para Sur. |
| 9 | Verificar que al menos un registro contiene el valor "Este" en la columna región. | 1. Validar que hay datos con el nombre "Este". |
| 10 | Determinar cuántos registros tienen el valor "Este" en la columna región. | 1. Validar número total de registros con el valor "Este". |
| 11 | Confirmar que hay n cantidad de registros con el nombre Este. | 1. Validar que hay n cantidad para Este. |
| 12 | Verificar que al menos un registro contiene el valor "Oeste" en la columna región. | 1. Validar que hay datos con el nombre "Oeste". |
| 13 | Determinar cuántos registros tienen el valor "Oeste" en la columna región. | 1. Validar número total de registros con el valor "Oeste". |
| 14 | Confirmar que hay n cantidad de registros con el nombre Oeste. | 1. Validar que hay n cantidad para Oeste. |
| 15 | Verificar que al menos un registro contiene el valor "Centro" en la columna región. | 1. Validar que hay datos con el nombre "Centro". |
| 16 | Determinar cuántos registros tienen el valor "Centro" en la columna región. | 1. Validar número total de registros con el valor "Centro". |
| 17 | Confirmar que hay n cantidad de registros con el nombre Centro. | 1. Validar que hay n cantidad para Centro. |
| 18 | Determinar cuántos registros en la columna región comienzan con la letra "A". | 1. Validar número total de registros que empiezan con la letra "A". |
| 19 | Confirmar que hay n cantidad de registros con el nombre A. | 1. Validar que hay n cantidad para A. |
| 20 | Determinar cuántos registros en la columna región comienzan con la letra "E". | 1. Validar número total de registros que empiezan con la letra "E". |
| 21 | Confirmar que hay n cantidad de registros con el nombre E. | 1. Validar que hay n cantidad para E. |
| 22 | Determinar cuántos registros en la columna región comienzan con la letra "T". | 1. Validar número total de registros que empiezan con la letra "T". |
| 23 | Confirmar que hay n cantidad de registros con el nombre T. | 1. Validar que hay n cantidad para T. |
| 24 | Determinar cuántos registros en la columna región comienzan con la letra "N". | 1. Validar número total de registros que empiezan con la letra "N". |
| 25 | Confirmar que hay n cantidad de registros con el nombre N. | 1. Validar que hay n cantidad para N. |
| 26 | Determinar cuántos registros en la columna región comienzan con la letra "C". | 1. Validar número total de registros que empiezan con la letra "C". |
| 27 | Confirmar que hay n cantidad de registros con el nombre C. | 1. Validar que hay n cantidad para C. |
| 28 | Verificar que no existan valores nulos o vacíos en la columna región. | 1. Validar que no haya datos nulos o vacíos. |
| 29 | Confirmar que no hay valores numéricos en la columna región. | 1. Validar que no haya datos numéricos. |
| 30 | Determinar cuántos registros tienen los valores "Norte" o "Sur" en la columna región. | 1. Validar número total de registros combinados de "Norte" y "Sur". |

## Campo metodo\_pago

|  |  |  |
| --- | --- | --- |
| ID Caso | Descripción caso | Resultado esperado |
| 1 | Comprobar que los valores en metodo\_pago no contienen caracteres especiales. | 1. Validar datos sin caracteres especiales. |
| 2 | Determinar cuántos registros en metodo\_pago están libres de caracteres especiales. | 1. Validar cuántos datos sin caracteres especiales hay. |
| 3 | Confirmar si el número de registros sin caracteres especiales coincide con un valor esperado n. | 1. Validar que hay n cantidad de datos sin caracteres especiales. |
| 4 | Verificar si existen registros en metodo\_pago que contengan caracteres especiales. | 1. Validar datos con caracteres especiales. |
| 5 | Determinar cuántos registros contienen caracteres especiales en la columna metodo\_pago. | 1. Validar cuántos datos con caracteres especiales hay. |
| 6 | Confirmar si el número de registros con caracteres especiales coincide con un valor esperado n. | 1. Validar que hay n cantidad de datos con caracteres especiales. |
| 7 | Confirmar si existe al menos un registro con el valor "PayPal" en metodo\_pago. | 1. Validar que hay datos con el nombre "PayPal". |
| 8 | Determinar cuántos registros contienen el valor "PayPal" en metodo\_pago. | 1. Validar cuántos datos hay con el nombre "PayPal". |
| 9 | Confirmar si el número de registros con "PayPal" coincide con un valor esperado n. | 1. Validar que hay n cantidad para "PayPal". |
| 10 | Confirmar si existe al menos un registro con el valor "Efectivo" en metodo\_pago. | 1. Validar que hay datos con el nombre "Efectivo". |
| 11 | Determinar cuántos registros contienen el valor "Efectivo" en metodo\_pago. | 1. Validar cuántos datos hay con el nombre "Efectivo". |
| 12 | Confirmar si el número de registros con "Efectivo" coincide con un valor esperado n. | 1. Validar que hay n cantidad para "Efectivo". |
| 13 | Confirmar si existe al menos un registro con el valor "Transferencia Bancaria" en metodo\_pago. | 1. Validar que hay datos con el nombre "Transferencia Bancaria". |
| 14 | Determinar cuántos registros contienen el valor "Transferencia Bancaria" en metodo\_pago. | 1. Validar cuántos datos hay con el nombre "Transferencia Bancaria". |
| 15 | Confirmar si el número de registros con "Transferencia Bancaria" coincide con un valor esperado n. | 1. Validar que hay n cantidad para "Transferencia Bancaria". |
| 16 | Confirmar si existe al menos un registro con el valor "Tarjeta de Crédito" en metodo\_pago. | 1. Validar que hay datos con el nombre "Tarjeta de Crédito". |
| 17 | Determinar cuántos registros contienen el valor "Tarjeta de Crédito" en metodo\_pago. | 1. Validar cuántos datos hay con el nombre "Tarjeta de Crédito". |
| 18 | Confirmar si el número de registros con "Tarjeta de Crédito" coincide con un valor esperado n. | 1. Validar que hay n cantidad para "Tarjeta de Crédito". |
| 19 | Contar cuántos registros en metodo\_pago comienzan con la letra A. | 1. Validar cuántos datos hay que empiecen con la letra A. |
| 20 | Contar cuántos registros en metodo\_pago comienzan con la letra E. | 1. Validar cuántos datos hay que empiecen con la letra E. |
| 21 | Contar cuántos registros en metodo\_pago comienzan con la letra T. | 1. Validar cuántos datos hay que empiecen con la letra T. |
| 22 | Contar cuántos registros en metodo\_pago comienzan con la letra N. | 1. Validar cuántos datos hay que empiecen con la letra N. |
| 23 | Contar cuántos registros en metodo\_pago comienzan con la letra P. | 1. Validar cuántos datos hay que empiecen con la letra P. |
| 24 | Confirmar si el número de registros que comienzan con "A" coincide con un valor esperado n. | 1. Validar que hay n cantidad con la letra A. |
| 25 | Confirmar si el número de registros que comienzan con "E" coincide con un valor esperado n. | 1. Validar que hay n cantidad con la letra E. |
| 26 | Confirmar si el número de registros que comienzan con "T" coincide con un valor esperado n. | 1. Validar que hay n cantidad con la letra T. |
| 27 | Confirmar si el número de registros que comienzan con "N" coincide con un valor esperado n. | 1. Validar que hay n cantidad con la letra N. |
| 28 | Confirmar si el número de registros que comienzan con "P" coincide con un valor esperado n. | 1. Validar que hay n cantidad con la letra P. |
| 29 | Confirmar que no haya registros nulos o vacíos en la columna metodo\_pago. | 1. Validar que no haya datos nulos o vacíos. |
| 30 | Confirmar que no existan valores numéricos en la columna metodo\_pago. | 1. Validar que no haya datos numéricos. |
| 31 | Determinar cuántos registros contienen los valores "PayPal" o "Efectivo" en la columna metodo\_pago. | 1. Validar cuántos datos hay para "PayPal" y "Efectivo". |

# Carga del archivo

Se carga el archivo ventas.csv suministrado para la prueba

# Scripts de las Funciones

Los caos de prueba y funciones que se presentarán a continuación están orientados a validar la integridad, calidad y consistencia de los datos de un archivo de ventas. La verificación incluye formatos explícitos, garantizar valores correctos tanto en datos numéricos con string y también verificando datos nulos, errores de formato, o caracteres especiales.

Se adjuntan las funciones para cada uno de los casos de pruebas descritos anteriormente, de igual manera, el código de estas funciones se las puede consultar en la carpeta de archivos en el tests\_cases.py

## Scripts de las pruebas para el campo fecha\_venta

Estos srcipts se hicieron con pandas con framework Pytest

## Script 1

## Script 2

## Script 3

## Script 4

## Script 5

## Script 6

## Script 7

## Script 8

## Script 9

## Script 10

## Script 11

## Script 12

# Resultados de las pruebas:

Tabla de los resultados de la ejecución de las funciones anteriormente adjuntas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nombre del tester: | José Luis García Quinayás | | | | |
| Fecha ejecución | 27-10-2024 | | | | |
| Descripción de las pruebas | Casos de prueba para validar información del archivo ventas.csv | | | | |
| Versión | 1.0 | | | | |
| ID Caso | Descripción Caso de Prueba | Resultado | Descripción del Resultado | Número de Incidentes | Observaciones |
| Caso 01 |  |  |  |  |  |
| Caso 02 |  |  |  |  |  |
| Caso 03 |  |  |  |  |  |
| Caso 04 |  |  |  |  |  |
| Caso 05 |  |  |  |  |  |
| Caso 06 |  |  |  |  |  |
| Caso 07 |  |  |  |  |  |
| Caso 08 |  |  |  |  |  |
| Caso 09 |  |  |  |  |  |
| Caso 10 |  |  |  |  |  |
| Caso 11 |  |  |  |  |  |
| Caso 12 |  |  |  |  |  |

De igual manera, esta tabla también se la puede consultar en la carpeta en donde están los archivos, ya que del fruto de la ejecución de los scripts se generó un reporte en donde están los resultados de las pruebas, el nombre del archivo es report\_test.xlsx

# Análisis de los problemas encontrados

1. Formato de fecha incorrecto

Se identifican fechas en formato DD/MM/AAAA y MM-DD-AAAA, eso afecta la precisión de los datos en los reportes y su interpretación.

1. Valores negativos en campos numéricos

En campos como precio, cantidad vendida se identifican valores negativos, lo cual no es lógico para los registros de ventas

1. Caracteres especiales

Se identifican caracteres especiales en campos ‘nombre\_cliente’, ‘categoria’ y ‘metodo\_pago’

1. Valores nulos y/o vacíos

Se encuentran registros o campos vacíos que afectan sustancialmente la capacidad de organizar, categorizar y realizar el análisis de datos correctamente

Estos problemas comprometen la integridad y coherencia de los datos, con lo cual es evidente la necesidad de un proceso de validación amplio y robusto

# Propuestas de mejora y Plan de acción

## Estrategias preventivas

1. Control y validación en las entradas de datos

Configurar validaciones minuciosas, estrictas y robustas en los formularios de entrada en los cuales se puedan restringir los formatos, rango de valores, y selección de opciones, ejemplo, listas de los campos que se necesite

1. Capacitación

Entrenar y/o indicar a los desarrolladores de software para que en los formularios de ingreso de datos activen controles o restricciones en cuando al tipo de dato que se ingresa

## Estrategias Correctivas

1. Limpieza o depuración de datos

Desarrollar scripts que ayuden en la identificación de datos erróneos y los corrijan, sean valores negativos, nulos, formatos incorrectos, entre otros.

1. Automatizar validaciones

Implementar scripts en Python que ayuden a identificar errores y al mismo tiempo los corrija en tiempo real, adicional, generando alertas en caso de que se identifiquen irregularidades

## Monitoreo y control

1. Implementar métricas y alertas

Como por ejemplo, el porcentaje de registros válidos par cada uno de los campos, y en caso de que se identifiquen errores, se activen automáticamente alertas

1. Revisiones

Continuas auditorias en pequeños rangos de tiempo para evaluar posibles errores con el fin de garantizar la limpieza o depuración continua del conjunto de datos

# Conclusiones y recomendaciones

Con lo hallado se identifica que existen distintos problemas de calidad de datos que afectan negativamente los análisis tanto presentes como futuros, de la misma manera, afecta la toma de decisiones basada en estos datos. Las metodologías ágiles nos permiten iterar rápido e identificar problemas por medio de pruebas independientes o modulares

Por último, como recomendaciones:

1. Implementación de validaciones en tiempo real para prevenir y disminuir problemas en el momento de la entrada de datos
2. Implementar scripts que tengan la capacidad de limpiar o depurar la información y con lo cual se garantice la calidad de la información
3. Capacitar a los desarrolladores en la importancia del control de la entrada de datos en los formularios