

Índice

1. Introducción	1
1.1. Propósito	1
1.2. Alcance	2
1.3. Resumen del producto	2
1.3.1. Perspectiva del producto	2
1.3.2. Funciones del producto	3
1.3.3. Características del usuario	3
1.3.4. Limitaciones	3
1.4. Definiciones	3
1.5. Siglas y abreviaturas	3
2. Referencias	3
3. Requisitos	3
Microbenchmarks	3
3.1. Funciones	3
3.1.1. Interfaz web	4
3.1.2. Interfaz CLI	4
3.1.3. Microbenchmarks	5
3.2. Requisitos de usabilidad	5
3.3. Interfaces externas	6
3.4. Requisitos de rendimiento	6
3.5. Requisitos de la base de datos lógica	6
3.5.1. Tipos de Información Utilizada por Diversas Funciones	6
3.5.2. Frecuencia de Uso	6
3.5.3. Capacidades de Acceso	6
3.5.4. Entidades de Datos y Sus Relaciones	6
3.5.5. Restricciones de Integridad	6
3.5.6. Seguridad	6
3.5.7. Requerimientos de retención de datos	6
3.6. Restricciones de diseño	6
3.7. Atributos del sistema de software	6
3.8. Información de soporte	7
4. Verificación	7
5. Apéndices	8
5.1. Supuestos y dependencias	8
5.2. Matriz de trazabilidad	8

1. Introducción

Este Documento de Requisitos de Software (SRS) detalla el desarrollo de una suite de microbenchmarks diseñada para medir características de rendimiento de diferentes GPUs a través de diferentes plataformas y APIs gráficos. El software constará de dos interfaces principales: una web, para ejecutar los microbenchmarks usando WebGPU desde un navegador, y una CLI, para ejecutar los microbenchmarks de forma nativa sobre Vulkan. Este documento especifica los requisitos funcionales, las características del usuario, el alcance del producto, y las verificaciones.

1.1. Propósito

El propósito principal del software es proporcionar una suite de microbenchmarks para medir características específicas del rendimiento de GPUs en plataformas que soportan WebGPU y Vulkan.

También facilita la ejecución de microbenchmarks propios de los usuarios.

Este software aborda la necesidad de obtener datos de rendimiento detallados y comparativos a través de diferentes combinaciones de hardware y plataformas, permitiendo a desarrolladores e investigadores optimizar su software para diferentes escenarios y evaluar el desempeño general de las GPUs.

1.2. Alcance

El alcance del software, denominado “wgpu microbench” abarca el diseño, desarrollo e implementación de los microbenchmarks para evaluar características del hardware de GPUs y las interfaces para ejecutarlos y visualizar resultados.

El software tiene dos objetivos principales:

- Simplificar el proceso de escribir microbenchmarks para GPU.
- Proveer un banco de microbenchmarks comparativos con operaciones comunes para entender las capacidades del hardware sobre el que se ejecutan.

El software incluirá dos interfaces que dan acceso a estas capacidades:

1. Interfaz web: Permite ejecutar microbenchmarks desde un navegador, recolectar datos, visualizar resultados, y proporcionar una opción para ejecutar microbenchmarks personalizados.
2. Interfaz CLI: Permite ejecutar microbenchmarks nativamente, compilando a Vulkan y ofrece resultados en texto.

El producto es autónomo y no interactúa con sistemas externos, aunque opera dentro del contexto de los APIs gráficos mencionados y las plataformas sobre las que se ejecutan.

1.3. Resumen del producto

1.3.1. Perspectiva del producto

Los microbenchmarks se implementarán utilizando el API de gráficos *wgpu* que se puede correr en navegadores web que tengan soporte para WebGPU o nativamente traduciendo a los APIs nativos Vulkan, Metal y DX12. Aunque solo se verificará la capacidad de correr los microbenchmarks sobre Vulkan, por las razones especificadas en la sección de limitaciones.

La figura 1 muestra las plataformas en las que los microbenchmarks pueden ser evaluados y las capas de APIs gráficos, plataformas y hardware.

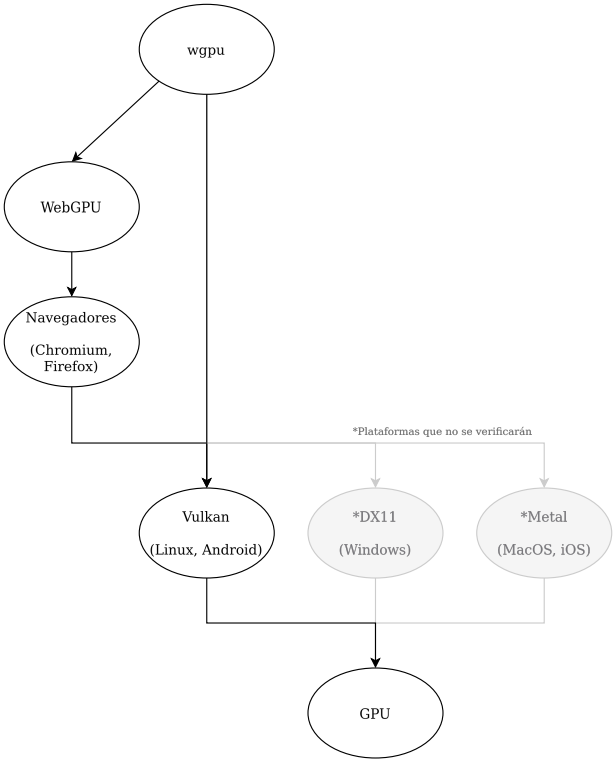


Figura 1: Jerarquía de APIs gráficos, plataformas y hardware sobre los que se puede ejecutar el API gráfico wgpu

No forma parte de un sistema mayor, el software es el producto entero que se ofrece. El desarrollo abarca todas las partes del producto, incluyendo la implementación de interfaces web y CLI.

1.3.2. Funciones del producto

Las principales funciones del software incluyen:

- **Ejecución de banco de microbenchmarks:** Medición de características específicas del rendimiento de GPUs.
- **Recolección y visualización de datos:** En la interfaz web, se recolectan y visualizan resultados; en la interfaz CLI, se generan resultados en texto.
- **Ejecución de microbenchmarks personalizados:** Facilita a los usuarios ejecutar y evaluar microbenchmarks personalizados a través de ambas interfaces.

1.3.3. Características del usuario

Los usuarios esperados incluyen desarrolladores de software de gráficos por computadora e investigadores en GPU compute u otros temas de rendimiento en GPU. Se espera que los usuarios tengan conocimientos sobre WebGPU y APIs gráficas, así como habilidades para interpretar resultados de benchmarks y optimizar el rendimiento de software para GPUs.

1.3.4. Limitaciones

El API de gráficos sobre el que se implementará el software, *wgpu*, permitiría compilar el software nativamente (interfaz CLI) para cualquier plataforma con los APIs gráficos Vulkan, Metal o DX12; lo cual incluye los sistemas operativos Windows, Mac y Linux. Sin embargo, el desarrollador se ve limitado en las plataformas a las que tiene acceso ya que no cuenta con hardware de Apple (necesario para utilizar la API gráfica Metal) ni licencia de Windows (necesario para utilizar la API gráfica DX12). Por lo tanto, solo se podrán verificar el funcionamiento del software para la plataforma Linux y consecuentemente el API de gráficos Vulkan.

De igual manera, la interfaz web solo se podrá verificar para los navegadores web disponibles en Linux (notablemente excluyendo a Safari que tiene soporte experimental de WebGPU) y en Android (notablemente excluyendo verificación en iOS).

1.4. Definiciones

- **Microbenchmark:** Prueba que mide el rendimiento de alguna característica específica de bajo nivel. Se diferencia de un “benchmark” ya que ese es un concepto más general que aplica a cualquier tipo de prueba que podría incluir el rendimiento de sistemas enteros.
- **GPU Compute:** Uso del GPU para cálculos computacionales de manera general.

1.5. Siglas y abreviaturas

GPU: Unidad de procesamiento gráfico. **CLI:** Interfaz de línea de comandos.

2. Referencias

- [Proyecto wgpu](#)
- [Estándar de WebGPU](#)
- [Proyecto pVulkanCompute](#)
- [Dissecting GPU Memory Hierarchy through Microbenchmarking](#)
- [Demystifying GPU Microarchitecture through Microbenchmarking](#)
- [Nvidia CUDA Programming Guide](#)
- [GPU Atomic Performance Modeling with Microbenchmarks](#)

3. Requisitos

Microbenchmarks

3.1. Funciones

Las funciones se especificaran en subsecciones para cada una de las interfaces propuestas y otra subsección sobre los requerimientos de los microbenchmarks que ambas interfaces implementará.

3.1.1. Interfaz web

- [11001]: Cuando el usuario inicie la acción de ejecutar microbenchmarks, la interfaz web deberá ejecutar el conjunto de microbenchmarks predeterminados.
- [11002]: Si el usuario indica que solo se ejecute uno de los microbenchmarks (puede ser a través de un selector), cuando el usuario inicie la acción de ejecutar microbenchmarks, la interfaz web deberá ejecutar solo el microbenchmark predeterminado que fue indicado.
- [11003]: La interfaz web deberá proporcionar un botón de “desactivar recopilación de datos” para el usuario.
- [11004]: Si se ejecuta el conjunto completo o solo uno de microbenchmarks predeterminados mientras el botón de “desactivar recopilación de datos” está inactivo, la interfaz web deberá recopilar los datos de rendimiento obtenidos de los microbenchmarks, el hardware y la plataforma del usuario y guardar esos datos en una base de datos.
- [11005]: Si se ejecuta el conjunto de microbenchmarks predeterminados mientras el botón de “desactivar recopilación de datos” está activo, la interfaz web no deberá recopilar ningún dato.
- [11006]: Cuando se terminan de ejecutar el conjunto completo o solo uno de los microbenchmarks predeterminados, la interfaz web deberá mostrar en la interfaz la plataforma, navegador web y hardware identificados para el usuario así como las características de rendimiento obtenidas de los microbenchmarks.
- [11007]: La interfaz web deberá proporcionar una caja de texto para que el usuario escriba el código de su propio microbenchmark personalizado.
- [11008]: Cuando el usuario lo indique, la interfaz web deberá ejecutar el microbenchmark personalizado del usuario.
- [11009]: Si el código del microbenchmark personalizado no puede compilar o ejecutarse por algún otro motivo cuando el usuario intenta ejecutarlo, entonces la interfaz web deberá indicar el error al usuario con una *alerta*.
- [11010]: Cuando finalice la ejecución correcta del microbenchmark personalizado, la interfaz web deberá mostrar cuánto tiempo duró en ejecutarse.
- [11011]: Si la plataforma o hardware del usuario no tiene soporte para alguna operación que se utilice en alguno de los microbenchmarks predeterminados cuando el usuario intente ejecutarlo, la interfaz web deberá indicar esta falta de soporte de operación al usuario y no ejecutará el microbenchmark específico.
- [11012]: La interfaz web deberá permitir a los usuarios acceder y ver estadísticas de los datos recopilados.
- [11013]: Cuando el usuario acceda las estadísticas, la interfaz web deberá obtener las estadísticas del servidor encargado de la base de datos.
- [11014]: Si el usuario aplica filtros (posiblemente a través de selectores) cuando accede las estadísticas, la interfaz web deberá obtener las estadísticas para un subconjunto de datos filtrados a como lo indicó el usuario; los filtros disponibles son para un hardware específico, un navegador web específico, una plataforma específica y un microbenchmark específico.
- [11015]: Si la petición falla (por algún error de conectividad o cualquier otra razón) cuando se intentan descargar las estadísticas para visualización, la interfaz web deberá abortar la operación de visualización e indicar el fallo al usuario por medio de una *alerta*, indicándole que vuelva a intentar más tarde.
- [11016]: La interfaz web deberá proveer un botón para descargar todos los datos recopilados de ejecuciones de microbenchmarks en formato CSV.
- [11017]: Si la petición falla (por algún error de conectividad o cualquier otra razón) cuando se intentan descargar el archivo de formato CSV con todos los datos de microbenchmarks realizados, la interfaz web deberá abortar la operación e indicar el fallo al usuario por medio de una *alerta*, indicándole que vuelva a intentar más tarde.

3.1.2. Interfaz CLI

- [12001]: Cuando el usuario ejecute el comando de la interfaz CLI, esta deberá ejecutar el conjunto de microbenchmarks predeterminados.
- [12002]: Si el usuario indica que solo se ejecute uno de los microbenchmarks (puede ser a través de una bandera en el comando), cuando el usuario ejecute el comando, la interfaz CLI deberá ejecutar solo el microbenchmark predeterminado que fue indicado.

- **[12003]:** Cuando se terminan de ejecutar el conjunto completo o solo uno de los microbenchmarks predeterminados, la interfaz CLI deberá mostrar dentro de la terminal las características de rendimiento obtenidas de los microbenchmarks.
- **[12004]:** La interfaz CLI deberá proporcionar la opción (posiblemente a través de una bandera en el comando) de leer un archivo de texto donde el usuario haya escrito el código de su propio microbenchmark personalizado y que este sea ejecutado.
- **[12005]:** Si el usuario le indica al comando que lea un archivo de texto para ejecutar su propio microbenchmark personalizado, al ejecutar el comando la interfaz CLI no deberá ejecutar ninguno de los microbenchmarks predeterminados.
- **[12006]:** Si el código del microbenchmark personalizado no puede compilar o ejecutarse por algún otro motivo cuando el usuario intenta ejecutarlo, entonces la interfaz CLI deberá indicar el error al usuario con un mensaje de texto en la terminal.
- **[12007]:** Cuando finalice la ejecución correcta del microbenchmark personalizado, la interfaz CLI deberá mostrar cuánto tiempo duró en ejecutarse.
- **[12008]:** Si el hardware del usuario no tiene soporte para alguna operación que se utilice en alguno de los microbenchmarks predeterminados cuando el usuario intente ejecutarlo, la interfaz CLI deberá indicar esta falta de soporte de operación al usuario y no ejecutará el microbenchmark específico.

3.1.3. Microbenchmarks

- **[13001]:** El banco de microbenchmarks deberá incluir al menos 2 microbenchmarks con métodos distintos de realizar convoluciones.
- **[13002]:** El banco de microbenchmarks deberá incluir al menos 2 microbenchmarks con métodos distintos de realizar multiplicación matricial.
- **[13003]:** El banco de microbenchmarks deberá incluir al menos 2 microbenchmarks con métodos distintos de realizar reducciones.
- **[13004]:** El banco de microbenchmarks deberá incluir al menos 2 microbenchmarks con métodos distintos de realizar la operación conocida como suma de prefijos o “scan”.
- **[13005]:** El banco de microbenchmarks deberá incluir al menos un microbenchmark para medir el ancho de banda de accesos de memoria secuenciales del GPU.
- **[13006]:** El banco de microbenchmarks deberá incluir al menos un microbenchmark para medir el ancho de banda de accesos de memoria desordenados del GPU.
- **[13007]:** El banco de microbenchmarks deberá incluir al menos un microbenchmark para medir el ancho de banda de copiar memoria entre buffers del GPU.
- **[13008]:** El banco de microbenchmarks deberá incluir al menos un microbenchmark para medir el ancho de banda de copiar memoria de buffer a texturas del GPU.
- **[13000]:** El banco de microbenchmarks deberá incluir al menos un microbenchmark para medir el ancho de banda de copiar memoria entre texturas del GPU.

3.2. Requisitos de usabilidad

- **[20001]:** Cada microbenchmark individual deberá producir resultados consistentes con una variación de no más del 5 % en múltiples ejecuciones en el mismo hardware y plataforma en un computador que no esté realizando otros procesamiento significativos.
- **[20002]:** Cada microbenchmark individual deberá ejecutarse en menos de 5 minutos ¹ en la computadora del desarrollador, específicamente una laptop equipada con una GPU integrada de un procesador Intel(R) Core(TM) i7-1260P de 12^a generación.

¹Este límite de tiempo ha sido seleccionado de manera arbitraria, pero razonable, basado en lo que se considera aceptable para la mayoría de los usuarios. Se anticipa que si un microbenchmark se completa dentro de este periodo en el hardware del desarrollador, probablemente se ejecutará más rápido en la mayoría de sistemas modernos en especial si cuentan con un GPU discreto. En casos donde la ejecución exceda este límite en otros dispositivos, se espera que el tiempo adicional no impacte significativamente la usabilidad.

3.3. Interfaces externas

No aplica.

3.4. Requisitos de rendimiento

No aplica.

3.5. Requisitos de la base de datos lógica

3.5.1. Tipos de Información Utilizada por Diversas Funciones

- [51001]: La base de datos deberá almacenar los resultados de los microbenchmarks, incluyendo métricas de rendimiento (por ejemplo, tiempo de ejecución o uso de memoria, variará dependiendo del microbenchmark), información de la plataforma (sistema operativo, navegador web, versión del navegador) y detalles del hardware (modelo del GPU).

3.5.2. Frecuencia de Uso

No aplica.

3.5.3. Capacidades de Acceso

- [53001]: La base de datos deberá permitir filtrar los datos de microbenchmarks según diversos criterios como plataforma, configuración de hardware y métricas de rendimiento específicas.
- [53002]: Si los usuarios solicitan descargar los datos, la base de datos deberá proporcionar la capacidad de generar y descargar archivos CSV que contengan todo el cuerpo de datos.

3.5.4. Entidades de Datos y Sus Relaciones

- [54001]: La base de datos deberá incluir una entidad para cada categoría de microbenchmark (como convolución, reducción, otros) con los datos relevantes para esa categoría.
- [54002]: La base de datos deberá tener la capacidad de incluir una entidad para microbenchmark específico si requiere datos adicionales específicos al microbenchmark.
- [54003]: La base de datos deberá incluir una entidad para plataformas que incluya la información de sistema operativo, navegador web, versión de navegador y hardware.

3.5.5. Restricciones de Integridad

- [55001]: Siempre que se almacene un resultado de microbenchmark, la base de datos deberá imponer integridad referencial, asegurando que un resultado de microbenchmark no pueda existir sin datos válidos asociados de plataforma y hardware.

3.5.6. Seguridad

No aplica.

3.5.7. Requerimientos de retención de datos

No aplica.

3.6. Restricciones de diseño

No aplica.

3.7. Atributos del sistema de software

- [70001]: La interfaz web deberá ser portátil, funcionando de manera efectiva en todas las plataformas objetivo oficiales: Chromium y Firefox, en Linux y en Android.

- **[70002]**: Los microbenchmarks deberán ser escritos utilizando el API gráfico *wgpu*, el cual permite portabilidad entre múltiples entornos web y nativos; garantizando un rendimiento y usabilidad consistentes en estos entornos de manera individual (es decir, que el rendimiento no tiene que ser consistente entre plataformas diferentes).

3.8. Información de soporte

No aplica.

4. Verificación

Esta sección especifica el método de verificación para cada requerimiento.

- **[11001]**: Ejecutar todos los microbenchmarks, verificar que se ejecutaron todos viendo los resultados.
- **[11002]**: Ejecutar un solo microbenchmark, verificar que solo se ejecutó ese porque debería de durar considerablemente menos que ejecutarlos todos y solo se generaron resultados para ese microbenchmark.
- **[11003]**: Abrir la interfaz web y verificar visualmente la existencia del selector y que se puede estripar con una confirmación visual de si está seleccionado o no.
- **[11004]**: Ejecutar microbenchmarks y luego revisar manualmente en la base de datos que llegaron los resultados de la ejecución.
- **[11005]**: Ejecutar microbenchmarks con la recolección de datos deshabilitada y verificar que no se guardan los datos en la base de datos y ver en el tab de “Network” del navegador que no se hizo ninguna solicitud para enviar los datos.
- **[11006]**: Ejecutar microbenchmarks y verificar que la interfaz web muestra los datos solicitados.
- **[11007]**: Entrar a la interfaz web y verificar que se cuenta con un espacio para escribir microbenchmarks personalizados.
- **[11008]**: Ejecutar un microbenchmark personalizado y verificar su ejecución correcta.
- **[11009]**: Ingresar diferentes códigos con errores de sintaxis y verificar que la página alerta de que no se pudo compilar el microbenchmark e informa el porqué.
- **[11010]**: Ejecutar uno de los microbenchmark predeterminados pero ingresándolo como si fuera uno personalizado, verificar que la interfaz web reporta el tiempo de ejecución correcto y acorde a lo que se espera cuando se ejecuta como predeterminado.
- **[11011]**: Temporalmente agregar un microbenchmark cuyo único requisito es usar alguna característica para la cual el GPU del dispositivo no tenga soporte, verificar que la página informa la falta de soporte.
- **[11012], [11013]**: Ejecutar varios microbenchmarks y luego verificar que se muestran los resultados recopilados de esas ejecuciones.
- **[11014]**: Ejecutar los microbenchmarks en los navegadores con soporte oficial y conseguir ejecutarlos en un computador con hardware diferente, tomar nota de los resultados. Luego, verificar que las estadísticas retornadas con los filtros aplicados sean consistentes con los resultados obtenidos en cada plataforma y hardware diferente.
- **[11016]**: Después de múltiples ejecuciones de los microbenchmarks en plataformas y hardware diferentes, verificar que puedo descargar el CSV con los datos de todas las ejecuciones realizadas.
- **[11015], [11017]**: Poner el dispositivo en modo avión y verificar que la interfaz informa del fallo al intentar descargar los datos.
- **[12001]**: Ejecutar el comando, verificar que se ejecutaron todos viendo los resultados.
- **[12002]**: Ejecutar el comando utilizando la bandera para ejecutar un solo microbenchmark, verificar que solo se ejecutó ese porque debería de durar considerablemente menos que ejecutarlos todos y solo se generaron resultados para ese microbenchmark.
- **[12003]**: Ejecutar los microbenchmarks y verificar que se imprimen los resultados para cada uno.
- **[12004]**: Ejecutar el comando con la bandera y un microbenchmark personalizado mínimo, verificar que no hay ningún error.

- **[12005]**: Ejecutar el comando con la bandera y un microbenchmark personalizado mínimo, verificar que duró el tiempo esperado (mucho menos que con los microbenchmarks reales) y que no se muestran resultados de ningún microbenchmark predeterminado.
- **[12006]**: Ingresar diferentes códigos con errores de sintaxis y verificar que el comando informa que no se pudo compilar el microbenchmark e informa el porqué.
- **[12007]**: Ejecutar uno de los microbenchmark predeterminados pero ingresándolo como si fuera uno personalizado, verificar que la terminal reporta el tiempo de ejecución correcto y acorde a lo que se espera cuando se ejecuta como predeterminado.
- **[12008]**: Temporalmente agregar un microbenchmark cuyo único requisito es usar alguna característica para la cual la GPU del desarrollador no tenga soporte, verificar que la página informa la falta de soporte.
- **[51001]**, **[53001]**, **[54001]**, **[54002]**, **[54003]**: Para todos los requerimientos relacionados a la existencia del esquema, de las entidades y la capacidad de accederlas y de poder filtrarlas se realizarán pruebas unitarias con una base de datos local de prueba que verifique la capacidad de insertar y recuperar datos de las maneras especificadas.
- **[53002]**: Crear una prueba unitaria para verificar la capacidad de insertar ciertos datos en la base de datos de prueba y luego obtenerlos en formato CSV.
- **[55001]**: Crear una prueba unitaria que demuestre que si se intenta guardar un dato que rompe con la integridad referencial la base de datos dé un error.
- **[70001]**: Llevar a cabo la verificación de requisitos funcionales (los que empiezan con '1') en cada una de las 4 combinaciones de navegador web y sistema operativo que conforman las plataformas objetivo oficiales.
- **[70002]**: En cada una de las 4 plataformas objetivo oficiales, verificar la ejecución correcta de todo el banco de microbenchmarks y verificar con un solo microbenchmark la consistencia de resultados.

5. Apéndices

5.1. Supuestos y dependencias

- **Soporte de *wgpu***: La portabilidad de los microbenchmarks depende de que el API gráfico *wgpu* mantenga el soporte para todas las plataformas objetivo.
- **Compatibilidad de Navegadores**: La implementación de la interfaz web depende de que los navegadores en las plataformas objetivo mantengan el soporte para WebGPU y WASM.
- **Compatibilidad del Hardware**: Se asume que los sistemas donde se ejecutará el software cuentan con hardware que es compatible con los drivers a través de los que trabaja *wgpu* en el sistema operativo, particularmente Vulkan.
- **Desempeño del Hardware del Desarrollador**: La especificación de tiempos de ejecución de los microbenchmarks **[20002]** depende del desempeño del hardware específico del desarrollador (una laptop con una GPU integrada de un procesador Intel(R) Core(TM) i7-1260P de 12^a generación). Cambios en el hardware o en las características de desempeño podrían requerir ajustes en los tiempos estimados en los requerimientos, por ejemplo si se obtuviera una tarjeta de gráficos discreta se requeriría un tiempo de ejecución menor.

5.2. Matriz de trazabilidad

No aplica.